X.systems.press

X.systems.press ist eine praxisorientierte Reihe zur Entwicklung und Administration von Betriebssystemen, Netzwerken und Datenbanken.

Thomas Schwenkler

Sicheres Netzwerkmanagement

Konzepte, Protokolle, Tools

Mit 103 Abbildungen und 52 Tabellen

Springer

Thomas Schwenkler
Soest
thomas.schwenkler@suedufer.de

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

ISSN 1611-8618
ISBN-10 3-540-23612-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-23612-2 Springer Berlin Heidelberg New York

Springer ist ein Unternehmen von Springer Science+Business Media
springer.de

Satzerstellung: durch den Autor
Herstellung: LE-TEX, Jelonek, Schmidt & Vöckler GbR, Leipzig
Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg
Gedruckt auf säurefreiem Papier 33/3142 YL – 5 4 3 2 1 0

Dieses Buch ist all den IT Managern gewidmet,
die in ihrer täglichen Arbeit unentdeckt bleiben
und nur selten das ihnen zustehende Lob erhalten,
nur weil sie einen guten Job machen.

Vorwort

Das Thema dieses Buches ist das Sichere Netzwerkmanagement. Auf den folgenden Seiten dreht sich alles um das Netzwerkmanagement, das seit dem Aufbau der ersten Netzwerke eine entscheidende Rolle gespielt hat. In vielen Jahren sind dabei die unterschiedlichsten Konzepte, Protokolle und Tools entwickelt worden, mit denen sich diese Aufgabe besser bewältigen lässt. An dieser Stelle werden keine grundsätzlich neuen Ideen eingeführt; vielmehr soll in diesem Buch die Wichtigkeit der Absicherung des Netzwerkmanagements vor den zahllosen Bedrohungen aus dem Intranet und dem Internet hervorgehoben werden. Trotzdem auch die Netzwerksicherheit in der heutigen Zeit zu einem großen Thema geworden ist, wird gerade im Zusammenhang mit dem Netzwerkmanagement noch allzuoft auf unsichere Methoden zurückgegriffen. Dabei ist das Netzwerkmanagement mit seinen unbegrenzten Möglichkeiten im Netzwerk immer mehr in das Fadenkreuz von skrupellosen Angreifern gerückt. Schließlich besitzt derjenige die Macht über das Netzwerk, der auch das Netzwerkmanagement kontrolliert. Mein Ziel ist es daher, dass die beiden großen Themenbereiche Netzwerkmanagement und Netzwerksicherheit zusammenfinden, damit auch morgen die Netzwerkinfrastruktur noch zu unseren Diensten steht und wir ihr unser Vertrauen aussprechen können.

Aufbau des Buches

Dieses Buch ist aus vier großen Teilen aufgebaut. Im ersten Teil finden sich hauptsächlich die Grundlagen für den Rest des Buches. Kapitel 1 liefert einen Überblick über die Begrifflichkeiten im Netzwerkmanagement und führt gleichzeitig den Begriff des Sicheren Netzwerkmanagements ein. In Kapitel 2 werden die Grundlagen des Netzwerkmanagements beleuchtet und verschiedene Kategorien eingeführt.

Der zweite und größte Teil des Buches widmet sich den unterschiedlichen Protokollen des Netzwerkmanagements und dessen Umfeld. Kapitel 3 taucht

in die Tiefen der Netzwerktechnologie ab und beschreibt mit ICMP dasjenige Protokoll, mit dessen Hilfe der gesamte IP Netzwerkverkehr auf unterster Ebene verwaltet und kontrolliert wird. Im Anschluss findet sich auch eine Liste verschiedener Tools und Programme, mit denen ein Administrator direkt auf ICMP Funktionen zugreifen kann. Kapitel 4 beleuchtet das wohl wichtigste Netzwerkmanagementprotokoll SNMP in allen seinen Versionen. Besonderes Augenmerk wird dabei auf das neuere SNMPv3 mit seinen Sicherheitsfunktionalitäten gelegt. In Kapitel 5 wird mit dem SYSLOG und dem SYSLOG-NG ein ganz anderer Mechanismus des Netzwerkmanagements vorgestellt. Auch hier wird eine Einführung in die eher unbekannten Sicherheitsmechanismen gegeben. Kapitel 6 versucht in aller Kürze einen Überblick des Systemmanagementprotokolls IPMI zu liefern. Die von mehreren großen Herstellern gemeinschaftlich erstellte Spezifikation enthält die unterschiedlichsten Sicherheitsmechanismen, die auch das im weitesten Sinne zum Netzwerkmanagement zählende Systemmanagement absichern kann. In Kapitel 7 wird eine Einführung in den IEEE 802.1X Standard gegeben, der eine Zugangskontrolle für Hardware-Komponenten auf das Netzwerk realisiert. Die Authentizität der einzelnen Systeme ist ein entscheidender Faktor eines Sicheren Netzwerkmanagements. Schließlich listet Kapitel 8 noch weitere Kommunikationsformen des Netzwerkmanagements auf. Zu diesen zählen neben RMON vor allem die unterschiedlichen Zugangsmöglichkeiten zur Administration von Netzwerkkomponenten.

Teil drei dieses Buches widmet sich ganz den vielfältigen Bedrohungen für Netzwerke und das Netzwerkmanagement vom Internet ausgehen. Kapitel 9 liefert eine Klassifikation für die verschiedenen Bedrohungen und Angriffsformen. Eine Analyse der häufigsten Angriffsziele rundet die Betrachtung ab. In Kapitel 10 wird anschließend in mehreren Beispielen versucht, für jede der Bedrohungen einen passenden Schutzmechanismus vorzustellen. Eine kurze Vorstellung von Honeypots zeigt außerdem eine Möglichkeit zur offensiveren Verteidigung von Netzwerken und deren Management auf.

Der vierte und letzte Teil des Buches steht ganz im Zeichen der Praxis. Aus diesem Grund werden in Kapitel 11 auch eine ganze Reihe von verschiedensten Werkzeugen aus den Bereichen SNMP, IPMI und IEEE 802.1X vorgestellt. Da eine vollständige Auflistung aller Tools unmöglich ist, soll zumindest eine Übersicht über einige kommerzielle und nicht-kommerzielle Programme gegeben werden. Kapitel 12 hinterleuchtet ganz andere Hindernisse, die sich in der Praxis einem Sicheren Netzwerkmanagement entgegenstellen. Eine flüchtiger Einblick in die Kostenrechnung zeigt die Schwierigkeiten auf, mit denen das Querschnittsthema Sicherheit im täglichen Leben zu kämpfen hat – ohne eine endgültige Lösung dafür bieten zu können. Schließlich wagt Kapitel 13 einen vorsichtigen Ausblick auf kommende Chancen und auch Risiken.

Im Anhang des Buches finden sich noch umfangreichere Tabellenwerke. Anhang A listet mehrere Hundert Request for Comments des SNMP Standards auf. In Anhang B findet sich eine Liste mit allen Hardware und Software Herstellern, die zur Zeit der Erstellung dieses Buches die IPMI Spezifikation

unterstützen. Auf Grund der überaus großen Anzahl an Abkürzungen, die im gesamten Themenkomplex des Sicheren Netzwerkmanagements zu finden sind, wurde abschließend Anhang C mit einer vollständigen Liste sämtlicher in diesem Buch verwendeten Abkürzungen angehängt.

Das Literaturverzeichnis ist vor allem Dank der vielen veröffentlichten Standards äußert umfangreich und liefert ausreichend Material für weiterführende Recherchen.

Konventionen

In diesem Buch werden zur besseren Lesbarkeit für einige Inhalte besondere Textauszeichnungen verwendet. In den abgedruckten Beispielen, bei denen eine Interaktion über eine Kommandozeile stattfindet, sowie bei Angaben von SNMP OIDs werden dabei über das gesamte Buch die folgenden Konventionen eingehalten:

`Schreibmaschinenschrift`	wird für Ausgaben des Systems auf der Kommandozeile verwendet. Der Inhalt von Konfigurationsdateien, Logging-Dateien und Programmquelltexte wird ebenfalls auf diese Weise abgedruckt. Außerdem besitzen Dateinamen dieselbe Formatierung.
fette, kursive Schrift	wird für Eingaben des Benutzers verwendet.
geschwungene Schrift	wird für SNMP OIDs verwendet.

Danksagung

An dieser Stelle möchte ich allen Personen danken, die das Erscheinen dieses Buches erst möglich gemacht haben. Da ist zunächst Frank Schmidt zu nennen, der mich in seiner damaligen Funktion als Lektor des Springer Verlages davon überzeugt hat, dieses Buch zu schreiben. Ihm verdanke ich die Idee und den Ansporn, den Themenkomplex des Sicheren Netzwerkmanagements in einem Buch niederzuschreiben. Des weiteren möchte ich meinem Kollegen Kai Simon danken, der mit aufschlussreichen Diskussionen und einiger tatkräftiger Unterstützung insbesondere in den Bereichen IEEE 802.1X und Sicherheit im Allgemeinen bei der Erstellung des Buches mitgeholfen hat. Danach danke ich meiner Mutter, die mit viel Mühe für ein hochwertiges Erscheinungsbild gesorgt hat. Außerdem möchte ich Manfred Schedlbauer danken, der bei der Gestaltung einiger der Abbildungen für ein weniger trockenes und schlichtes Erscheinungsbild gesorgt hat. Und schließlich möchte ich allen meinen Freunden danken, die eine große Geduld mit mir aufgebringen mussten. Ein Buch

von dieser Art schreibt sich nicht von alleine und erfordert ein großes Maß an Entbehrungen, was meine Freunde häufig zu spüren bekommen haben.

Mein letzter Dank geht an die Mitarbeiter des Springer Verlags, die mir die Möglichkeit eröffnet und die richtige Plattform geliefert haben, dieses Buch zu schreiben und zu veröffentlichen.

Soest, September 20005 *Thomas Schwenkler*

Inhaltsverzeichnis

Teil I Grundlagen

Teil II Protokolle

1

Sicheres Netzwerkmanagement: Begriffserklärungen

Seit der Entwicklung von Computern hat der Drang und der Bedarf an Vernetzung dieser Rechner stetig an Bedeutung und Gewicht zugenommen. Bei den ersten mechanischen Rechenmaschinen wurde eine Vernetzung noch dadurch erreicht, dass die errechneten Daten über externe Datenträger an andere Rechenmaschinen übergeben wurden. Der berühmte Physiker Richard Feynman erzählt recht amüsant in seinem biographischen Buch ‚Sie belieben wohl zu scherzen, Mr. Feynman' [67], wie in Amerika zur Zeit des Zweiten Weltkriegs die Forschung an der Entwicklung einer Nuklearbombe vorangetrieben wurde. Zur Lösung der mathematischen Probleme, welche den physikalischen Formeln beispielsweise zur Berechnung der freigesetzten Energie zugrunde lagen, wurden damals im militärischen Forschungszentrum in Los Alamos mechanische Rechenmaschinen eingesetzt. Für jede spezielle Aufgabe gab es eine eigene Maschine; es gab Addiermaschinen, Tabulatoren zum Bilden von längeren Summen, Multipliziermaschinen, Sortierer, Kollationiermaschinen zum Vergleichen und andere mehr. Die Maschinen arbeiteten mit Lochkarten, über welche sämtliche Ein- und Ausgaben abgewickelt wurden. Wollte man nun eine kompliziertere Berechnung mit den Rechenmaschinen anstellen, so mussten die Eingabewerte über Lochkarten in die Maschinen gegeben werden, die daraus Zwischenergebnisse berechneten, welche wiederum wiederholt als Eingabe für die nächste Maschine dienten. Auf der letzten Lochkarte nach dem letzten Rechenschritt war dann das Ergebnis ablesbar. Bei diesen Rechenmaschinen war das Netzwerk noch einfach und mechanisch über Lochkarten realisiert. Im Zuge der Weiterentwicklung wurden dann die Vernetzungen automatisiert, und die angeschlossenen Geräte konnten und mussten entsprechend für das Netzwerk konfiguriert werden. Zu Beginn lief auch dieser Prozess noch statisch. Vor Einführung des Domain Name Service (DNS), der eine dynamische Verwaltung von IP Adressen erlaubt, und des Dynamic Host Control Protocol (DHCP), durch welches Komponenten im Netzwerk dynamisch mit einer IP Adresse sowie Informationen zum DNS und grundlegenden Routing-Informationen versorgt werden können, wurden Netzwerkgeräte einmalig statisch konfiguriert und spätere Änderungen waren selten. Erst mit der steigen-

den Anzahl an Rechnern und dem stetig wachsenden Vernetzungsgrad kam die Notwendigkeit auf, Netzwerkgeräte schneller und dynamisch verwalten zu können. In diesem Zusammenhang spricht man von *Netzwerkmanagement*. Eine genauere Definition des Begriffs ‚Netzwerkmanagement' wurde 1989 von der International Organization for Standardization (ISO) im Open Systems Interconnection (OSI) Managementmodell festgelegt [96]. Die einzelnen dort festgelegten Kategorien des Netzwerkmanagements lassen sich in einer vereinfachten Klassifizierung in zwei grundlegend verschiedene Aufgabentypen zerlegen: die passive, beobachtende Aufgabe der Netzwerküberwachung und die aktive, beeinflussende Aufgabe der Netzwerkkonfiguration.

1.1 OSI Managementmodell

Im Open Systems Interconnection (OSI) Managementmodell werden primär fünf Funktionalitäten des Netzwerkmanagements unterschieden. Diese Funktionalitäten sind das Fehlermanagement, das Abrechnungsmanagement, das Konfigurationsmanagement, das Leistungsmanagement und das Sicherheitsmanagement. Weiterhin sind im OSI Managementmodell die verwalteten Objekte und Systeme näher definiert sowie die Kommunikationswege und Protokolle zwischen den Systemen. Unabhängig vom OSI Managementmodell kann noch eine Zeit-Dimension eingeführt werden, die aus den Phasen Planung, Realisierung, Betrieb und Migration besteht.

1.1.1 Funktionalität

In der Funktionalitäts-Dimension finden sich die Inhalte und Aufgaben des Netzwerkmanagements wieder. Im Vordergrund stehen selbstverständlich Erreichbarkeit und Verfügbarkeit der Systeme, aber auch die Sicherheit und die Abrechnung der vermittelten Daten.

Fehlermanagement

Das Fehlermanagement ist eng mit der Erreichbarkeit eines Systems verknüpft. Gutes Fehlermanagement bietet deshalb Mittel, um auftretende Fehler frühzeitig zu erkennen, zu isolieren und zu beheben. Zum Erkennen der Fehler eignen sich vor allem Maßnahmen wie eine Überwachung der Fehlerprotokolle oder die Entgegennahme von Fehlermeldungen. Dies schließt selbstverständlich eine geeignete Reaktion auf die erkannten Fehler ein.

Gerade im Netzwerkbereich, wo der Ausfall einer einzelnen Komponente eine große Anzahl von Fehlern und Folgefehlern erzeugen kann, stellt das Isolieren eines Fehlers ohne geeignete Hilfsmittel oftmals eine besondere Herausforderung dar. Zur Isolation der erkannten Fehler sollte daher sowohl eine Fehlerverfolgung durchgeführt werden als auch entsprechende Diagnosetests angewendet werden. Der wichtigste Schritt ist aber zweifelsohne die Behebung der erkannten und identifizierten Fehler.

Abrechnungsmanagement

Im Normalfall stellt ein Internet Service Provider (ISP) den Zugang zum Internet nicht kostenfrei zur Verfügung, sondern es fallen Gebühren für die Benutzung der Ressourcen an, welche nach den unterschiedlichen Preismodellen berechnet werden. Oftmals findet man ein volumenabhängiges Abrechnungsmodell, bei dem zusätzlich noch eine Grundgebühr für die Bereitstellung des Dienstes anfallen kann. Ein Teil der Gebühren berechnet sich demnach in Abhängigkeit vom übermittelten Datenvolumen. Nicht nur für den ISP ist deshalb die Erfassung von Abrechnungsdaten existenziell; auch die Endkunden besitzen ein berechtigtes Interesse an der Nachvollziehbarkeit und Überprüfbarkeit der erhobenen Gebühren. Abrechnungsdaten werden idealerweise am Übergang zwischen den Kunden und dem Dienstleister ermittelt, wobei dies an beiden Seiten gleichermaßen möglich ist. Das Abrechnungsmanagement beschäftigt sich mit der Verarbeitung und Verwaltung der anfallenden Abrechnungsdaten. Dazu zählt auch das Verwalten und Überwachen eventueller Kosten- und Ressourcenlimits sowie die Konfiguration der Netzwerkkomponenten bezüglich der Datenerfassung und Datenaggregation.

Konfigurationsmanagement

Die allgemeine Verwaltung der zu überwachenden Komponenten und Systeme fasst man unter dem Konfigurationsmanagement zusammen. Die darunter vereinten Funktionalitäten sind äußert vielschichtig und verfolgen gleichzeitig die unterschiedlichsten Ziele. Im OSI Managementmodell findet sich eine prägnante Beschreibung des Konfigurationsmanagements. Demnach ist es die Aufgabe des Konfigurationsmanagements, die am Management beteiligten Systeme zu identifizieren, Kontrolle über sie auszuüben, Daten von ihnen zu sammeln und ihnen Daten zur Verfügung zu stellen. Die möglichen Ziele dabei können das Vorbereiten von Verbindungen im Netzwerk sein, das Initialisieren und Starten dieser Verbindungen, das Sicherstellen einer kontinuierlichen Verbindung und das abschließende Beenden der Verbindungen. Typische Beispiele für Aufgaben des Konfigurationsmanagements sind:

- Die eindeutige Zuweisung von Namen für verwaltete Objekte und Objektgruppen.
- Konfiguration der Systeme und deren normale Betriebszustände.
- Starten, Stoppen und Konfigurieren der verwalteten Dienste und Objekte des Systems.
- Bedarfsorientierte Ermittlung von Informationen über das System und seinen aktuellen Zustand.
- Entgegennahme von Meldungen über Zustandsänderungen und außergewöhnliche Ereignisse bei den überwachten Systemen und Komponenten.
- Änderung der allgemeinen Konfiguration eines verwalteten Systems.

Leistungsmanagement

Über die Aufgaben des Fehlermanagements hinaus befasst sich das Leistungsmanagement insbesondere mit der Auslastung eines Systems. Hier zeigen sich die Unterschiede zwischen der einfachen Erreichbarkeit und der tatsächlichen Verfügbarkeit eines Systems. Typischerweise werden nicht nur aktuelle Werte bezüglich der Auslastung einer Komponente erfasst, sondern auch eine Historie über die Verfügbarkeit erstellt. So lassen sich beispielsweise Abweichungen vom Normalzustand (der „Baseline“) erfassen. Außerdem ermöglicht gutes Leistungsmanagement das Vorhersagen über die zukünftige Verfügbarkeit von Systemen, so dass man frühzeitig auf die sich ändernde Nutzung der Systeme reagieren kann. Schließlich enthält das Leistungsmanagement auch Aufgaben zur Konfiguration des Systems mit dem Ziel der Verbesserung der Verfügbarkeit.

Sicherheitsmanagement

Sicherheit ist heutzutage eines der zentralen Themen in Datennetzen. Nahezu alle Geschäfts-, Verwaltungs- oder Entwicklungsprozesse hängen zumindest teilweise von der Netzwerkinfrastruktur ab. Neben einer hinreichenden Verschlüsselung stehen hier vor allem Faktoren wie Zugangskontrolle oder effektive Abschirmung von der Außenwelt zum Schutz vor Angriffen im Vordergrund. Das Sicherheitsmanagement beinhaltet vorrangig die Überwachung und das Errichten oder Abbauen dieser Sicherheitsmechanismen. Außerdem fällt die Identifizierung und das Propagieren von Sicherheitsverstößen ebenfalls in den Bereich des Sicherheitsmanagements.

1.1.2 Management Information Base

Ein wichtiger Teil der Definition des OSI Managementmodells ist die klare Definition der Management Information Base (MIB). Unter der MIB ist die Gruppe der verwalteten Objekte innerhalb eines verwalteten Systems zu verstehen. Diese Objekte unterliegen grundsätzlich den beiden Funktionen Netzwerkverwaltung und Netzwerküberwachung. Daraus ergeben sich die verschiedenen Informationsflüsse im OSI Managementmodell. Bei der Netzwerküberwachung wandern die Informationen von den Objekten der MIB zu den Managementstationen. Bei der Netzwerkverwaltung sind es die Managementstationen, welche Informationen mit dem Ziel der Administration der MIB zu den verwalteten Geräten senden. Zu diesem Zweck muss nach dem OSI Managementmodell ein entsprechendes Managementprotokoll implementiert sein, welches die verschiedenen Aufgaben erfüllen kann.

1.1.3 Zeit-Dimension

Unabhängig vom OSI Managementmodell lassen sich die verschiedenen Funktionalitäten in unterschiedliche Zeit-Dimensionen eingruppieren. Jede der drei

chronologisch hintereinander angeordneten Phasen besitzt dabei andere Anforderungen an die fünf Funktionalitäten. Zu Beginn des Netzwerkmanagements steht die Planung, gefolgt von der Realisierung oder Umsetzung der Pläne. Zum Schluss steht die Überwachung des Betriebes in der Produktivphase. Eine weitere besondere Phase findet sich noch in der Migration innerhalb eines Netzwerkes, die eine Zyklus von Betrieb zurück zur Planung zur Folge hat.

Planung

In der Planungsphase eines Netzwerks sollte idealerweise auch das Netzwerkmanagement geplant werden. Hier müssen wichtige Entscheidungen getroffen werden, die einen unmittelbaren Einfluss auf die verschiedenen Funktionalitäten des Netzwerkmanagements haben. Dies gilt im Speziellen auch für die Sicherheit des Netzwerkes und des Netzwerkmanagements. Planungsfehler lassen sich häufig nur schwer oder mit großem Aufwand wieder beseitigen. Zusätzlich entstehen weitere Risiken bei der notwendigen Migration. Aus diesem Grund ist der Planungsphase besondere Aufmerksamkeit zu widmen.

Realisierung

Nach abgeschlossener Planung müssen Netzwerk und Netzwerkmanagement im Unternehmen installiert werden. Häufig zeigen sich gerade bei der Umsetzung unberücksichtigte Detailprobleme, die es zu lösen gilt. So kann es durchaus vorkommen, das mitten in der Realisierungsphase eine neue Planungsphase notwendig wird. Die Realisierungsphase ist mit der Umsetzung aller funktionalen und nicht-funktionalen Anforderungen abgeschlossen.

Betrieb

Das installierte Netzwerk muss für einen reibungslosen Betrieb ständig überwacht, angepasst und administriert werden. Das Netzwerkmanagement ist demnach ein zentraler Bestandteil der Produktivphase. Typische Aufgaben sind die Überwachung, Fehlerbeseitigung oder auch das Durchführen präventiver Maßnahmen zur Sicherstellung des laufenden Betriebs.

Migration

Eine Migration ist nicht nur bei der Bereinigung von Planungsfehlern notwendig. Auch das Angleichen der Netzwerkkomponenten und der Netzwerkinfrastruktur an die ständig steigenden Anforderungen macht regelmäßig eine Migration sowohl der Hardware als auch der Software notwendig. Die Migrationsphase vereint alle anderen Phasen der Zeit-Dimension in sich. Während der laufende Betrieb sichergestellt werden muss, müssen die Neuerungen gründlich geplant und eventuell im Produktivbetrieb umgesetzt werden.

1.2 Netzwerkmanagement = Konfiguration + Überwachung

Die im OSI Managementmodell definierten fünf Aufgaben der Funktionalitäts-Dimension lassen sich noch weiter in zwei Kategorien unterteilen, die sich durch die Richtung ihres Informationsflusses unterscheiden. Netzwerkmanagementaufgaben, bei denen Informationen ausschließlich von den administrierten Geräten zu den Managementsystemen übertragen werden, können als Netzwerküberwachung bezeichnet werden. Fließen die Informationen jedoch von den Managementsystemen zu den überwachten Komponenten, so lässt sich diese Aufgabe als Netzwerkkonfiguration bezeichnen. Oft geht die Netzwerkkonfiguration direkt mit einer Netzwerküberwachung einher, vor allem zur Überprüfung der korrekten Umsetzung der Konfigurationsanweisungen an die Netzwerkgeräte. Unter dem Begriff Netzwerkmanagement schließlich versteht man die Kombination aus einer Netzwerküberwachung und einer Netzwerkkonfiguration. Prinzipiell sind alle fünf Funktionalitäts-Kategorien des OSI Managementmodells sowohl mit einer Netzwerküberwachung als auch mit einer Netzwerkkonfiguration verbunden. Zur Verdeutlichung lassen sich die Informationsflüsse jedoch noch weiter gewichten. Abbildung 1.1 veranschaulicht die vereinfachten Zusammenhänge zwischen Informationsfluss und den fünf Funktionalitäten des OSI Managementmodells.

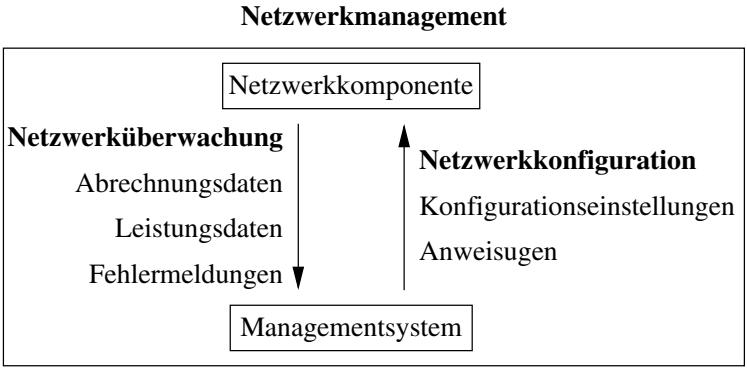

Abb. 1.1. Informationsfluss zu und von den überwachten Geräten eines Netzwerks.

Beim Netzwerkmanagement werden von einer oder mehreren Managementstationen, die von den Administratoren bedient werden, sowohl Lesezugriffe (Netzwerküberwachung) als auch Schreibzugriffe (Netzwerkkonfiguration) getätigt. Im Folgenden soll gezeigt werden, wie das Simple Network Management Protocol (SNMP) beide Aufgaben übernehmen kann. Deshalb wird sich Kapitel 4 ausführlich mit SNMP Rahmenwerk in seinen verschiedenen Versionen beschäftigen.

1.2.1 Netzwerkkonfiguration

Unter Netzwerkkonfiguration versteht man die Durchführung von Aufgaben im Netzwerk, bei denen die Einstellungen und Konfigurationen von Netzwerkkomponenten beeinflusst und verändert werden. Eine manuelle Konfiguration aller Netzwerkgeräte ist bei den heutigen komplexen Netzwerken nur noch schwer möglich. Hier kann eine vereinheitlichte Schnittstelle wie das Simple Network Management Protocol (SNMP) die Arbeit wesentlich erleichtern und vereinfachen. Zur Verdeutlichung soll an dieser Stelle das Beispiel einer typischen Netzwerkkonfigurationsaufgabe betrachtet werden. Konkret soll in diesem Beispiel die IP Adresse einer Netzwerkschnittstelle eines Cisco Routers umkonfiguriert werden. Diese Administrationsaufgabe wurde früher (und wird es heute noch sehr häufig) manuell über eine bequeme TELNET Verbindung zum Gerät durchgeführt. Abbildung 1.2 veranschaulicht die notwendigen Schritte zur Erledigung dieser Aufgabe. Die entstehenden Sicherheitsprobleme bei Verwendung des TELNET Dienstes werden später in Abschnitt 9.2.2 behandelt und sollen hier zunächst unberücksichtigt bleiben.

```
nms$ telnet 172.17.2.1
Login: root
Password: ******
Router> enable
Password: ******
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z
Router(config)# interface fastethernet 0
Router(config-if)# ip address 10.29.11.1 255.0.0.0
Router(config-if)# exit
Router(config)# exit
Router# disable
Router> exit
nms$
```

Abb. 1.2. Administration eines Cisco Routers über eine TELNET-Verbindung.

Um diesen Prozess – vor allem bei einer deutlich größeren Anzahl an Netzwerkgeräten – zu vereinfachen und damit die Administrierbarkeit des Netzwerkes weiterhin gewährleisten zu können, wurden im Laufe der Zeit verschiedene Netzwerkmanagementprotokolle entwickelt und etabliert, von denen das Simple Network Management Protocol (SNMP) das bekannteste und verbreitetste ist. Dieselbe administrative Aufgabe der Umkonfiguration der IP Adresse einer Netzwerkschnittstelle eines Cisco Routers kann man mittels SNMP mit zwei Befehlen über die Kommandozeile erledigen. Grafische Netzwerkmanagement-Werkzeuge machen sogar die Textkonsole überflüssig, und die Konfiguration

von Netzwerkgeräten erfordert nur wenige Mausklicks. Abbildung 1.3 zeigt die zur Erledigung der Administrationsaufgabe notwendigen Befehle auf der Konsole.

```
nms$ snmpset -c private 172.17.2.1 ipAdEntAddr.4 a 10.29.11.1
ipAdEntAddr.4 : IpAddress: 10.29.11.1
nms$ snmpset -c private 172.17.2.1 ipAdEntNetMask.4 a 255.0.0.0
ipAdEntNetMask.4 : IpAddress: 255.0.0.0
nms$
```

Abb. 1.3. Administration eines Cisco Routers mittels SNMP.

1.2.2 Netzwerküberwachung

Ein zweiter wichtiger Aspekt bei der Administration von Netzwerken ist die Überwachung der Netzwerkkomponenten. Die stetig steigende Anzahl an vernetzten Systemen erlaubt es einem Administrator heute kaum noch, jedes Gerät einzeln anzusprechen und sich Informationen über seinen Status anzueignen. Außerdem hat die ständig wachsende Abhängigkeit vom einwandfreien Funktionieren der Netzwerke und der daran angeschlossenen Systeme einen großen Einfluss auf die Netzwerküberwachung. Verfügbarkeit, Auslastung und weitere Parameter der einzelnen Systeme sind heute möglichst zeitnah zu erfassen, damit entsprechende Reaktionen auf Störungen schnell und zielgerichtet durchgeführt werden können. Dazu ist nicht nur die Überwachung der einzelnen Komponenten, sondern manchmal auch des Netzwerkes als Ganzes wünschenswert. Ein hilfreiches Konstrukt zur Ermittlung von Statistiken ganzer Netzwerke ist beispielsweise das Remote Monitoring (RMON). Der deutliche Unterschied zur Netzwerkkonfiguration liegt in der Passivität der Netzwerküberwachung. Die Aufgabe der Netzwerküberwachung liegt einzig im Sammeln und gegebenenfalls Auswerten von Daten über das Netzwerk und seine Komponenten.

Wieder soll an dieser Stelle ein Beispiel den Sachverhalt verdeutlichen. Eine mögliche administrative Aufgabe aus dem Bereich der Netzwerküberwachung ist das Ermitteln der Zeitspanne seit der letzten Neuinitialisierung eines Systems („Uptime"). In diesem Fall handelt es sich um denselben Cisco Router wie in den vorhergehenden Beispielen. Abbildung 1.4 veranschaulicht die zur Erfüllung der Aufgabe mittels einer TELNET-Verbindung notwendigen Schritte. Die Ausgabe wurde für eine bessere Lesbarkeit auf die relevante Zeile mit der Angabe zur Uptime gekürzt.

Auch diese Administrationsaufgabe lässt sich mit Hilfe von SNMP deutlich einfacher erledigen. Dazu ist lediglich ein einziger Befehl in der Textkonsole notwendig. Abbildung 1.5 zeigt den notwendigen Schritt, der allerdings eine

```
nms$ telnet 172.17.2.1
Login: root
Password: ******
Router> show version
...
Router uptime is 27 days, 13 minutes
...
Router> exit
nms$
```

Abb. 1.4. Überwachung eines Cisco Routers über eine TELNET-Verbindung.

weniger gut lesbare Form der Uptime enthält. Angezeigt werden die Hundertstelsekunden seit der letzten Neuinitialisierung des Gerätes.

```
nms$ snmpget -c public 172.17.2.1 system.3
system.sysUpTime : TimeTicks: 233359456
nms$
```

Abb. 1.5. Überwachung eines Cisco Routers mittels SNMP.

1.3 Sicheres Netzwerkmanagement

Dieses Buch beschäftigt sich mit dem Thema „Sicheres Netzwerkmanagement“, das nicht mit dem „Sicherheitsmanagement“ des OSI Managementmodells verwechselt werden darf. In den vergangenen Jahren ist Sicherheit zu einem zentralen Gegenstand der Informations- und Kommunikationstechnologie avanciert. Sicherheit als nicht-funktionale Anforderung ist ein Querschnittsthema, das nicht nur jeden einzelnen Internet-Nutzer betrifft, sondern vor allem auch Organisationen und Unternehmen, die größere Netzwerke betreiben und zu betreuen haben. Während das Sicherheitsmanagement die Administration und Verwaltung der Sicherheitsmechanismen im Netzwerk zur Aufgabe hat, darf dabei die Sicherheit des Netzwerkmanagements selbst nicht vernachlässigt werden.

Zu Beginn der Netzwerktechnik, als die ersten Managementfunktionen direkt über das Netzwerk erledigt wurden, hat sich kaum jemand ernsthafte Gedanken über die Sicherheit des Netzwerkmanagements gemacht, wie es in der heutigen Zeit geboten wäre. Frühere Zielsetzungen lagen primär in der Funktionalität; wichtig war also vorrangig das einwandfreie Funktionieren der

Netzwerkmanagement-Lösungen. Dies spiegelt sich vor allem in der ursprünglichen Version des Simple Network Management Protocols (SNMPv1) wider. Sicherheitsfunktionalitäten sind dort nur in minimalstem Umfang berücksichtigt worden. In der Nachfolgeversion SNMPv2[1] wurde das Thema Sicherheit zwar weiter angegangen, jedoch fehlte es an der letzten Konsequenz, die Sicherheitsfunktionalitäten auch bindend umzusetzen.

Das Problem der mangelnden Sicherheitsfunktionalitäten ist kein spezifisches Problem des Netzwerkmanagements. In vielen anderen Bereichen der Informations- und Kommunikationstechnologie findet sich dasselbe Problem der Vernachlässigung von Sicherheitsmechanismen. Auch die obigen Beispiele aus Abbildung 1.2 und Abbildung 1.4 arbeiten mit dem ebenfalls unsicheren *Telnet* Protokoll (siehe Seite 271). Auch das File Transfer Protocol (FTP) [164], welches eine einfache Möglichkeit zur Datenübermittlung zwischen zwei Rechnern bietet, unterstützt nur rudimentäre Sicherheitsfunktionalitäten. Parallel dazu existierten über einen langen Zeitraum die gleichermaßen unsicheren „r-Werkzeuge“ („r-Tools“) RCP (remote copy), RDIST (remote distribution), RLOGIN (remote login) und RSH (remote shell), die nur mit primitivsten Sicherheitsfunktionen ausgestattet waren. Erst viel später wurde das sicherere SSH (secure shell) mit den Ergänzungen SCP (secure copy) und SFTP (secure file transfer program) entwickelt, bei denen die gesamte Kommunikation verschlüsselt abläuft. Trotz der freien Verfügbarkeit von SSH durch „OpenSSH“ [146] finden sich viel zu häufig noch Bereiche, in denen – oftmals aus Bequemlichkeit – TELNET Verbindungen über unsichere Kommunikationswege zu sicherheitskritischen Komponenten aufgebaut werden.

Mit SNMPv3 wurden schließlich die dringend benötigten Sicherheitsmechanismen in das bewährte Netzwerkmanagementprotokoll auch praktisch eingeführt und umgesetzt. Bis zu diesem Zeitpunkt hatten sich allerdings die beiden ersten Versionen SNMPv1 und SNMPv2 in der Praxis derart etabliert, dass die Umstellung auf die sicherere SNMPv3 Version teilweise sträflich vernachlässigt worden ist. Noch heute existieren Netzwerkgeräte, welche das SNMPv3 Protokoll unzureichend oder gar nicht unterstützen. Die Management-Werkzeuge in der Praxis arbeiten nicht zuletzt aus diesem Grund oftmals in einer der älteren SNMP Versionen, da Funktionalität aus gutem Grund noch immer eine große Rolle spielt.

In diesem Buch soll vorrangig ein tieferes Verständnis und Bewusstsein für die Sicherheitsprobleme in heutigen Netzwerken geschaffen werden, die sich zwangsweise auch auf das Netzwerkmanagement ausweiten. Dazu wird das sichere Netzwerkmanagementprotokoll SNMPv3 im Kontext mit seinen Vorgängerversionen erläutert sowie dessen Vorteile hervorgehoben und näher beschrieben. Einen weiteren Schwerpunkt bilden die verschiedenen Bedrohungsformen und Angriffsformen aus dem Internet, die gleichermaßen für Netzwerke wie auch für das Netzwerkmanagement bestehen. Dabei soll ver-

[1] Abschnitt 4.5 beschäftigt sich mit den verschiedenen SNMP Versionen und deren Namensgebung.

deutlicht werden, dass gerade die Netzwerkmanagementinfrastruktur besonders schützenswert ist. Ziel soll es schließlich sein, dass die verfügbaren Sicherheitsmechanismen des Netzwerkmanagements entsprechend den heutigen Bedürfnissen angepasst und vor allem auch eingesetzt werden. Nur so kann in letzter Konsequenz von einem „Sicheren Netzwerkmanagement“ gesprochen werden.

2

Netzwerkmanagement Kategorien

Netzwerkmanagement ist immer abhängig von dem zu verwaltenden Netzwerk und dessen Struktur. Zur Kategorisierung des Netzwerkmanagements können verschiedene Eigenschaften herangezogen werden. Eine Unterscheidung findet sich in den im Netzwerk vorhandenen Systemen. Kommen vorrangig Geräte eines Herstellers zum Einsatz, so kann das Netzwerkmanagement anders aufgezogen werden, da die zur Verfügung stehenden Mechanismen einheitlich und im Normalfall auch kompatibel sind. Bei stark unterschiedlichen Systemen im Netzwerk treten ganz andere Probleme auf, die sich oftmals auf die verschiedenen Implementierungen und unterschiedliche Unterstützung von Standards zurückführen lassen.

Ein anderes Unterscheidungskriterium im Netzwerkmanagement sind die verschiedenen Anbindungen der verwalteten Systemen an die Netzwerkinfrastruktur. Während der Betrieb mit den Nutzdaten des Netzwerkes direkt mit den Protokollen und Paketen des Netzwerkmanagements in einem Datennetz kombiniert werden kann, lässt sich auch eine parallele Infrastruktur ausschließlich für Verwaltungszwecke etablieren. Dies hat insbesondere auf die Sicherheit des Netzwerkmanagements einen großen Einfluss.

Werden die Netzwerkmanagementdaten im selben Netz versendet, wie die Nutzdaten, dann ist eine ausreichende Kategorisierung und Priorisierung der verschiedenen Datenpakete von zunehmender Bedeutung. Aus diesem Grund ermöglicht das Internet Protokoll eine Einteilung aller Datenpakete in unterschiedliche Kategorien, die jeweils andere Anforderungen an das Netzwerk stellen.

2.1 Homogene Netzwerke

Unter einem homogenen Netzwerk ist ein Netzwerk zu verstehen, welches sich primär aus Komponenten von nur wenigen Herstellern zusammensetzt. Nicht immer bedeutet dies zwangsweise, dass sämtliche Geräte von einem einzigen

Hersteller stammen. Dies ist vor allem deshalb äußerst selten, da nur wenige Hersteller die vollständige Palette an verschiedenen Netzwerkkomponenten wie Firewalls, Layer-2/Layer-3 Router, einem Intrusion Detection System (IDS), Einwahl-Servern, Archivierungssystemen, Applikations-Servern für die verschiedensten Dienste oder auch Netzwerkmanagementstationen aufwarten können. Häufig können aber Gruppen von verschiedenen Netzwerksystemen vom selben Hersteller bezogen und eingesetzt werden. In vielen Netzwerken sind beispielsweise die vermittelnden Geräte der Sicherungsschicht des OSI Referenzmodells aus Komponenten desselben Herstellers aufgebaut. Gleichzeitig finden sich auch viele Netzwerke, bei denen die Applikations-Server der höheren Schichten des OSI Referenzmodells vom selben Hersteller stammen. In solchen Netzen ist eine bereits eingerichtete Überwachung und Konfiguration über das Managementprotokoll SNMP mit vergleichsweise wenig Aufwand um mehrere Einzelsysteme erweitert, da nur selten grundlegende Konfigurationsänderungen beim Netzwerkmanagementsystem notwendig sind. Implementierungsabhängige Abweichungen oder Unterstützungen von Standards sind bei Geräten desselben Herstellers in vielen Fällen ähnlich, so dass sich einmal erarbeitete Verfahren leicht auf zusätzliche Geräte anwenden und erweitern lassen.

Ein weiterer Vorteil von homogenen Netzwerken liegt in der besseren Werkzeugunterstützung. In den meisten Fällen bietet ein Hersteller von Systemen einer größeren Produktpalette auch ein geeignetes Managementwerkzeug für alle seine Gerätevarianten an. Eine hohe Kompatibilität zwischen der Hardware und der Netzwerkmanagement-Software ist somit beinahe immer gewährleistet. Vor allem die herstellerspezifischen verwaltungsfähigen Objekte in den Systemen werden kaum in einer anderen Software besser unterstützt. Durch den Einsatz einer perfekt zugeschnittenen Netzwerkmanagement-Software sind gleichzeitig auch die besten Voraussetzungen für eine hohe Sicherheit und eine optimale Funktionalität geschaffen. Abbildung 2.1 verdeutlicht den Aufbau eines homogenen Netzwerks.

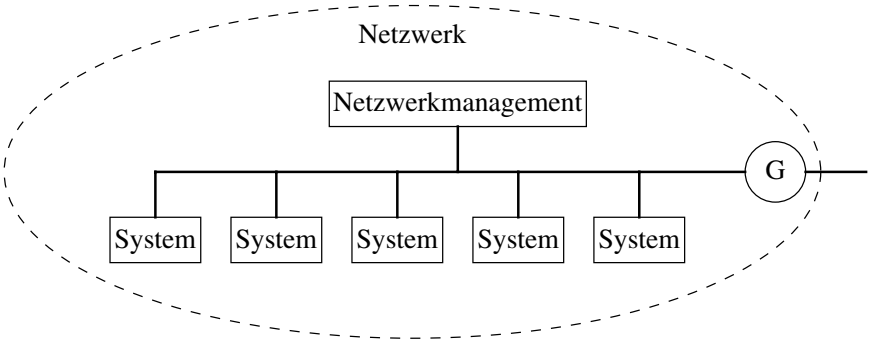

Abb. 2.1. Bei einem homogenen Netzwerk sind die Systeme alle vom selben Hersteller.

2.2 Heterogene Netzwerke

Die Vorstellung eines perfekten, homogenen Netzwerkes lässt sich nur in den wenigsten Fällen auch in die Praxis umsetzen. Mit dem Beginn der ersten Vernetzung von Computersystemen vor mehreren Jahrzehnten bis zur heutigen Zeit sind Netzwerke mit den an sie gestellten Anforderungen und den parallel dazu ausgearbeiteten Neuerungen und Fortschritten ständig weitergewachsen. Kaum ein heutiges Netzwerk ist seit seiner Planung und initialen Implementierung nicht einmal oder mehrmals strukturell umorganisiert und zu neueren Technologien migriert worden. Vor diesem Hintergrund ist es auch leicht verständlich, dass die meisten Netzwerke nicht homogen sind, sondern sich aus verschiedensten Systemen zusammensetzen, die teilweise auch aus unterschiedlichen „Zeitaltern“ stammen[1]. Bei derart gewachsenen Strukturen ist es auch verständlich, wenn nicht über den gesamten Zeitraum Geräte desselben Herstellers zum Einsatz kommen. Und selbst dann ist die Kompatibilität der Systeme untereinander fragwürdig, wenn sie aus verschiedenen Zeitaltern stammen. Abbildung 2.2 veranschaulicht die Struktur eines heterogenen Netzwerks.

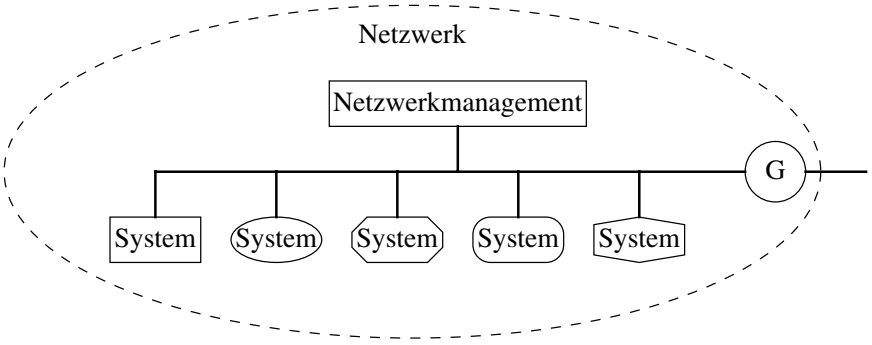

Abb. 2.2. Bei einem heterogenen Netzwerk herrscht Diversität zwischen den Herstellern der einzelnen Systeme.

Eine Netzwerkinfrastruktur, die aus Geräten verschiedenster Hersteller und unterschiedlicher Zeitalter zusammengesetzt ist, stellt das Netzwerkmanagement häufig vor nicht unerhebliche Probleme. Jedes System hat seine eigene Implementierung der Standards und unterstützt seinen eigenen Teilausschnitt

[1] Wenn hier die Rede von Zeitaltern ist, dann sind damit nicht geschichtliche Zeitalter oder Zeiträume in der Größenordnung gemeint, wie sie sonst nur Historiker oder gar Geologen verwenden würden. Vielmehr verläuft die Entwicklung in der Informationstechnologie in einem derart hohen Tempo, dass zwischen zwei „Zeitaltern“ nur wenige Jahre liegen können. Betrachtet man die Systeme von den Anfängen der Computertechnik und vergleicht sie mit heutigen Systemen, so wird sehr schnell klar, was an dieser Stelle unter „Zeitalter“ zu verstehen ist.

der vorgegebenen Richtlinien. Dazu kommen veraltete Systeme, die aktuelle Mechanismen nicht unterstützen, oder auch neuartige Systeme, deren Möglichkeiten gar nicht voll ausgeschöpft werden können. Gerade letztere Situation entsteht häufig dann, wenn das Netzwerkmanagement selbst veraltet oder zumindest nicht auf dem neuesten Stand ist. Als Folge aller dieser Probleme fließt ein nicht unerheblicher Teil des Aufwands vom Netzwerkmanagement in die Anpassung und Angleichung von Hardware und Netzwerkmanagement-Software aneinander. Im Vordergrund steht dann häufig der Aspekt Funktionalität – also das erfolgreiche Implementieren der unterschiedlichen geforderten Überwachungs- und Konfigurationsmechanismen. Gleichzeit rückt der Aspekt Sicherheit unweigerlich weiter in den Hintergrund. Für optimale Voraussetzungen eines sicheren Netzwerkmanagements ist folglich eine gut aufeinander abgestimmte Hardware und Software im Netzwerk durchaus hilfreich.

Andererseits darf ein heterogenes Netzwerk nicht verteufelt werden. Durch eine bewusste Vermischung verschiedener Systeme im Netzwerk kann unter ausgewählten Bedingungen auch die Sicherheit erhöht werden. Gerade in einem auf mehreren Schichten basierenden Sicherheitsmodell kann die Wirkung umso höher sein, je unterschiedlicher die Systeme in den verschiedenen Schichten sind. Und mit einer von Grund auf höheren Sicherheit ist gleichzeitig auch die Sicherheit des Netzwerkmanagements größer. Auf jeden Fall gilt es, die besonderen Umstände und Anforderungen an die Netzwerkinfrastruktur und das Netzwerkmanagement abzuwägen, bevor man einen steuernden Einfluss auf die Homogenität des Netzwerks ausübt.

2.3 Klassifikation des Datenverkehrs

In den Zeiten größerer Bandbreiten haben sich auch neue Möglichkeiten für Datennetzwerke eröffnet. Während zu Beginn des Internets und der Netzwerke die Bandbreite noch sehr gering war und damit auch eine entscheidende Grenze für die Nutzung darstellte, hat sich die Leistungsfähigkeit der Netzwerkanbindungen vervielfacht. Gleichzeitig ist auch die Nutzung des Internets um einen ähnlichen Faktor angestiegen, da mit der höheren Bandbreite auch die sich neu bietenden Möglichkeiten genutzt werden.

Diese neue Vielfalt an Netzwerkverkehr bringt gleichzeitig teilweise stark unterschiedliche Anforderungen an die darunterliegenden Netzwerke und Protokolle mit sich. In der auch heute noch weitgehend eingesetzten TCP/IP Protokollfamilie der Version 4 hatte man seit Beginn nur eine eingeschränkte Wahlmöglichkeit an erfüllbaren Anforderungen. Verantwortlich zeichnen dafür im Wesentlichen das verbindungslose Protokoll UDP und das verbindungsorientierte Protokoll TCP. Während das Protokoll UDP für eine Kommunikation mit wenig Overhead sorgt, aber dafür keine garantierte Auslieferung von Paketen verspricht, verfolgt das Protokoll TCP mit seinen expliziten Verbindungen zwischen den Kommunikationspartnern zwar für eine gute Verfolgbarkeit und Nachvollziehbarkeit des Datenaustausches. Erkauft wird die Auslieferungsga-

rantie[2] des verbindungsorientierten Protokolls TCP durch einen deutlich größeren Overhead, der sich beispielsweise schon beim initial notwendigen Drei-Wege-Verbindungsaufbau zeigt. Die beiden einzigen Metriken, zwischen denen eine Entscheidung besteht, sind die sich gegenseitig negativ beeinflussenden Attribute Übertragungsgeschwindigkeit und Zuverlässigkeit.

Das Internet Protokoll in der Version 6 (IPv6) – und in eingeschränktem Maße auch das Protokoll IPv4 – erlaubt eine genauere Klassifikation des Datenverkehrs, die eine detailliertere Spezifikation von Anforderungen ermöglicht. Zu diesem Zweck wurde in RFC 1349 [3] für IPv4 das vier Bit große Type of Service (ToS) Feld im IP Paketkopf spezifiziert. Ein vergleichbares Oktett existiert auch im Paketkopf von IPv6 unter der Bezeichnung „Traffic Class". Die Bedeutung dieses Oktetts ist identisch zum ToS Feld aus IPv4.

Die ursprünglich in [161] aufgestellte Definition des ToS Oktetts wurde mehrmals abgeändert, und nach RFC 1349 existiert nun die Wahlmöglichkeit zwischen genau einer der vier Anforderungen:

- geringe Verzögerung
- hoher Durchsatz
- hohe Zuverlässigkeit
- niedrige Kosten

Unterschiedliche Applikationen, zu denen auch das Netzwerkmanagement zählt, haben im Normalfall jeweils andere Anforderungen an das verbindende Netzwerk und werden daher auch das ToS Feld unterschiedlich belegen. Abbildung 2.3 zeigt den Aufbau desjenigen Oktetts im IP Paketkopf, welches auch das vier Bit große ToS Feld enthält.

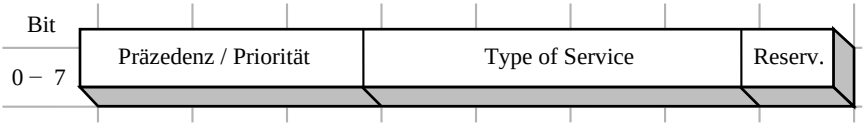

Abb. 2.3. Schematische Darstellung des dreigeteilten Oktetts im IP Paketkopf, welches das ToS Feld enthält.

Zusätzlich zu dem vier Bit großen ToS Feld befinden sich noch zwei weitere Angaben im selben Oktett des IP Paketkopfes. Die ersten drei Bit des Oktetts werden durch das Präzedenz-Feld[3] belegt. Ursprünglich für das Amerikanische Verteidigungsministerium Department of Defense (DoD) reserviert,

[2] Das Protokoll TCP kann genau genommen nicht die Auslieferung von Paketen garantieren oder versprechen. Eine physikalische Trennung des Kommunikationsweges verhindert unabhängig vom Protokoll eine korrekte Auslieferung der Pakete. Vielmehr bleibt aber eine Verfälschung oder der Verlust von Informationen bei der Übermittlung nicht unentdeckt.

[3] Der im Englischen verwendete Begriff „Precedence" lässt sich im speziellen Fall am besten mit dem deutschen Wort „Priorität" übersetzten, da die Angabe in diesem

wird das Präzedenz Feld des ToS Oktetts im IP Paketkopf mittlerweile auch von anderen verwendet. In diesem drei Bit großen Feld kann eine Angabe über die Wichtigkeit des Paketes gemacht werden. Bei der Weitervermittlung von Paketen sollen Router die in ihrem Ausgangspuffer befindlichen Pakete nach Wichtigkeit sortiert behandeln und Pakete mit höherer Präzedenz bevorzugen. Dieser Mechanismus steht im Widerspruch zu der Anforderung an eine geringe Verzögerung von Paketen, die sich im ToS Feld spezifizieren lässt. Allerdings sollte man an dieser Stelle nicht vergessen, dass die Angabe von verschiedenen ToS Anforderungen oder Präzedenzen keinen Anspruch auf Erfüllung erhebt. Es wird lediglich nach dem „Best-Effort“ Prinzip verfahren, das einzig nach der bestmöglichen Erfüllung und nicht nach der Garantie der Anforderungen bestrebt ist.

Das letzte Bit des Oktetts aus Abbildung 2.3 ist für zukünftige Erweiterungen reserviert. Im Laufe der Zeit wurden tatsächlich Wünsche für Erweiterungen und Verbesserungen des ToS und des „Traffic Class“ Feldes im IP Paketkopf laut, jedoch waren diese nicht mit dem hier beschriebenen Interpretationsverfahren der acht Bit aus dem Präzedenz/ToS Oktett vereinbar. Aus diesem Grund wurde in RFC 2474 [141] und in RFC 3168 [171] die Bedeutung des Präzedenz/ToS Oktetts umdefiniert und ein sechs Bit großes Differentiated Services (DS) Feld sowie ein zwei Bit großes Explicit Congestion Notification (ECN) Feld eingeführt (siehe Abbildung 2.4). Über diese acht Bit lassen sich nun noch detailliertere Angaben zur Behandlung der einzelnen IP Pakete machen.

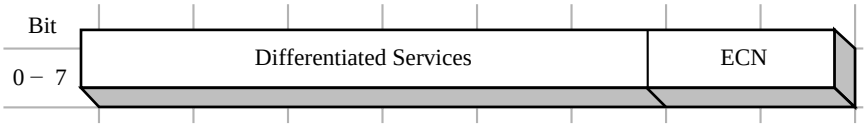

Abb. 2.4. Schematische Darstellung des zweigeteilten Oktetts im IP Paketkopf, welches das Differentiated Services (DS) Feld und das Explicit Congestion Notification (ECN) Feld enthält.

2.3.1 Datenverkehr mit geringer Verzögerung

Für manche Applikationen ist eine Datenübermittlung mit einer möglichst geringen Verzögerung der einzelnen Pakete von höchster Wichtigkeit. Zu diesen Applikationen zählt beispielsweise die Internettelefonie Voice over IP (VoIP), bei der bereits kleinere Verzögerungen zu einer spürbaren Beeinträchtigung der Kommunikation führen können. Um eine angenehme Sprachverbindung aufbauen zu können, dürfen daher nicht die Kosten der Verbindung oder die

Feld eine Aussage über die Wichtigkeit und damit auch über die Priorität bei der Behandlung des Paketes macht.

Auslieferungszuverlässigkeit eine maßgebende Rolle spielen. Eine Audiokommunikation kann eher mit einem einzelnen verlorenen Paket umgehen, als mit einem verzögert eintreffenden Paket. In geringem Maße lassen sich zwar Pakete zwischenpuffern und etwas zeitverzögert abspielen, jedoch existiert hier nur ein kleiner Spielraum, der nicht mit einer spürbaren Verschlechterung der Kommunikation verbunden ist. Wird die Audiokommunikation um ein Videosignal erweitert, so gelten für das Audiosignal grundsätzlich dieselben Regeln. Das Videosignal spielt für eine komfortable Unterhaltung jedoch eine untergeordnete Rolle und kann auch mit einer anderen Anforderung ausgestattet werden. Aus diesem Grund werden audiovisuelle Daten in vielen Fällen auch mit einem separaten Audiokanal und einem separaten Videokanal übertragen. Zur Kenntlichmachung von Datenverkehr mit der Anforderung an eine möglichst geringe Verzögerung wird das vier Bit große ToS Feld im IP Paketkopf auf den Wert 1000 gesetzt.

2.3.2 Datenverkehr mit hoher Bandbreite

Bei vielen Anwendungen ist eine Echtzeitübertragung von nur geringer Bedeutung. Häufiger stehen nämlich hohe Übertragungsraten im Vordergrund, die mit einer großen Bandbreite verknüpft sind. Beispielsweise spielt es bei der Übertragung von großen Dateien über das File Transfer Protocol (FTP) keine Rolle, ob die einzelnen Pakete jeweils eine geringe Verzögerung aufweisen. Vielmehr steht im Mittelpunkt, dass möglichst viele Pakete in kürzester Zeit den Empfänger erreichen. Dabei dürfen die Pakete auch unterschiedliche Wege mit unterschiedlicher Laufzeit und Verzögerung nehmen. Auch eine unsortierte Reihenfolge des Eintreffens der Pakete hat keinerlei negativen Einfluss auf die Datenübertragungsgeschwindigkeit. Gleiches gilt für die Auslieferungszuverlässigkeit, da auf dem Weg verlorengegangene Pakete einfach erneut übertragen werden können. Datenpakete mit der Anforderung einer hohen Bandbreite finden sich sehr häufig in verschiedenen Applikationen und tragen im ToS Feld des IP Paketkopfes den Wert 0100.

2.3.3 Datenverkehr mit hoher Auslieferungszuverlässigkeit

Neben einer geringen Verzögerung und einer hohen Bandbreite kann bei verschiedenen Applikationen auch die Zuverlässigkeit der Auslieferung der einzelnen Pakete eine große Rolle spielen. Zu diesen Applikationen zählen vorrangig die verschiedenen Netzwerkmanagementsysteme. Wird bei der Wahl des IP Protokolls für das Simple Network Management Protocol (SNMP) noch auf das verbindungslose Protokoll UDP gesetzt, das auch eine unzuverlässige Auslieferung der Pakete mit sich bringt, so kann durch die Wahl der ToS Anforderung einer möglichst hohen Auslieferungszuverlässigkeit diesem Nachteil entgegengewirkt werden. Auch das Internet Control Message Protocol (ICMP) leidet unter einer verbindungslosen und damit auch gleichzeitig unzuverlässigen Auslieferung der Pakete, so dass sich hier ebenfalls derselbe ToS Wert 0010 im IP Paketkopf empfiehlt.

2.3.4 Kostengünstiger Datenverkehr

Die vierte und letzte Anforderung, die ein Datenpaket in seinem IP Paketkopf an das Netzwerk stellen kann, bezieht sich auf die durch die Übermittlung der Pakete verursachten Kosten. Damit sind nicht die Kosten im Sinne von Router Hops gemeint, sondern die durch die Benutzung der Infrastruktur entstehenden monetären Kosten. Ein gutes Beispiel für eine Kommunikation mit der Anforderung einer möglichst geringen Kostenverursachung ist das Network News Transfer Protocol (NNTP), das zur Übermittlung von Nachrichten der verschiedenen Newsgroups dient. Für diese Kommunikation ist weder eine geringe Übertragungsverzögerung noch eine hohe Bandbreite oder eine zuverlässige Auslieferung der Pakete von besonderer Bedeutung. Vielmehr können bei den Nachrichten aus den Newsgroups die verursachten Kosten im Vordergrund stehen. Datenpakete mit der Anforderung an eine möglichst geringe Kostenverursachung tragen in ihrem ToS Feld im IP Paketkopf den Wert 0001.

Es lässt sich nicht verallgemeinern, zu welchen der anderen Anforderungen die Kosten komplementär sind. Beispielsweise kann eine Leitung sehr breitbandig aber auch stark frequentiert und damit mit hohen Verzögerungen belegt sein, während eine zweite Leitung wenig belastet ist aber nur über eine geringe Bandbreite verfügen kann. Die Kosten für die Nutzung dieser beiden Leitungen lassen sich aber nicht pauschalisieren. Es sollte also beachtet werden, dass geringe Kosten mit nicht vorhersagbaren Nachteilen verbunden sein kann. Im Grunde gilt für alle vier Anforderungen, dass sie einander in nur begrenzt vorhersagbarem Maße beeinflussen.

2.3.5 Unpriorisierter Datenverkehr

Nicht jedes IP Paket muss zwingend eine besondere Anforderung an das Netzwerk stellen. Es mag durchaus Datenkommunikation geben, die keine außergewöhnliche Behandlung der Netzwerkpakete erfordert. Ein einfaches Beispiel für diese Situation ist die Kommunikation eines einzelnen Rechners im Netzwerk mit einem DHCP Server mit dem Ziel, eine IP Adresse zu erhalten oder die Laufzeit der erhaltenen IP Adresse zu verlängern. Diese Art von unpriorisiertem Datenverkehr kann ohne eine besondere Kategorisierung übertragen werden und erhält daher den Wert 0000 für das ToS Feld im IP Paketkopf.

2.4 In-Band Management

Unter dem In-Band Management versteht man ein Netzwerkmanagement, bei dem sich die Nutzdaten des Netzwerks mit den Daten für die Überwachung und Konfiguration des Netzwerkes im selben Bereich befinden. Über die Kommunikationswege, mit denen die einzelnen Komponenten untereinander verbunden sind, werden sowohl die eigentlichen Nutzdaten als auch die Managementdaten verschickt. Die Nutzdaten entsprechen den Daten, welche den

Hauptzweck des Netzwerkes bilden – also den zu übermittelnden Informationen. Unabhängig von den Nutzdaten existieren noch die zur Verwaltung des Netzwerks notwendigen Daten und Pakete, wie beispielsweise die SNMP Pakete eines Netzwerkmanagementsystems. Werden diese Managementdaten im selben Netzwerk versendet, das auch die Nutzdaten transportiert, so arbeitet das Netzwerkmanagement „In-Band“[4].

Die Vorteile des In-Band Managements liegen klar auf der Hand. Kann zur Überwachung und Konfiguration der verwalteten Systeme auf die bereits bestehende Infrastruktur zurückgegriffen werden, so fallen für das Netzwerkmanagement keine zusätzlichen Kosten und Installationsarbeiten an. Ein In-Band Management ist vergleichsweise schnell zu einem bestehenden Netzwerk hinzugefügt, da alle notwendigen Verbindungsstrecken bereits existieren. Aber auch ein klarer Nachteil fällt beim In-Band Management sofort auf. Eine Störung des Netzwerkes wirkt sich nicht nur negativ auf das Funktionieren der im Netzwerk vorhandenen Komponenten aus, sondern auch das Netzwerkmanagement selbst wird durch Netzwerkstörungen negativ beeinflusst. Ist ein Kommunikationsweg unterbrochen, so sind die Applikations-Server ebensowenig erreichbar, wie die Netzwerkmanagementsysteme.

Die in Abbildung 2.2 gezeigte schematische Darstellung eines heterogenen Netzwerks sowie die in Abbildung 2.1 gezeigte Darstellung eines homogenen Netzwerks stellen beide gleichzeitig ein Beispiel für ein In-Band Management dar. In beiden Fällen befindet sich die Netzwerkmanagementstation im selben Netzwerk wie die restlichen Komponenten.

2.5 Out-of-Band Management

Beim Out-of-Band Management ist die Infrastruktur des Netzwerkmanagements streng vom eigentlichen Netzwerk und dessen Nutzdaten getrennt. Jedes einzelne am Netzwerk angeschlossene Gerät benötigt daher einen zweiten Anschluss, der ausschließlich für das Netzwerkmanagement vorgesehen ist. Während auf der einen Seite die Systemkomponenten in gewohnter Weise an das Netzwerk angeschlossen sind, werden über diesen Weg keinerlei Managementaufgaben erledigt. Zur Überwachung und Konfiguration der verschiedenen Komponenten im Netzwerk besitzen diese einen separaten Anschluss, der ausschließlich für Managementaufgaben reserviert ist. In diesem Managementnetzwerk befinden sich auch die Netzwerkmanagementstationen. Einen Zugang zum übrigen Netzwerk, das für die Nutzdaten verwendet wird, sowie einen Zugang zur Außenwelt besitzt das Managementnetzwerk in vielen Fällen erst gar nicht. Es kann jedoch vorkommen, dass die Netzwerkmanagementstation einen separaten Zugang für einen entfernten Administrator zur Verfügung stellt. Abbildung 2.5 zeigt beispielhaft die Struktur eines Out-of-Band Managements.

[4] „In-Band“ bedeutet innerhalb desselben „Bandes“ (innerhalb derselben Leitungen und Kommunikationsstrecken), welches die Komponenten verbindet.

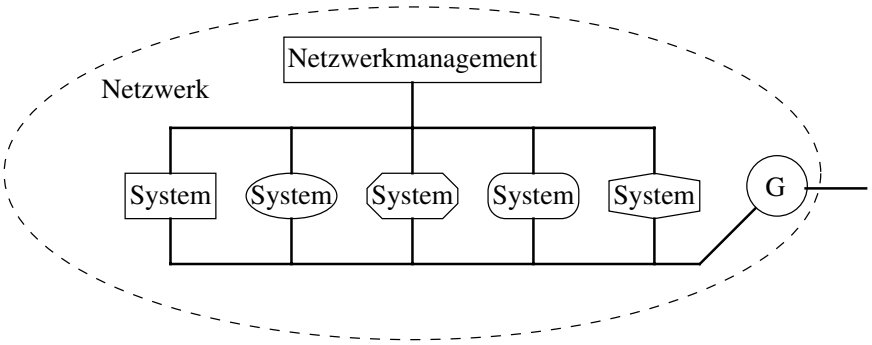

Abb. 2.5. Beim Out-of-Band Management besitzt jedes verwaltete System einen separaten Anschluss zur Erfüllung von Netzwerkmanagement-Aufgaben. Das Managementnetzwerk besitzt außerdem keinen Zugang zum restlichen Netzwerk.

2.5.1 IP Managementnetzwerk

Zur Implementierung eines Out-of-Band Netzwerkmanagements können verschiedene Techniken zum Einsatz kommen. Einerseits kann für das Management ein komplettes dupliziertes IP Netzwerk über beispielsweise Ethernet aufgebaut werden. Auch in diesem Fall gilt es, die Vorteile und Nachteile gegeneinander abzuwägen. Positiv bei einem IP Management Netzwerk anzumerken ist die Verwendung gleicher Techniken wie im eigentlichen Netzwerk und die damit verbundene Ähnlichkeit der Systeme. Auch die angewendeten Techniken und die eingesetzten Protokolle und Mechanismen gleichen denen des Hauptnetzwerks, so dass kein unverhältnismäßig großer, zusätzlicher Wissens- und Informationsbedarf für die mit der Betreuung beauftragten Personen besteht. Die Nachteile lassen sich aber auch nicht verschweigen. Setzt man für das Managementnetzwerk eine vollständige IP Infrastruktur ein, so gelten für dieses Netzwerk mit höheren Protokollen des OSI Referenzmodells ansatzweise ähnliche Anforderungen an die Überwachung und Konfiguration. In einem sehr großen Netzwerk kann das dazu proportionale Managementnetzwerk selbst so groß werden, dass es im Grunde genommen ebenfalls überwacht werden müsste. Ein zweiter großer Nachteil findet sich bei der Sicherheit des Netzwerkmanagements. Zwar ist die Implementierung eines Out-of-Band Managements ein großer Fortschritt gegenüber einem In-Band Management, da hier die sensiblen Daten über separate Verbindungen gesendet werden, die für einen Angreifer von außen zunächst einmal nicht zu erreichen sind. Andererseits ist jedes System des Netzwerks auf zwei unabhängigen Kommunikationswegen mit den anderen Systemen verbunden, so dass sich zur Kontrolle des Datenverkehrs eine doppelte Arbeit ergibt. Wird im Hauptnetzwerk dasselbe Protokoll verwendet, wie im Managementnetzwerk, so besteht auch eine nicht unerhebliche Gefahr für eine Brückenbildung zwischen den beiden aufwändig voneinander isolierten Netzwerken. Wenn lediglich eines der Systeme Pakete vom einen Netzwerk in das andere weiterleitet und somit die Rolle

eines Gateways übernimmt, ist damit gleichzeitig die Sicherheit des Managementnetzwerkes massiv gefährdet. Einen Ausweg bietet dann nur eine ähnlich aufwändige Datenvermittlungskontrolle im Managementnetzwerk wie im Hauptnetzwerk.

2.5.2 IPX Managementnetzwerk

Eine alternative Lösung zur Verwendung eines vollwertigen IP Netzwerks für das Management ist ein Netzwerk auf Basis eines anderen höheren Protokolls. Denkbar ist beispielsweise der Einsatz eines von Novell [143] stammenden Internetwork Packet Exchange (IPX) [144] Netzwerkes. Selbst die Nutzung des Simple Network Management Protocol (SNMP) (siehe Kapitel 4) über das Protokoll IPX ist möglich, wie sich aus RFC 1420 [21] entnehmen lässt. Unter diesen Bedingungen ist die Bildung einer Kommunikationsbrücke zwischen dem Hauptnetzwerk und dem Managementnetzwerk deutlich schwieriger, da zuerst der Protokollbruch überwunden werden muss. In der Praxis ist längst nicht in allen Fällen ein separates Managementnetzwerk auf Basis eines zweiten Protokolls aus der Transportschicht des OSI Referenzmodells sinnvoll. Schließlich verbleibt das Problem des komplexen und damit auch wartungsbedürftigen zweiten Protokolls. Außerdem gesellt sich zu dem bestehenden Problem noch die gestiegene Komplexität. Beim Einsatz von zwei verschiedenen Protokollen müssen auch beide Protokolle gleichermaßen verstanden, implementiert und administriert werden.

2.5.3 Zentralisierte Punkt-zu-Punkt Verbindungen

Ein Weg zur Vermeidung des Overheads durch zwei parallele höhere Protokolle ist der Einsatz einer anderen Technologie für das Managementnetzwerk. Ein beliebter Mechanismus ist die Administration über die Seriellen Schnittstellen der verwalteten Geräte. Mit Konsolen-Servern lassen sich die Einzelverbindungen in einem Gerät zentralisieren und zusammenführen. Die Punkt-zu-Punkt Verbindungen über die RS-232 Schnittstelle werden dazu vom zentralen Konsolen-Server zu den jeweiligen verwalteten Geräten gelegt. Von diesem System aus kann dann auf alle Geräte über einen zweiten Kommunikationsweg zugegriffen werden. Selbst bei Ausfall des Hauptnetzwerks kann vom Konsolen-Server eine Verbindung zu den Netzwerkkomponenten hergestellt werden, um beispielsweise das Problem auf dem Zentral-Switch und somit auch den Ausfall des Netzwerks zu beseitigen. Zur bequemen Administration ist der Konsolen-Server selbst ebenfalls per Netzwerk erreichbar – oder eben auch nicht, wenn das Netzwerk ausgefallen ist! Man sollte sich im Klaren sein, dass bei größeren Problemen im Netzwerk der Konsolen-Server ebenfalls nicht mehr erreichbar sein kann. Zur Umgehung dieser Problematik kann man den Konsolen-Server mit einem separaten und unabhängigen Zugang ausstatten, so dass man sich im Ernstfall zur Durchführung der notwendigen Managementaufgaben auch noch mit dem System verbinden kann.

Statt der seriellen Verbindung zu den verwalteten Geräten kann bei manchen Servern mit graphischer Managementoberfläche der direkte Zugang zu den Eingabe- und Ausgabeschnittstellen erforderlich sein. In diesem Fall wird das serielle Kabel durch Anschlüsse an den Tastatur-, Maus- und Video-Schnittstellen ersetzt. Auch hier lassen sich die Anschlüsse zentralisieren und an Keyboard-Video-Mouse (KVM) Switches zusammenführen. Die für die Konsolen-Server gemachten Aussagen treffen im Wesentlichen auch für die KVM Switches zu. Fällt das Netzwerk aus, über das auch auf den KVM Switch zugegriffen wird, so fällt gleichzeitig auch die alternative Zugangsmöglichkeit zu den verwalteten Geräten weg. Als Lösung bietet sich auch hier ein redundanter separater Zugang unabhängig vom Hauptnetzwerk und dem Managementnetzwerk an.

Ein kleiner Nachteil bei den beiden letzten Varianten des Out-of-Band Managements lässt sich jedoch nicht verschweigen. Durch die Reduktion der Kommunikationsform auf einfache Schnittstellen und Protokolle ist gleichzeitig eine ausführliche und detaillierte Überwachung und Konfiguration der Systeme nur schwer möglich. Komplexe Managementsysteme wie das SNMP Rahmenwerk können über Konsolen-Server oder KVM Switches nur mit separatem Aufwand eingesetzt werden.

Aber auch der Vorteil bezüglich der Sicherheit eines über Konsolen-Server oder KVM Switches administrierten Netzwerks soll an dieser Stelle nicht vergessen werden. Der Zugriff auf die verwalteten Geräte über Punkt-zu-Punkt Verbindungen bietet nur sehr wenig Angriffspunkte. Die Kommunikationswege zwischen den verwalteten Geräten und dem zentralen Managementsystem sind dediziert und nur mit physikalischem Zugriff auf die Komponenten möglich. Der einzige Ansatzpunkt für eine Kompromittierung des Netzwerks sind der Konsolen-Server oder der KVM Switch selbst. Zwar besteht aus diesem Grund auch ein erhöhter Schutzbedarf für diese zentralen Komponenten, allerdings lässt sich ein einzelner Punkt auch wesentlich besser absichern, als ein komplexes Netzwerk. Wer zusätzlich zur Sicherheit auch noch eine hohe Verfügbarkeit des Managementnetzwerks benötigt, kann die zentralen Managementsysteme und beim Konsolen-Server sogar die seriellen Verbindungen redundant auslegen.

2.5.4 Management über IPMI

Eine noch weiterführende Alternative sind Managementsysteme, welche die verwalteten Geräte über das Intelligent Platform Management Interface (IPMI) [95] administrieren (siehe auch Kapitel 6). Diese Schnittstelle bietet neben dem separaten Zugangsweg gleichzeitig noch weiterführende Mechanismen an, um auch bei einem kompletten Absturz des Systems die Arbeitsfähigkeit wiederherstellen zu können. Aus diesem Grund befindet sich nach der Definition von IPMI eine redundante Hardware im verwalteten Gerät, die mit einem eigenen Betriebssystem arbeitet und die restliche Hardware und Software des Systems überwachen kann. Falls das Gerät einmal durch einen Absturz des

Betriebssystems nicht mehr erreichbar wird, so kann auf herkömmlichen Wegen das Gerät nicht mehr in Gang gesetzt werden. Sofern kein Schaden der Hardware vorliegt, kann das System in vielen Fällen die Arbeitsfähigkeit durch eine Neuinitialisierung wiederhergestellt werden. Ohne IPMI würde der Administrator manuell am System den „Reset"-Knopf betätigen, um das System neu zu starten. Bei einer IPMI Implementierung ist selbst nach komplettem Absturz des Betriebssystems im System der redundante Hardware-Teil mit dem separaten Betriebssystem noch erreichbar. Der Administrator kann nun aus der Ferne Kontakt mit dem Managementsystem aufnehmen und einen Neustart des Systems initiieren. Die Arbeitsfähigkeit kann also ohne physikalischen Zugriff auf das ausgefallene System wiederhergestellt werden.

Der IPMI Mechanismus ist nicht als reiner Managementmechanismus sondern eher als ein paralleles Managementsystem zur Erweiterung der Möglichkeiten der bereits vorhandenen Netzwerkmanagement-Schnittstellen zu betrachten. Neuere Trends bei IPMI erlauben allerdings neben der reinen Kontrolle über die Hardware auch den Zugriff auf eine Konsole zur Administration. Damit wird IPMI als alternative Managementschnittstelle zur Verfügung stehen, sobald die neuesten Standards implementiert werden.

2.6 Kombinierte Managementlösungen

In vielen Fällen ist die Umsetzung der hier vorgestellten Kategorien des Netzwerkmanagements in ihrer reinsten Form nicht möglich. Fast immer finden sich Teilbereiche, die aus homogenen Komponenten bestehen. Ein Beispiel dafür sind die in vielen Fällen vom selben Hersteller eingesetzten Geräte zum Betrieb des Netzwerk-Backbones. Aber in beinahe allen Netzwerken finden sich auch heterogen zusammengesetzte Systeme, die nicht zuletzt aus den gewachsenen Strukturen resultieren. Ähnlich verhält es sich mit dem Einsatz von In-Band und Out-of-Band Managementstrategien. Für welche der beiden Strategien man sich letztendlich auch entschieden hat, finden sich in vielen Fällen dennoch Ausnahmen zur Regel. Ein einfaches Beispiel kann ein Zugangsrouter in einem ansonsten In-Band verwalteten Netzwerk sein, der über einen Notfallzugang über ein separates Modem und eine eigene Telefonverbindung verfügt. Andererseits verfügen nicht alle Komponenten im Netzwerk über eine Schnittstelle zur Errichtung eines separaten und redundanten Zugangs für das Netzwerkmanagement. So können in einem ansonsten Out-of-Band verwalteten Netzwerk dennoch einige Systeme In-Band verwaltet werden, und wenn es nur die Access-Points eines Wireless Local Area Network (WLAN) sind.

Gerade die Ausnahmen bei der Einhaltung der verwendeten Kategorien des Netzwerkmanagements bilden gleichzeitig auch die großen Gefahren für die Sicherheit. Eine große Vielfalt an Systemen, die zum Teil nur auf äußerst unterschiedlichen Wegen administriert werden können oder sollen, erfordert ein flexibles und manchmal auch dynamisches Netzwerkmanagementsystem.

Als häufigste Ursache für Sicherheitsprobleme können verschiedene Punkte angeführt werden

2.6.1 Komplexität

Unabhängig vom Netzwerkmanagement lässt sich sagen, dass mit steigender Komplexität eines Systems die Menschen auch größere Probleme bei der Übersicht und dem Detailverständnis dieses Systems haben. Je komplexer ein Netzwerkmanagementsystem also ist, desto größer ist auch die Gefahr, dass die Administratoren den Überblick über das System verlieren; zumindest steigt die Wahrscheinlichkeit, dass unter der Vielzahl von Systemen, Mechanismen, Protokollen und Besonderheiten im Netzwerk auch das tiefergehende Verständnis aller Details leidet. Im selben Maße steigt auch die Wahrscheinlichkeit, dass Fehler bei der Administration gemacht werden. Erschwerend kommt hinzu, dass Fehler mit Auswirkungen auf die Funktionalitäten des Netzwerkes im Normalfall sehr schnell erkannt werden, nicht aber Fehler mit Auswirkungen auf nicht-funktionale Anforderungen. Der unterbrochene Zugang zu einem Produktivsystem zieht vergleichsweise schnell die Aufmerksamkeit auf sich; ein nicht entferntes Standard-Passwort auf einem Managementsystem kann allerdings längere Zeit unbemerkt bleiben.

2.6.2 Flexibilität

Damit ein Netzwerkmanagementsystem mit den unterschiedlichsten Gegebenheiten im Netzwerk umgehen kann, sollte es möglichst flexibel ausgelegt sein. Diese grundsätzlich positive Eigenschaft kann aber auch negative Auswirkungen auf die Sicherheit des Netzwerkmanagements haben. Dies begründet sich auf die oftmals gegenläufig ausgerichteten Ziele der beiden Anforderungen Funktionalität und Sicherheit. Während aus funktionalen Gesichtspunkten möglichst viele Wege zur Verfügung stehen sollten, die bei der Erfüllung einer Aufgabe helfen können, sollten aus dem Blickwinkel der Sicherheit möglichst wenige Alternativen bestehen. Schließlich muss für jeden Weg explizit der Zugriff auf genau diejenigen Funktionen eingeschränkt werden, die zur Erfüllung der Aufgaben absolut notwendig sind.

Flexibilität ist keine schlechte Eigenschaft eines Netzwerkmanagementsystems; man sollte nur eine wichtige Maxime berücksichtigen:

So wenig wie möglich, so viel wie nötig!

2.6.3 Dynamik

Die Flexibilität des Netzwerkmanagementsystems bezieht sich auf die verschiedenen unterstützten Funktionalitäten. Demgegenüber bildet die Dynamik dazu gleichzeitig eine orthogonale Ebene der Zeit. Selbst wenn nur wenige Funktionalitäten vom Netzwerkmanagementsystem zur Verfügung gestellt werden, so unterliegen Netzwerke und damit auch deren Management

einem ständigen Wandel. Die unvermeidliche Dynamik hat aber auch ihre Auswirkungen auf die Sicherheit des Netzwerkmanagements. Ändern sich die Anforderungen oder Aufgaben der Systeme des Netzwerks, so müssen diese Änderungen auch im Netzwerkmanagement abgebildet werden. Nicht selten kommt es dann vor, dass die Deaktivierung von nicht mehr vorgesehenen Kommunikationswegen aus Versehen vergessen wird. Manchmal müssen auch Funktionalitäten für einen begrenzten Zeitraum freigeschaltet werden, die eine Ausnahme zu den bestehenden Sicherheitsrichtlinien darstellen. Auch in diesem Fall kann das anschließende Deaktivieren der temporär eingerichteten Mechanismen in Vergessenheit geraten und so für Sicherheitsprobleme sorgen. Gut ausgearbeitete organisatorische Schritte können in vielen Fällen das Risiko zwar mindern, aber dennoch nicht gänzlich verhindern.

3

ICMP: Netzwerkmanagement auf unterer Ebene

3.1 Ursprünge des Protokolls ICMP

Das Internet Control Message Protocol (ICMP) [160] stammt aus derselben Zeit wie das Internet Protocol (IP) [161] und ist nun beinahe ein Vierteljahrhundert alt. Zur damaligen Zeit existierte das Internet noch längst nicht in seiner heutigen Form. Das aus dem militärischen Bereich resultierende Netzwerk trug 1981 den Namen „Catenet“ [39] und bestand im Wesentlichen aus mehreren Netzwerken, die über „Gateways“ miteinander verbunden waren. Das im Catenet vorherrschende Protokoll war IP, das bis heute in seiner Grunddefinition verbindungslos und unzuverlässig ist[1]. Damit dennoch eine zuverlässige Kommunikation möglich ist, müssen andere Protokolle verwendet werden, die auf dem Internet Protokoll aufbauen können. Das bekannteste Beispiel ist das verbindungsorientierte Transmission Control Protocol (TCP) [162]. Ebenfalls aus dem militärischen Bereich stammend, sollte mit dem Protokoll TCP ganz klar ein robustes und hochgradig zuverlässiges Protokoll geschaffen werden, das die strategische Kommunikation zwischen Computern auch unter widrigsten Bedingungen weitestgehend ermöglicht. Wie sich gezeigt hat, kann die TCP/IP Protokollfamilie auch heute noch alle damals gestellten Anforderungen sehr gut erfüllen.[2]

Zur Verbindung zwischen den verschiedenen Netzwerken standen im Catenet – und stehen auch noch heute – die vielen Gateways. Zum Informati-

[1] Der Begriff ‚unzuverlässig‘ soll hier in rein technischem Zusammenhang verstanden werden. Millionen von Anwendern des Internets würden vermutlich bestätigen, dass das Internet zu weiten Teilen durchaus zuverlässig erscheint.

[2] Die heutigen Kritikpunkte an der TCP/IP Protokollfamilie – wie beispielsweise unzureichende Sicherheitsmechanismen oder ein zu kleiner Adressraum – beruhen im Wesentlichen auf geänderten Anforderungen. Man kann also höchstens den damaligen Entwicklern vorwerfen, dass sie die Anforderungen nicht gut genug spezifiziert haben. Man sollte aber dabei nicht vergessen, dass heute das Internet und die Protokollfamilie TCP/IP außerhalb des damals geplanten Anwendungsbereiches zum Einsatz kommen.

onsaustausch verwendeten die Gateways des Catenet untereinander das dynamische Routingprotokoll Gateway-to-Gateway Protocol (GGP) [85]. Die Aufgabe der Gateways bestand darin, über das Protokoll GGP Routinginformationen über bekannte Netzwerke mit anderen Gateways auszutauschen, so dass von ihnen die IP Pakete über den kürzesten Weg zum Ziel vermittelt werden konnten. Im Gegensatz zu den Protokollen IP und TCP gilt das Routingprotokoll GGP heute als veraltet und ist durch verschiedene andere Routingprotokolle ersetzt worden [82, 112, 133, 134, 27].

Während TCP eine Verbindung zwischen zwei Endgeräten in einem IP-Netzwerk realisiert und GGP eine Kommunikationsmöglichkeit für die Vermittlungsstellen untereinander darstellt, füllt das Protokoll ICMP die Lücke des Informationsaustausches zwischen den Gateways und den Endgeräten des Netzwerks. ICMP ist auf einer niedrigeren Ebene des OSI Referenzmodells als TCP angesiedelt und dient wie das GGP Routingprotokoll hauptsächlich der Sicherstellung einer reibungslosen und zuverlässigen Kommunikation im Netzwerk. In einigen Fällen können ICMP Nachrichten auch zwischen Endgeräten ausgetauscht werden, allerdings beziehen sich die ausgetauschten Informationen nur auf den reibungslosen Austausch von Paketen und nicht auf deren Inhalte. Die in der ursprünglichen Request for Comments RFC 792 spezifizierten Nachrichtentypen von ICMP beschränken sich deshalb auch auf einige wenige Meldungsarten zur Steuerung des Paketflusses. In RFC 950 [132] und RFC 1256 [55] wurden später noch vier weitere ICMP Nachrichtentypen hinzugefügt. Schließlich findet sich in RFC 1393 [111] noch eine interessante Ergänzung für das ICMP Protokoll, die allerdings in der Praxis nur eine kleine Rolle spielt. Tabelle 9.2 aus Kapitel 9 liefert einen Überblick über die verschiedenen Nachrichtentypen und Unterkategorien. Wie die Protokolle IP und TCP spielt ICMP auch heute noch eine wichtige Rolle in IP Netzwerken. Im weitesten Sinne kann demnach ICMP als ein aktuelles Netzwerkmanagementprotokoll aufgefasst werden.

3.2 Paketformat

Das Protokoll ICMP ist Teil der TCP/IP Protokollfamilie und ist somit wie TCP ein Unterprotokoll des Internet Protokolls IP. Nach RFC 1700[3] [165] hat ICMP die Protokollnummer 1 erhalten (siehe auch Tabelle 3.1 mit einer Auflistung weiterer aktueller IP Protokolle).

Jedes ICMP Paket ist demnach in ein IP Paket eingebettet. Zur Verdeutlichung soll an dieser Stelle ein kurzer Überblick über den IP Paketkopf („IP Header“) gegeben werden. Jeder IP Paketkopf besteht aus mindestens 20 Oktetten, welche durch die optionale Angabe von weiteren Optionen auf maximal

[3] RFC 1700 ist heute durch die RFC 3232 ersetzt worden, in der lediglich darauf verwiesen wird, dass die verwalteten Nummern im Internet Protokoll nun unter der Schirmherrschaft der Internet Assigned Numbers Authority (IANA) stehen und über das Internet erreichbar sind (siehe `http://www.iana.org`).

Tabelle 3.1. Auswahl von Protokollen unterhalb des Internet Protokolls IP

Nummer	Protokoll	Referenz
1	Internet Control Message Protocol (ICMP)	RFC 792 [160]
2	Internet Group Management Protocol (IGMP)	RFC 1112 [54]
3	Gateway-to-Gateway Protocol (GGP)	RFC 823 [85]
4	IP-in-IP Encapsulation	RFC 2002 [152]
5	Internet Stream Protocol Version 2 (ST2)	RFC 1819 [57]
6	Transmission Control Protocol (TCP)	RFC 793 [162]
8	Exterior Gateway Protocol (EGP)	RFC 904 [130]
17	User Datagram Protocol (UDP)	RFC 768 [159]
27	Reliable Data Protocol (RDP)	RFC 1151 [150]
28	Internet Reliable Transaction Protocol (IRTP)	RFC 938 [129]

60 Oktette erweitert werden können. Tabelle 3.2 listet die Minimalangaben im IP Paketkopf auf.

Tabelle 3.2. Angaben im Paketkopf des Protokolls IP mit ihrer jeweiligen Länge in Bits und der genauen Position

Länge [Bit]	Oktett-Nr.	Angabe	Bedeutung
4	0	Version	Version des verwendeten IP Protokolls. Da nachfolgende Versionen von IP, wie beispielsweise IPv6 [56], eine andere Paketkopf-Struktur aufweisen, existiert im Wesentlichen nur ein gängiger Wert für die Version: 4.
4	0	IHL	Länge des Paketkopfes in 32-Bit Worten. Ein korrekter IP Paketkopf enthält mindestens die in dieser Tabelle aufgeführten Angaben und muss daher auch an dieser Stelle einen Wert von mindestens 5 aufweisen. Größere Werte sind bei zusätzlich im Paketkopf definierten Optionen möglich. Ist die Summe aller durch Optionen hinzugefügten Bits nicht durch 32 dividierbar, so muss der Paketkopf bis zum nächsten Vielfachen von 32 Bit aufgefüllt werden.

(Fortsetzung auf nächster Seite)

Tabelle 3.2. Angaben im Paketkopf des Protokolls IP mit ihrer jeweiligen Länge in Bits und der genauen Position (Fortsetzung)

Länge [Bit]	Oktett-Nr.	Angabe	Bedeutung
8	1	ToS	Über das Type of Service (ToS) Feld können jeweils getrennte Qualitätsangaben zur gewünschten maximalen Verzögerung, zur gewünschten Zuverlässigkeit und zum gewünschten minimalen Durchsatz für das Paket gemacht werden.
16	2-3	Länge	Länge des gesamten Pakets inklusive Paketkopf und Nutzdaten gemessen in Oktetten.
16	4-5	ID	Eindeutige Identifikationsnummer des Pakets, die beim korrekten Zusammenbau einzelner Fragmente helfen soll.
4	6	Flags	Flags zur Fragmentierung des Paketes. Neben dem „Don't Fragment" Flag, welches die Fragmentierung des Pakets verhindert, existiert noch ein zweites Flag, welches bei Fragmentierung einen Hinweis darauf gibt, ob es sich um das letzte Fragment eines Paketes handelt.
12	6-7	Fragment	Stelle im Gesamtdatagramm, an welcher der Inhalt dieses Fragments beim Zusammenbau eingefügt werden muss.
8	8	TTL	Das „Time-to-Live" (TTL) Feld gibt die maximale Lebensdauer des Paketes in Hops an. Ursprünglich war die Angabe in der Einheit Sekunden definiert. Da aber jedes Gateway beim Weiterleiten des Paketes unabhängig von der dafür verbrauchten Zeit das TTL Feld um Eins erniedrigen muss, verhält sich der TTL Wert bei den hohen Vermittlungsgeschwindigkeiten in der Praxis wie die Angabe der maximalen Anzahl von Gateways für den Übertragungsweg.
8	9	Protokoll	In den meisten Fällen werden innerhalb von IP Paketen höhere Protokolle des OSI Referenzmodells eingebettet. Tabelle 3.1 listet eine kleine Auswahl der zur Verfügung stehenden Protokolle auf.

(Fortsetzung auf nächster Seite)

Tabelle 3.2. Angaben im Paketkopf des Protokolls IP mit ihrer jeweiligen Länge in Bits und der genauen Position (Fortsetzung)

Länge [Bit]	Oktett-Nr.	Angabe	Bedeutung
16	10-11	Prüfsumme	Prüfsumme des IP Paketkopfes. Zur Berechnung der Prüfsumme wird zunächst das Feld für die Prüfsumme selbst auf Null gesetzt. Anschließend werden für alle 16-Bit Worte im Paketkopf die Einerkomplemente gebildet und aufsummiert. Die Prüfsumme ergibt sich schließlich aus dem Einerkomplement dieser Summe.
32	12-15	Quelle	IP Adresse des Absenders.
32	16-19	Ziel	IP Adresse des Empfängers.

Da ICMP Nachrichten im Normalfall keinerlei besondere Optionen benötigen, kommt man mit einem IP Paketkopf von 20 Oktetten aus. Zu diesen Angaben im Kopf des IP Paketes kommen dann noch die Nutzdaten hinzu. Im Falle von ICMP sind dies die Nachrichten in ihren verschiedenen Formaten mit unterschiedlicher und zum Teil variabler Länge.

3.3 Nachrichtentypen

Die in RFC 792 ursprünglich definierten Nachrichten beschränken sich auf elf verschiedene Typen, die teilweise noch in bis zu sechs Unterkategorien (Code) differenziert werden. Mittlerweile wurden die Nachrichtentypen und Unterkategorien weiter ergänzt und verfeinert (siehe Tabelle 9.2). Jeder Nachrichtentyp erfordert andere Parameter, so dass kein einheitliches ICMP Paketformat existiert. Die ersten vier Oktette sind für alle ICMP Nachrichten identisch. Sie enthalten jeweils ein Oktett mit Angaben zum Typ und der optionalen Unterkategorie der Nachricht sowie zwei Oktette mit einer Prüfsumme über das vollständige ICMP Paket. Alle anderen noch folgenden Oktette hängen stark vom verwendeten Nachrichtentyp ab. Während beispielsweise *Information Request* und *Information Reply* Nachrichten lediglich jeweils zwei Oktette für eine Identifikationsnummer und eine Sequenznummer implementieren, müssen zu den *Timestamp* und *Timestamp Reply* Nachrichten noch drei Zeitangaben mit jeweils vier Oktetten hinzugefügt werden. Im Folgenden sollen daher die elf Grundtypen für ICMP Nachrichten sowie die in RFC 950, RFC 1256 und RFC 1393 spezifizierten fünf Ergänzungen dazu näher beschrieben werden.

3.3.1 *Echo Reply (Typ 0)*

Zu den wichtigsten ICMP Nachrichtentypen zählen die beiden Typen *Echo* und *Echo Reply*. Das wohl bekannteste Werkzeug, das diese beiden Nach-

richtentypen implementiert, ist das Packet Internetwork Groper (PING) Kommando, mit dem *Echo* Nachrichten versendet und die passenden *Echo Reply* Antworten empfangen und ausgewertet werden können. Zu den drei Standardangaben Typ, Unterkategorie und Prüfsumme enthält eine ICMP *Echo* Nachricht noch eine Identifikationsnummer und eine Sequenznummer. Im Anschluss daran befindet sich ein variabler Datenbereich, der beliebige Informationen enthalten kann. Eine *Echo Reply* Nachricht als Antwort auf eine ICMP *Echo* Nachricht muss diesen Datenbereich unverändert wiederholen und an den Absender zurückschicken. Abbildung 3.1 zeigt noch einmal schematisch den Aufbau eines *Echo Reply* ICMP Paketes.

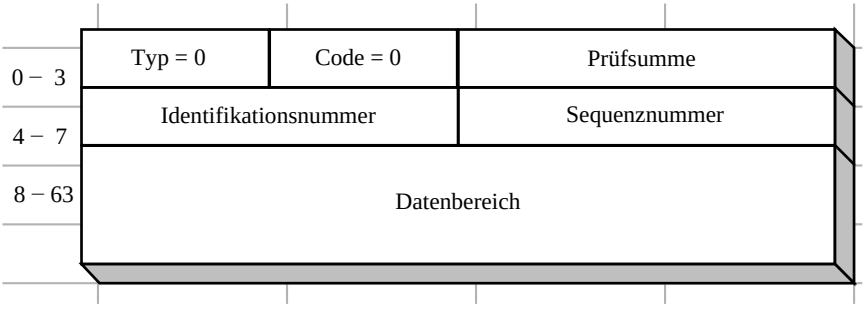

Abb. 3.1. Schematische Darstellung eines ICMP *Echo Reply* Paketes. Die Länge des Datenbereiches ist prinzipiell variabel, wird aber in der Praxis häufig auf 56 Oktette beschränkt.

3.3.2 *Destination Unreachable (Typ 3)*

Der ICMP Nachrichtentyp *Destination Unreachable* kann in sechs verschiedenen Varianten auftreten, die eine Aussage über den Grund der Nichterreichbarkeit eines angesprochenen Netzwerkgerätes machen. Abbildung 3.2 zeigt den schematischen Aufbau der verschiedenen *Destination Unreachable* Nachrichtenvarianten.

net unreachable (Typ 3, Code 0)

Ein Gateway in einem Netzwerk kann einem Endgerät über eine *net unreachable* Nachricht mitteilen, dass es keinen Weg zu dem Netzwerk kennt, in dem sich das Ziel des Pakets vom Endgerät befindet. In vielen Fällen ist in Gateways ein Standard-Gateway konfiguriert, so dass alle Pakete von nicht näher bekannten Netzwerken zu diesem Gerät weitergeleitet werden. Erst beim Fehlen oder bei Ausfall des Standard-Gateways sind die entsprechenden Netzwerke nicht mehr erreichbar und liefern die ICMP Nachricht *net unreachable*.

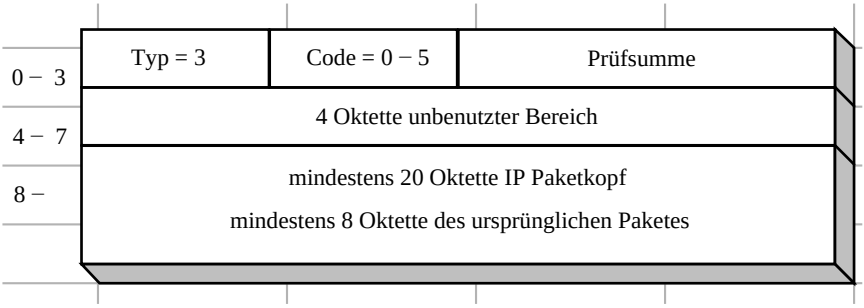

Abb. 3.2. Schematische Darstellung eines ICMP *Destination Unreachable* Paketes. Die Länge des Datenbereiches ist variabel. Enthalten ist der originale IP Paketkopf der eingegangenen Nachricht, der mindestens 20 Oktette umfasst, sowie mindestens die ersten 8 Oktette des ursprünglichen Datenbereiches des eingegangenen Paketes. Ist die *Destination Unreachable* Nachricht eine Antwort auf ein Paket eines höheren Protokolls, so stehen hier beispielsweise Angaben zu den verwendeten Ports, was eine Zuordnung zum Originalpaket vereinfacht.

host unreachable (Typ 3, Code 1)

In einigen Fällen kann ein Gateway nicht nur erkennen, dass das Netzwerk des vom Endgerät angesprochenen Systems nicht erreichbar ist. Ist das Gateway in der Lage, die Nichterreichbarkeit des Zielsystems festzustellen, so kann es auch die etwas konkretere ICMP Nachricht *host unreachable* senden.

protocol unreachable (Typ 3, Code 2)

Kann ein Gateway ein Paket nicht ausliefern, weil das Zielsystem nicht das verwendete Protokoll unterstützt, so kann das Gateway als Fehlermeldung an das ursprünglich sendende Endgerät eine *protocol unreachable* ICMP Nachricht senden.

port unreachable (Typ 3, Code 3)

Kann ein Gateway ein Paket nicht ausliefern, weil das Zielsystem zwar das verwendete Protokoll unterstützt, auf dem angegebenen Port jedoch kein Prozess Pakete entgegennimmt, so kann das Gateway als Fehlermeldung an den ursprünglichen Absender eine *port unreachable* ICMP Nachricht senden.

fragmentation needed with „Don't fragment" (DF) Flag set (Typ 3, Code 4)

Netzwerkgeräte unterstützen je nach verwendetem Protokoll der Sicherungsschicht des OSI Referenzmodells Datagramme mit einer unterschiedlichen maximalen Länge. Diese als Maximum Transfer Unit (MTU) bekannte Größe bestimmt, wie groß ein Paket werden darf, bevor es fragmentiert und in mehrere

Teile zerlegt werden muss. Das Protokoll IP unterstützt ein Fragmentierungs-Flag, über welches angegeben werden kann, ob das zu sendende Paket fragmentiert werden darf. Ist dieses „Don't fragment" (DF) Flag gesetzt und die Größe des zu generierenden ICMP Paketes überschreitet die MTU, so dass das Paket fragmentiert werden müsste, so sendet das Gateway zum Anzeigen dieses Fehlers die ICMP Nachricht *fragmentation needed.*

source route failed (Typ 3, Code 5)

Das Prinzip des „Source Routing" umgeht die unberechenbare Dynamik von IP Netzwerken, in dem für die Vermittlung eines Paketes auch ein expliziter Pfad über alle Gateways in der zu nehmenden Reihenfolge spezifiziert ist. Ein Gateway im Internet muss also bei der Weiterleitung eines quellvermittelten Paketes dieses an das in der Liste nachfolgende Gateway senden. Kann aus irgendeinem Grund diese Vermittlung nicht stattfinden, beispielsweise weil eine direkte Verbindung zwischen zwei angegebenen Gateways gar nicht oder temporär nicht existiert, so muss das Gateway das Paket verwerfen und dem Absender als Fehlermeldung die ICMP Nachricht *source route failed* zurückschicken. Dies gilt sogar dann, wenn das Gateway einen alternativen Weg für das Paket kennt.

3.3.3 *Source Quench (Typ 4)*

Kann in einem Netzwerk der Empfänger die eingehenden Pakete nicht mit derselben Geschwindigkeit verarbeiten, wie sie der Absender liefern kann, so kann in ungünstigen Fällen der Eingangspuffer des Empfängers überlaufen, so dass er keine weitere Nachrichten mehr entgegennehmen kann. Damit in diesem Fall die gesendeten Pakete nicht verloren gehen, können überlastete Gateways oder Endgeräte eine ICMP Nachricht vom Typ *Source Quench* an den ursprünglichen Absender verschicken. Empfängt ein Sender einen „Source Quench"[4], so muss er die Rate verringern, mit der er im Folgenden seine Pakete an den Empfänger verschickt. Dabei ist es für den Sender zunächst völlig unwichtig, welches konkrete Gerät auf dem Übermittlungsweg den Flaschenhals darstellt. In vielen Fällen fällt der Sender nach Erhalt einer *Source Quench* Nachricht in eine minimale Übertragungsrate zurück, die ein weiteres Überlaufen des Empfangspuffers verhindern soll. Anschließend kann der Sender versuchen, seine Übertragungsrate sukzessive zu erhöhen, ohne dabei erneut *Source Quench* Nachrichten zu erzeugen. Abbildung 3.3 veranschaulicht den internen Aufbau eines ICMP Paketes vom Typ *Source Quench.*

[4] Eine eindeutige Übersetzung für den Begriff „Source Quench" zu finden, ist sehr schwierig. Obwohl der Begriff „Quench" häufig eine bildliche Beschreibung für das Stillen des Durstes liefert, erscheint die Bedeutung des Unterdrückens eines Aufruhrs an der Quelle (Source) eine bessere Beschreibung des Sachverhaltes darzustellen.

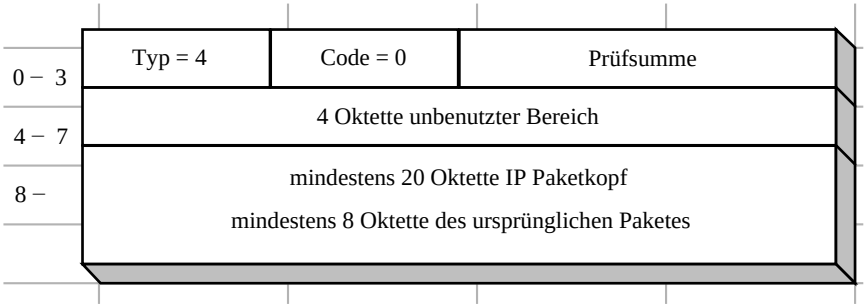

Abb. 3.3. Schematische Darstellung eines ICMP *Source Quench* Paketes. Die Länge des Datenbereiches ist variabel. Enthalten ist der originale IP Paketkopf der eingegangenen Nachricht, der mindestens 20 Oktette umfasst, sowie mindestens die ersten 8 Oktette des ursprünglichen Datenbereiches des eingegangenen Paketes.

3.3.4 *Redirect (Typ 5)*

Erhält ein Gateway ein Paket von einem Endgerät zur Weitervermittlung an ein Ziel, so besteht die Aufgabe des Gateways unter normalen Bedingungen darin, das Paket entweder direkt an das Ziel oder an das nächste Gateway auf dem Weg zum Ziel weiterzuleiten. Unter bestimmten Bedingungen kann dies zwar technisch möglich jedoch ineffizient und unerwünscht sein. Gemeint sind die Spezialfälle, in denen das Ziel auf einem kürzeren Weg hätte erreicht werden können. Befinden sich beispielsweise zwei Gateways G_1 und G_2 im selben Netzwerk wie der Sender, und das Ziel befindet sich hinter Gateway G_2, so sollten auch alle Pakete vom Sender an das Gateway G_2 zur Weiterleitung an das Ziel verschickt werden, da dies der kürzeste – und im speziellen Fall auch einzige – Weg zum Ziel ist (siehe Abbildung 3.4). Adressiert der Sender seine Pakete nun fälschlicherweise an das Gateway G_1, so wird dieses zunächst feststellen, dass Gateway G_2 als nächstes auf dem Weg zum Ziel liegt. Gleichzeitig kann Gateway G_1 anhand der IP Adressen aber auch erkennen, dass sich der Sender und das Gateway G_2 im selben Netzwerk befinden. Der kürzeste Weg für das Paket wäre also vom Sender direkt zum Gateway G_2 gewesen und nicht über den Umweg über Gateway G_1. Damit eine möglichst effektive Paketvermittlung stattfinden kann, hat das Gateway G_1 nun die Möglichkeit, den Absender über den kürzeren Weg zum Ziel zu informieren, der direkt über das Gateway G_2 verläuft. Der Sender sollte nun derart auf ein *Redirect* ICMP Paket reagieren, dass er die an das Ziel gerichteten Pakete im Anschluss nicht mehr über den Umweg sondern direkt an das optimale Gateway G_2 verschickt.

Eine Ausnahme zu dieser Situationen bilden quellvermittelte Pakete, die dem „Source Route“ Mechanismus unterliegen. Bei diesen Paketen ist der einzuschlagende Weg im Netzwerk mit allen beteiligten Gateways im Vorfeld bereits festgelegt. Solange ein Gateway seiner Aufgabe nachkommen kann und das Paket an das nächste spezifizierte Gateway ausliefern kann, wird es dies auch tun – selbst wenn dies aus Sicht des Gateways ineffizient erscheint.

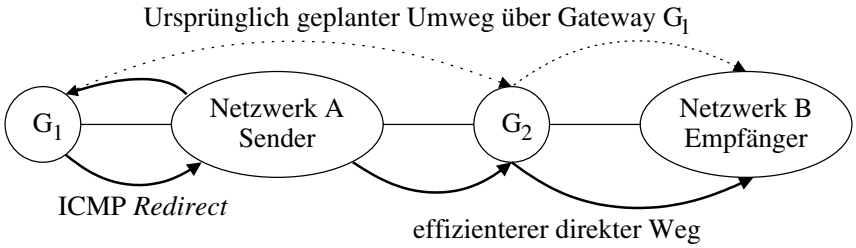

Abb. 3.4. Beim typischen Anwendungsfall einer *Redirect* Nachricht des Protokolls ICMP wird ein Paket zu einem Gateway G_1 zur Weitervermittlung an das Ziel geschickt, obwohl es einen kürzeren und direkteren Weg gibt. Die Nachrichten werden effizienter ausgeliefert, wenn sie direkt an das Gateway G_2 gesendet werden, ohne den Umweg über G_1 nehmen zu müssen.

In Abbildung 3.5 ist der schematische Aufbau von *Redirect* Nachrichten abgebildet, die nahezu identisch zu den meisten anderen ICMP Nachrichtentypen sind.

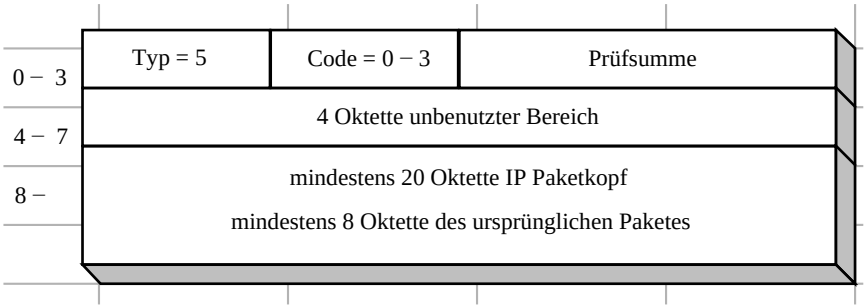

Abb. 3.5. Schematische Darstellung eines ICMP *Redirect* Paketes. Die Länge des Datenbereiches ist variabel. Enthalten ist der originale IP Paketkopf der eingegangenen Nachricht, der mindestens 20 Oktette umfasst, sowie mindestens die ersten 8 Oktette des ursprünglichen Datenbereiches des eingegangenen Paketes.

redirect for the network (Typ 5, Code 0)

Prinzipiell könnte ein Gateway für fast alle *redirect for the host* Nachrichten auch eine *redirect for the network* ICMP Nachricht generieren, da diese Routinginformationen zum Zielsystem im Normalfall auch für das gesamte Netzwerk des Zielsystems gelten. Der Sender könnte somit für zukünftige Pakete, die zu anderen Geräten im selben Zielnetzwerk gerichtet sind, sofort das effizienteste Gateway festlegen. Der Gewinn an Performanz hat sich allerdings als zu gering herausgestellt, als dass sich *Redirect* Nachrichten für ganze

Netzwerke nennenswert auszahlen würden. Außerdem bereitet insbesondere die Unterteilung von Netzwerken in weitere Subnetze größere Probleme bei der Ermittlung der umzuleitenden Netzwerkadressen. In der Praxis werden *redirect for the network* Nachrichten daher nicht unterstützt. Eine ausführliche Diskussion zu diesem Thema findet sich in RFC 1812 [8].

redirect for the host (Typ 5, Code 1)

In den meisten Fällen wird eine *Redirect* Nachricht notwendig, weil sich das Zielsystem in einem Netzwerk befindet, das über einen kürzeren Weg erreichbar ist. Dieses Beispiel ist auch in Abbildung 3.4 dargestellt und wird vom Gateway mit einer *redirect for the host* ICMP Nachricht quittiert.

redirect for the type of service and the network (Typ 5, Code 2)

Für die spezielle Unterkategorie *redirect for the type of service and the network* der ICMP Nachricht vom Typ *Redirect* gilt im Wesentlichen dasselbe, wie für die Unterkategorie *redirect for the network*. Wie die Diskussion in RFC 1812 deutlich macht, bereitet eine Umleitung ganzer Netzwerke insbesondere bei der Einrichtung von Subnetzen größere Probleme, die vollständig unabhängig von der konkreten Art der Pakete entstehen. Daher werden auch *redirect for the type of service and the network* Nachrichten in der Praxis nicht verwendet.

redirect for the type of service and the host (Typ 5, Code 3)

Die ICMP Nachricht *redirect for the type of service and the host* verhält sich direkt analog zur *redirect for the host* Nachricht. Einziger Unterschied besteht in der gesonderten Betrachtung des Type of Service (ToS) Felds der eingegangenen IP Pakete. Eine *redirect for the type of service and the host* Nachricht wird ausschließlich in dem Fall generiert, wenn bezüglich des ToS Feldes für unterschiedliche Arten von Paketen auch unterschiedliche Routen existieren. Besteht die Umleitung konkret für den verwendeten ToS des eingegangenen Paketes, so wird vom Gateway auch die speziellere *redirect for the type of service and the host* Nachricht generiert und zurück an den Sender geschickt.

3.3.5 *Echo (Typ 8)*

Eine ICMP *Echo* Nachricht wird typischerweise zu einem entfernten Host im Netzwerk gesendet, um dessen Erreichbarkeit und die Qualität der Verbindung in Erfahrung zu bringen. Das Format einer *Echo* Nachricht ist identisch zum Format der *Echo Reply* Nachricht und enthält die fünf Angaben Nachrichtentyp, Unterkategorie, Prüfsumme, Identifikationsnummer und Sequenznummer. Das besondere an ICMP *Echo* Nachrichten liegt in dem anschließenden Datenbereich des Paketes. Ein typisches ICMP *Echo* Paket hat eine

Gesamtlänge von 64 Oktetten, so dass für diesen Datenbereich üblicherweise 56 Oktette zur Verfügung stehen, die beliebig belegt werden können. Es ist aber auch möglich, die ICMP Pakete größer zu wählen, um einen größeren Datenbereich zu erhalten. Die Längenbeschränkung richtet sich dann oftmals an der Länger der Maximum Transfer Unit (MTU), bei deren Überschreitung das ICMP Paket fragmentiert werden müsste. Mit einem ICMP *Echo* Paket ist es demnach möglich, versteckt Daten zwischen zwei Rechnern auszutauschen. Abbildung 3.6 zeigt den schematischen Aufbau einer ICMP *Echo* Nachricht.

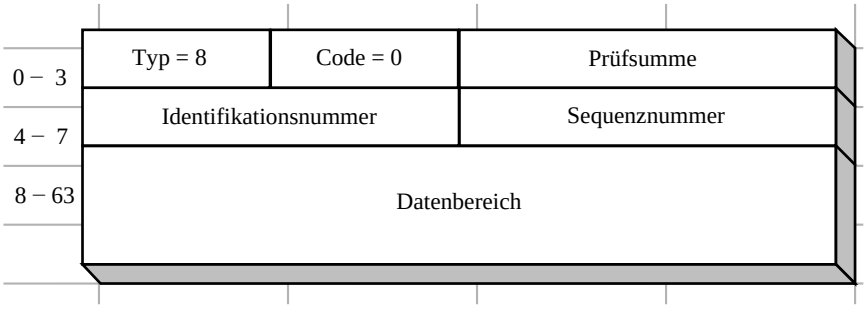

Abb. 3.6. Schematische Darstellung eines ICMP *Echo* Paketes. Die Länge des Datenbereiches ist prinzipiell variabel, wird aber in der Praxis häufig auf 56 Oktette beschränkt.

3.3.6 *Router Advertisement Message (Typ 9)*

Der ICMP Nachrichtentyp *Router Advertisement Message* ist erst in der RFC 1256, die sich mit einer Methode für Router zur Erkennung anderer Router im Netzwerk beschäftigt, zu den bereits vorhandenen Nachrichtentypen hinzugefügt worden. Innerhalb der TCP/IP Protokollfamilie sind mehrere dynamische Routingprotokolle definiert, mit deren Hilfe die einzelnen Router des Netzwerks ihre Informationen untereinander austauschen. Damit die Endgeräte in den Netzwerken auch ohne ein tiefergehendes Verständnis dieser höheren Routingprotokolle in die Lage versetzt werden können, die Gateways in ihrem Netzwerksegment zu erkennen und zu identifizieren, wird in der RFC 1256 eine Methode zur Nachbarrouter-Erkennung spezifiziert, die rein auf dem ICMP Protokoll basiert. Eine garantierte Kommunikationsmöglichkeit unabhängig von den IP Adressen der Systeme erhält man über den Multicast Mechanismus [53]. Dieser verwendet den speziell für Multicasting reservierten Klasse-D IP Adressbereich 224.0.0.0/4 (224.0.0.0 - 239.255.255.255). Jeder dieser IP Adressen kann eine Gruppe von Netzwerkgeräten zugewiesen werden, die alle als Empfänger für diese IP Adresse betrachtet werden. Eine spezielle Adresse aus der Klasse-D ist die Adresse 224.0.0.1, zu der jede Multicasting-fähige Komponente des Netzwerkes zählt. Die Router versenden

nun die ICMP *Router Advertisement Message* Nachricht in einem IP Paket mit der Zieladresse 224.0.0.1, so dass sie damit prinzipiell jedes andere Gerät erreichen können.

Abbildung 3.7 zeigt den schematischen Aufbau einer ICMP *Router Advertisement Message* Nachricht. Aus den Werten für die Adressanzahl und die Eintragsgröße kann letztendlich die genaue Größe des gesamten ICMP Paketes berechnet werden. Im Normalfall hat jeder Eintrag genau die Größe von zwei 32-Bit Werten, so dass sich für jede angegebene Router-Adresse die Paketgröße um 8 Oktette vergrößert. Die im Paket spezifizierte Lebensdauer bezieht sich auf die Gültigkeit der mitgesendeten Router-Adressen. Im Standardzustand senden Router alle 600 Sekunden (10 Minuten) ein *Router Advertisement Message* Paket, und die Lebensdauer der übermittelten Informationen beträgt 1800 Sekunden (30 Minuten).

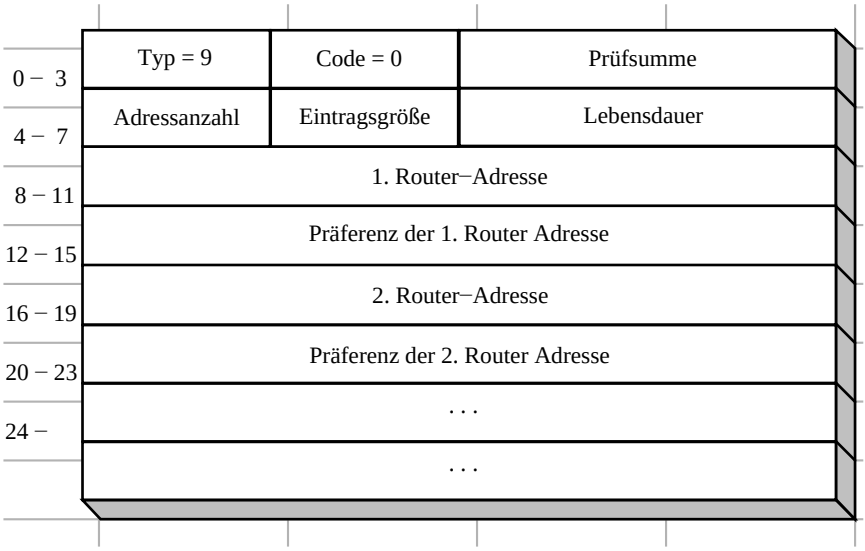

Abb. 3.7. Schematische Darstellung eines ICMP *Router Advertisement Message* Paketes. Die Anzahl der im Paket enthaltenen Router Adressen ist nicht explizit beschränkt.

Jeder Router-Adresse ist außerdem ein Präferenzwert hinzugefügt, über den ein Netzwerkadministrator bestimmen kann, welche Routen bevorzugt von den Endgeräten verwendet werden. Höhere Präferenzen haben dabei Vorrang vor niedrigeren Werten.

3.3.7 *Router Solicitation Message (Typ 10)*

Ebenso wie der Nachrichtentyp *Router Advertisement Message* ist auch der Typ *Router Solicitation Message* in RFC 1256 definiert und Bestandteil einer

Methode für Netzwerkendgeräte zur Ermittlung und Identifizierung von Gateways im selben Subnetz. Mit Hilfe von ICMP Nachrichten des Typs *Router Solicitation Message*, die an die IP Adresse 224.0.0.1 gesendet werden, kann ein Endgerät im Netzwerk die Gateways in seiner Umgebung unmittelbar anweisen, mittels ICMP *Router Advertisement Message* Nachrichten die ihnen bekannten Informationen über Routen zu verteilen. Dies ermöglicht es einem Endgerät, sich noch vor Ablauf der eingestellten Wiederholzeit für die *Router Advertisement Message* Nachrichten ein Bild über das Netzwerk zu machen und das richtige Gateway auszuwählen. Im Normalfall beträgt diese Zeitspanne 600 Sekunden (10 Minuten), die durch das Aussenden eines ICMP *Router Solicitation Message* Paketes erheblich verkürzt werden kann.

Mit dem in Abbildung 3.8 gezeigten schematischen Aufbau besitzt der ICMP Nachrichtentyp *Router Solicitation Message* die einfachste Form von allen ICMP Paketen. Neben den absolut notwendigen Angaben existiert lediglich ein zur Zeit noch nicht näher spezifizierter reservierter Datenbereich mit einer Länge von 4 Oktetten.

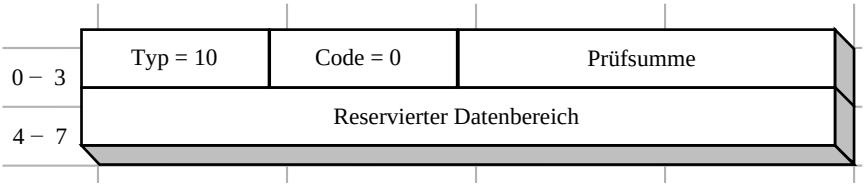

Abb. 3.8. Schematische Darstellung eines ICMP *Router Solicitation Message* Paketes. Der reservierte Datenbereich ist für zukünftige Erweiterungen vorgesehen und wird auf Null gesetzt.

3.3.8 *Time Exceeded (Typ 11)*

Bei der Übermittlung von IP Paketen können zwei grundsätzlich verschiedene Situationen auftreten, bei denen man von einer „Zeitüberschreitung" sprechen kann. Der ICMP Nachrichtentyp *Time Exceeded* liefert daher auch die beiden verschiedenen Unterkategorien *time to live exceeded* und *fragment reassembly time exceeded*. Abbildung 3.9 veranschaulicht passend dazu den schematischen Aufbau der *Time Exceeded* Nachrichten, die ähnlich zu den meisten anderen ICMP Paketen aufgebaut sind.

time to live exceeded (Typ 11, Code 0)

Eine *time to live exceeded* Nachricht kann immer dann von einem Gateway erzeugt werden, wenn ein empfangenes Paket im Time-to-Live (TTL) Feld des IP Paketkopfes auf den Wert Null absinkt. Über das TTL Feld kann

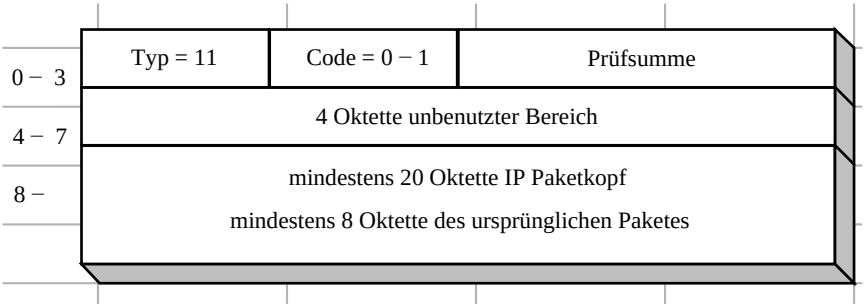

Abb. 3.9. Schematische Darstellung eines ICMP *Time Exceeded* Paketes. Die Länge des Datenbereiches ist variabel. Enthalten ist der originale IP Paketkopf der eingegangenen Nachricht, der mindestens 20 Oktette umfasst, sowie mindestens die ersten 8 Oktette des ursprünglichen Datenbereiches des eingegangenen Paketes.

bereits beim Erzeugen eines Paketes bestimmt werden, wieviele Gateways auf dem Übertragungsweg maximal passiert werden dürfen. Dazu erniedrigt jedes Gateway nach dem Erhalt eines IP Paketes zunächst dessen TTL Wert um Eins. Sinkt der TTL Wert dabei auf Null ab, so ist die „Lebenszeit" des Paketes abgelaufen und es wird vom Gateway verworfen. Anschließend kann das Gateway den Absender des Paketes über den Verlust informieren, indem es eine *time to live exceeded* Nachricht an den Ursprung der Nachricht sendet.

fragment reassembly time exceeded (Typ 11, Code 1)

Bei der Übermittlung von IP Paketen können auch andere Zeitgrenzen überschritten werden, die nicht mit der Anzahl der Gateways auf dem Vermittlungsweg zusammenhängen. Schließlich können IP Pakete beliebige Nachrichten mit beliebiger Größe enthalten. Übersteigt die Länge des Inhalts das Fassungsvermögen – also die Maximum Transfer Unit (MTU) – des IP Paketes, so wird der Inhalt einfach in mehrere Teile zerlegt, so dass die Einzelteile jeweils in ein IP Paket passen. Die entstandenen Fragmente werden anschließend völlig unabhängig voneinander über das IP Netzwerk verschickt, so dass die Fragmente auch unterschiedliche Wege zum Ziel nehmen können und naturgemäß auch in unsortierter Reihenfolge beim Empfänger eintreffen können. Der Empfänger speichert nun zunächst die erhaltenen Fragmente zwischen, um sie anschließend zum ursprünglichen Inhalt wieder zusammenzusetzen. Damit der Eingangspuffer des Empfängers für die Fragmente nicht Gefahr läuft, unkontrolliert überzulaufen, werden einzelne Fragmente nur für eine festgelegte Zeitspanne zwischengespeichert. Sind bei Ablauf dieser Zeitspanne noch nicht alle Fragmente beim Empfänger eingetroffen, so verwirft dieser sämtliche Fragmente. Außerdem kann er dem Absender die unvollständige Übermittlung durch eine *fragment reassembly time exceeded* Nachricht signalisieren.

3.3.9 *Parameter Problem (Typ 12)*

Kann ein Empfänger ein eingegangenes IP Paket nicht korrekt verarbeiten, weil es einen syntaktischen Fehler in seinem IP Paketkopf aufweist, so muss der Empfänger das Paket verwerfen. Gleichzeitig kann er den Absender über das aufgetretene Problem informieren, indem er eine *Parameter Problem* Nachricht zurücksendet. Zur weiteren Spezifizierung des Problems enthält eine *Parameter Problem* Nachricht zusätzlich noch ein Feld mit einem Zeiger auf dasjenige Oktett des eingegangenen IP Paketes, welches den Syntaxfehler enthält. Die Fehlererkennung beschränkt sich auf Angaben im IP Paketkopf, da aus Sicht von ICMP nur die Vermittlung von IP Paketen relevant ist. Zusätzliche Informationen über den konkreten Inhalt der Pakete und dessen genaue Bedeutung erschließen sich dem Empfänger erst zu einem späteren Zeitpunkt, wenn das IP Paket an die im OSI Referenzmodell höheren Protokolle weitergereicht wurden.

Um den Wert des angegebenen Zeigers richtig deuten zu können, muss man die Position der einzelnen Angaben im IP Paketkopf kennen. Tabelle 3.2 listet in der zweiten Spalte die Oktettnummern der jeweils aufgeführten Angaben des IP Paketkopfes auf, auf welche der Zeiger einer *Parameter Problem* Nachricht verweist.

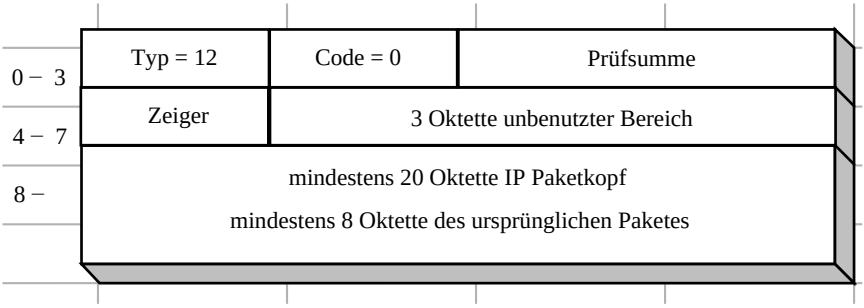

Abb. 3.10. Schematische Darstellung eines ICMP *Parameter Problem* Paketes. Die Länge des Datenbereiches ist variabel. Enthalten ist der originale IP Paketkopf der eingegangenen Nachricht, der mindestens 20 Oktette umfasst, sowie mindestens die ersten 8 Oktette des ursprünglichen Datenbereiches des eingegangenen Paketes.

3.3.10 *Timestamp (Typ 13)*

Mit Hilfe der beiden ICMP Nachrichtentypen *Timestamp* und *Timestamp Reply* können einfachste Zeitmessungen für die Erreichbarkeit einzelner Gateways und Endgeräte im Netzwerk durchgeführt werden. Zu diesem Zweck besitzen die beiden Nachrichtentypen neben den für alle ICMP Pakete gemeinsamen Datenfeldern drei weitere Felder, die jeweils für verschiedene Zeitstempel

vorgesehen sind. Alle drei Felder verwenden dasselbe Format, bei welchem der Zeitstempel in Millisekunden seit Mitternacht der Universal Time Constant (UTC) angegeben wird. In Ausnahmefällen können auch andere Zeitformate gewählt werden, jedoch muss dann das höchstwertige Bit des 32-Bit Wortes für den Zeitstempel gesetzt sein. Das funktioniert deshalb so gut, weil das ursprüngliche Format einen Wertebereich 0 bis 86 399 999 aufweist[5], wobei der größte mögliche Wert in binärer Schreibweise nur 27 Bit benötigt[6].

Für eine *Timestamp* Nachricht ist zunächst nur das erste Feld mit einem Zeitstempel zu belegen. Während die beiden anderen Felder nur für die Antwortpakete wichtig sind und daher auf Null gesetzt werden, enthält das erste Feld den Zeitstempel des Absenders unmittelbar bevor das Paket den Host verlässt. Diese Zeitangabe definiert somit gleichzeitig den Startpunkt der Zeitmessung.

Abbildung 3.11 zeigt den schematischen Aufbau eines ICMP *Timestamp* Paketes. Die beiden Felder mit Angaben des Empfängers werden auf Null gesetzt.

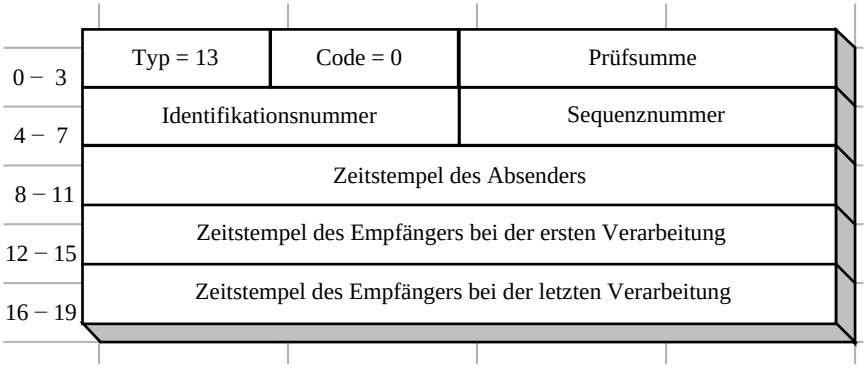

Abb. 3.11. Schematische Darstellung eines ICMP *Timestamp* Paketes. Ein von einem Host generiertes ICMP Paket vom Typ *Timestamp* enthält zunächst nur einen gültigen Wert im ersten Datenteil, welcher den Zeitstempel des Absenders enthält.

[5] Zu der Zahl 86 399 999 gelangt man recht einfach, wenn man berücksichtigt, dass ein Tag 24 Stunden mit 60 Minuten mit 60 Sekunden mit 1000 Millisekunden hat. Die Uhrzeit 23:59:59,999 UTC hat demnach den Zeitstempel $24 \times 60 \times 60 \times 1000 - 1 = 86\,399\,999$.

[6] Die Dezimalzahl 86 399 999 lässt sich in binärer Schreibweise durch die Zeichenkette 00000101 00100110 01011011 11111111 wiedergeben. Die ersten fünf Bit sind nicht gesetzt, so dass sich der Zeitstempel mit den restlichen 27 Bit repräsentieren lässt.

3.3.11 *Timestamp Reply (Typ 14)*

Empfängt ein System eine ICMP Nachricht vom Typ *Timestamp*, so muss das System zur Unterstützung einer möglichst genauen Zeitmessung zunächst einmal unmittelbar nach dem Empfang des Paketes einen zweiten Zeitstempel zum Paket hinzufügen, der im Datenfeld direkt im Anschluss an den Zeitstempel des Absenders platziert wird. Mit diesen beiden Zeitangaben könnten man bereits theoretisch eine Laufzeitangabe für das Paket ermitteln, aber diese würde dann einen weiter unten beschrieben Messfehler aufweisen. Stattdessen generiert der Empfänger als Antwort auf das eingegangene ICMP Paket eine *Timestamp Reply* Nachricht, die er an den ursprünglichen Absender zurückschickt. Auch hier wird wieder unmittelbar vor dem Verlassen des Gerätes dem Paket ein Zeitstempel hinzugefügt. Im Unterschied zu der Zeitangabe, die der Empfänger direkt bei Erhalt des Paketes hinzugefügt hat, spezifiziert der mittlerweile dritte Zeitstempel im Paket exakt den Zeitpunkt des Verlassens des Empfängers. Die Differenz zwischen diesen beiden Zeiten entspricht genau der Bearbeitungszeit des *Timestamp* Paketes im Empfänger.

Der Sender einer ICMP *Timestamp* Nachricht sollte unter normalen Bedingungen nach einiger Zeit ein Antwortpaket erhalten, das neben der von ihm bereits beim Absenden spezifizierten Zeitangabe t_1 noch die beiden Zeitstempel t_2 und t_3 des Empfängers enthält. Zusätzlich erfasst der Empfänger mit dem Zeitpunkt t_4 des Eintreffens der Antwort eine vierte Zeitangabe. Man könnte nun meinen, dass jeweils aus der Differenz zwischen t_1 und t_2 sowie t_3 und t_4 die beiden Vermittlungszeiten vom Sender zum Empfänger und in der Gegenrichtung bestimmt werden können. Unter optimalen Randbedingungen stimmt dies auch, aber in der Praxis werden die beiden Systeme nur in den seltensten Fällen auf die Hundertstelsekunde genau synchron getaktet sein. In den meisten Fällen wird also der Empfänger einen Zeitfehler Δt gegenüber dem Sender aufweisen, so dass die beiden beim Empfänger gemessenen Zeitpunkte t_2 und t_3 durch die beiden korrigierten Zeitangaben $t_2' + \Delta t$ und $t_3' + \Delta t$ ersetzt werden müssen.

$$t_1 := \text{Zeitpunkt beim Verlassen des Senders (Senderzeit)} \tag{3.1}$$
$$t_2 := \text{Zeitpunkt beim Erreichen des Empfängers (Empfängerzeit)} \tag{3.2}$$
$$t_2' := \text{Zeitpunkt beim Erreichen des Empfängers (Senderzeit)} \tag{3.3}$$
$$t_3 := \text{Zeitpunkt beim Verlassen des Empfängers (Empfängerzeit)} \tag{3.4}$$
$$t_3' := \text{Zeitpunkt beim Verlassen des Empfängers (Senderzeit)} \tag{3.5}$$
$$t_4 := \text{Zeitpunkt beim Erreichen des Senders (Senderzeit)} \tag{3.6}$$
$$\Delta t := \text{Zeitdifferenz zwischen Sender und Empfänger} \tag{3.7}$$

Da über die Zeitabweichung Δt keinerlei Informationen bekannt sind, enthalten also beide Einzelangaben für die Vermittlungszeit des Hinwegs (3.9) und des Rückwes (3.10) einen Messfehler Δt unbekannter und beliebiger Größe. Die nachfolgenden Gleichungen sollen diesen Fehler verdeutlichen:

$$t_2 - t_2' = t_3 - t_3' = \Delta t \tag{3.8}$$

$$t_2 - t_1 = t_2' + \Delta t - t_1 = \text{Berechnete Laufzeit Hinweg} \tag{3.9}$$

$$t_4 - t_3 = t_4 - (t_3' + \Delta t) = \text{Berechnete Laufzeit Rückweg} \tag{3.10}$$

Es existiert jedoch eine Möglichkeit, eine fehlerfreie Angabe für die Gesamtlaufzeit t_G der Pakete auf dem Hinweg und dem Rückweg zu machen. Diese berechnet sich schließlich als Summe aus den beiden Einzellaufzeiten, die sich dann noch wie folgt vereinfachen lässt:

$$\begin{aligned} t_G &= (t_2 - t_1) + (t_4 - t_3) \\ &= \Big((t_2' + \Delta t) - t_1\Big) + \Big(t_4 - (t_3' + \Delta t)\Big) \\ &= (t_2' - t_1) + (t_4 - t_3') \\ &= (t_4 - t_1) - (t_3' - t_2') \end{aligned} \tag{3.11}$$

Schaut man sich das Endergebnis (3.11) einmal genauer an, so stellt man fest, was einem auch schon vorher hätte klar sein können: Die Gesamtlaufzeit der Pakete entspricht der gesamten Zeitspanne zwischen Absenden des *Timestamp* Paketes und Empfangen des *Timestamp Reply* Paketes $(t_4 - t_1)$ abzüglich der Verarbeitungszeit $(t_3' - t_2')$ im Empfänger. Die Zeitabweichung zwischen Sender und Empfänger spielt bei der Ermittlung der Gesamtlaufzeit keine Rolle mehr.

Zum Abschluss ist in Abbildung 3.12 der schematische Aufbau der *Timestamp Reply* Pakete dargestellt, der sich nicht von den *Timestamp* ICMP Paketen unterscheidet. Lediglich die beiden letzten Datenfelder sind im Unterschied zu den *Timestamp* Nachrichten bei den Antwortpaketen ebenfalls mit Inhalt gefüllt.

3.3.12 *Information Request (Typ 15)*

Der ICMP Nachrichtentyp *Information Request* sowie dessen passende Antwort *Information Reply* waren ursprünglich für selbstkonfigurierende Systemkomponenten erdacht worden, die zum Zeitpunkt einer Initialisierung Informationen über das Netzwerk benötigen, in dem sie sich befinden. Die ursprüngliche Definition hat vorgesehen, dass ein Endgerät in einem Netzwerk bei der Initialisierung ein ICMP Paket vom Typ *Information Request* sendet, und dabei die Felder für die IP Adressen im IP Paketkopf auf Null setzt. Als Antwort erwartet der Host ein Paket vom Typ *Information Reply*, dessen Felder für die IP Adressen ausgefüllt sind. Die mitgelieferte IP Adresse sollte dann vom Host ausgelesen und übernommen werden. Dieser Mechanismus ist allerdings veraltet und wird nicht mehr verwendet. In RFC 1812 [8], die Anforderungen an IP Router des IP Protokolls der Version 4 definiert, wird sogar angegeben,

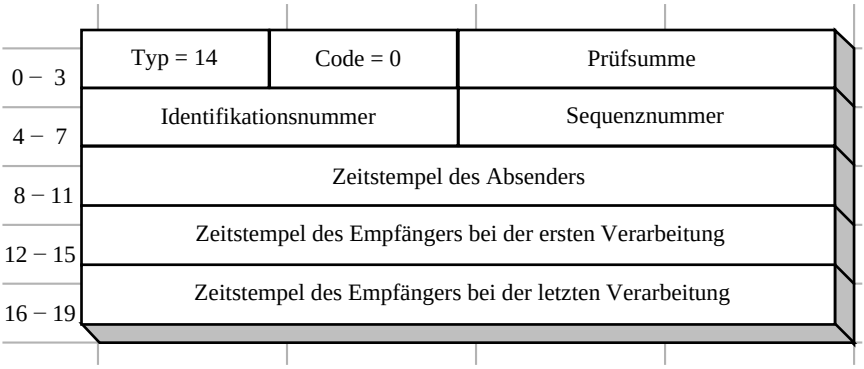

Abb. 3.12. Schematische Darstellung eines ICMP *Timestamp Reply* Paketes. Die Antwort auf ein ICMP Paket vom Typ *Timestamp* enthält neben dem originalen Zeitstempel des Absenders auch zwei Zeitstempel des Empfängers: einen für den Zeitpunkt der ersten Bearbeitung und einen für den Zeitpunkt der letzten Bearbeitung.

dass diese ICMP Pakete nicht mehr unterstützt werden und ignoriert werden sollen. Zur Vollständigkeit soll in Abbildung 3.13 dennoch die schematische Struktur der *Information Request* Pakete verdeutlicht werden.

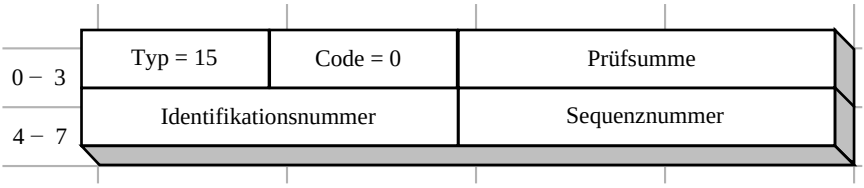

Abb. 3.13. Schematische Darstellung eines ICMP *Information Request* Paketes. Die wesentlichen Informationen befinden sich nicht im ICMP Paket, sondern im IP Paketkopf bei den IP Adressen.

3.3.13 *Information Reply (Typ 16)*

Ebenso wie die *Information Request* Pakete sind auch die ICMP Pakete vom Typ *Information Reply* mittlerweile veraltet und sollen nicht mehr verwendet werden. Zur Zuweisung einer IP Adresse existieren in heutigen IP Netzwerken andere und bessere Mechanismen, wie beispielsweise der Bootstrap Protocol (BOOTP) [47] oder der Dynamic Host Configuration Protocol (DHCP) [64] Mechanismus. Diese Mechanismen bieten wesentlich ausgereiftere Möglichkeiten einer zentralen Verwaltung von Hostkonfigurationen in Bezug auf das IP Protokoll. Beispielsweise können nicht nur IP Adressen via DHCP Protokoll

an Endgeräte vergeben werden, sondern außerdem auch die Netzwerkmaske oder die IP Adressen von Gateways und Nameservern verschiedener Art. Abbildung 3.14 zeigt die schematische Struktur der *Information Reply* ICMP Pakete, die sich nicht von der Struktur der *Information Request* Pakete unterscheidet.

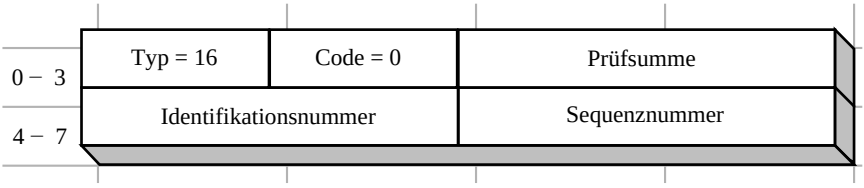

Abb. 3.14. Schematische Darstellung eines ICMP *Information Reply* Paketes. Die wesentlichen Informationen befinden sich nicht im ICMP Paket, sondern im IP Paketkopf bei den IP Adressen.

3.3.14 *Address Mask Request (Typ 17)*

Die öffentlichen IP Adressen sind in feste Klassen unterteilt, die deren Netzwerkgröße und damit auch Netzwerkmaske eindeutig definieren (siehe Tabelle 3.3). Allerdings können IP Netzwerke auch in mehrere Subnetze aufgeteilt werden, oder es können mehrere Netzwerke zu einem Supernetz zusammengefasst werden, so dass aus der Kenntnis einer IP Adresse nicht mit absoluter Sicherheit auf die passende Netzwerkmaske geschlossen werden kann. Aus diesem Grund wird auch nicht weiter an der strikten Klassenunterteilung festgehalten.

Tabelle 3.3. Klassenunterteilung der IPv4 Netzwerkadressen.

Klasse	Netzwerkmaske	Adressbereich
A	255.0.0.0	0.0.0.0 - 127.255.255.255
B	255.255.0.0	128.0.0.0 - 191.255.255.255
C	255.255.255.0	192.0.0.0 - 223.255.255.255
D	255.255.255.255	224.0.0.0 - 239.255.255.255
E	nicht definiert	240.0.0.0 - 255.255.255.255

Damit ein IP Endgerät, welches eventuell sogar Kenntnis über seine IP Adresse besitzt, auch die dazugehörige Netzwerkmaske in Erfahrung bringen kann, genügt das Aussenden eines ICMP *Address Mask Request* Paketes an ein Gateway im selben Subnetz. Dieses wiederum kennt die Netzwerkstruktur mit den

verwendeten Netzwerkmasken und kann daher auch erschöpfend Auskunft erteilen. Kennt das Endgerät seine IP Adresse nicht, so kann es im IP Paketkopf das Feld für die Absender-Adresse auf Null setzen.

Aus Abbildung 3.15 geht hervor, dass ICMP Nachrichten vom Typ *Address Mask Request* dieselbe Struktur wie die dazugehörigen Antworten *Address Mask Reply* besitzen. Bei den Anfragen ist das Datenfeld für die Netzwerkmaske jedoch noch nicht gefüllt und auf Null gesetzt.

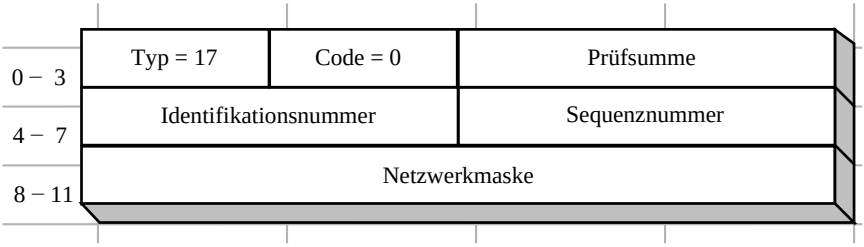

Abb. 3.15. Schematische Darstellung eines ICMP *Address Mask Request* Paketes. Das Datenfeld für die Netzwerkmaske wird bei einer Anfrage nicht verwendet und ist auf Null gesetzt.

3.3.15 *Address Mask Reply (Typ 18)*

Als Antwort auf ein ICMP *Address Mask Request* Paket sendet ein Gateway die verwendete Netzwerkmaske an das absendende Endgerät zurück. Die Information wird dazu in das dafür vorgesehene Datenfeld geschrieben, welches in der Anfrage noch auf Null gesetzt war.

Ist das Feld für die Absender-Adresse im IP Paketkopf leer, so muss das Gateway davon ausgehen, dass der Absender nicht über seine IP Adresse verfügt. In diesem Fall ist die Antwort zur Sicherstellung der Auslieferung mittels eines Broadcasts an alle Geräte zu senden. Diesen Umstand gilt es jedoch bereits auf Seiten des Absenders möglichst zu vermeiden, weil er unweigerlich eine unnötig erhöhte Netzwerklast mit sich zieht.

Abbildung 3.16 zeigt den schematischen Aufbau der ICMP Pakete vom Typ *Address Mask Reply*. Die Struktur ist identisch zu den Anfragen des Typs *Address Mask Request*. Es bleibt an dieser Stelle noch zu erwähnen, dass für einen reibungslosen Mechanismus zur Ermittlung von Netzwerkmasken nicht die zwingende Notwendigkeit zur Synchronisation von gesendeter Anfrage und erhaltener Antwort besteht. Aus diesem Grund sind sowohl beim ICMP Nachrichtentyp *Address Mask Request* als auch bei den passenden Antwortpaketen *Address Mask Reply* die beiden im Paket vorgesehenen Datenfelder für die Identifikationsnummer und für die Sequenznummer ungenutzt.

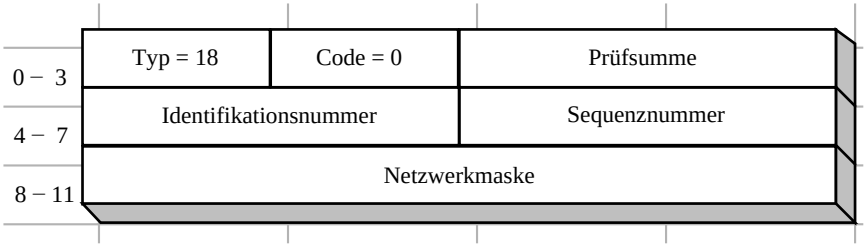

Abb. 3.16. Schematische Darstellung eines ICMP *Address Mask Reply* Paketes. Die angegebene Netzwerkmaske kann auch aus nicht-zusammenhängenden Bits bestehen.

3.3.16 *Traceroute (Typ 30)*

Der *Traceroute* Mechanismus wurde mit dem klaren Ziel entwickelt, dem Netzwerkadministrator eine bequeme Methode zur Verfügung zu stellen, mit deren Hilfe er deutlich mehr als nur die simple Erreichbarkeit eines Netzwerkgerätes überprüfen kann. Anders als der auf den ICMP Nachrichtentypen *Echo* und *Echo Reply* basierende PING Befehl soll der *Traceroute* Mechanismus die zusätzliche Erhöhung der Transparenz des gesamten Übertragungsweges vom Absender zum Empfänger schaffen – inklusiver aller auf dem Weg liegenden Zwischenstationen.

Das Prinzip dieses neuartigen *Traceroute* Mechanismus basiert auf zwei verschiedenen Konstrukten. Zum einen wurde der neue ICMP Nachrichtentyp *Traceroute* mit der Nummer 30 eingeführt. Zum anderen wurde das IP Protokoll um eine *Traceroute* Option im Paketkopf erweitert, die nicht zum ICMP Protokoll zählt. Erst die Kombination aus beiden Konstrukten erlaubt eine erfolgreiche Anwendung des *Traceroute* Mechanismus.

Ein Netzwerkgerät, welches detaillierte Informationen über die Verfügbarkeit einer anderen Netzwerkkomponente sucht, sendet zunächst ein nicht näher spezifiziertes IP Paket an das Zielsystem, dessen *Traceroute* Option im Paketkopf gesetzt ist (siehe Abbildung 3.17). Oftmals verwendet man für den Inhalt des Paketes ein normales ICMP *Echo* Paket als Nutzdaten. Da es sich um ein ausgehendes Paket handelt, ist in der *Traceroute* Option der Datenbereich für die Anzahl der Hops des Rückpaketes mit dem speziellen hexadezimalen Wert $FFFFh$ initialisiert. Der Datenbereich für die Anzahl der Hops des ausgehenden Paketes enthält eine Null, und der Bereich für die Absender-Adresse wird mit der eigenen IP Adresse gefüllt. Das erste Oktett der *Traceroute* Option im IP Paketkopf setzt sich aus drei unterschiedlichen Informationen zusammen:

- Das erste Bit zeigt an, ob die *Traceroute* IP Option auch in Fragmente eines aufgeteilten IP Paketes eingefügt werden soll. Da dies nicht gewünscht ist, steht das erste Bit auf 0.

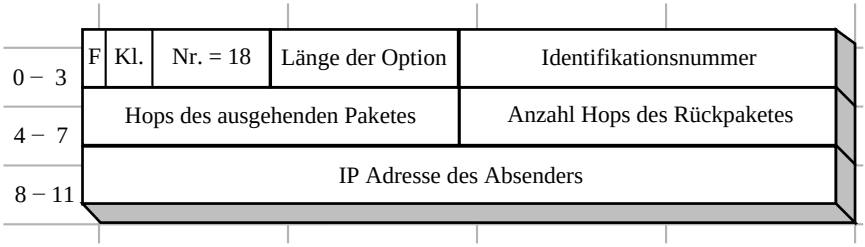

Abb. 3.17. Schematische Darstellung der *Traceroute* Option des Protokolls IP. Das allererste Bit „F" legt fest, dass diese IP Option nicht in Fragmente von Paketen übertragen werden darf, und ist daher auf Null gesetzt. Die nächsten beiden Bit geben Auskunft über die Klasse *Debugging & Measurement* der Option und sind auf 10 gesetzt. Die letzten fünf Bit des ersten Oktetts geben die eindeutige Nummer (10010) der Option an, so dass in allen IP Paketen mit dieser Option das erste Oktett eine 01010010 (82) enthält.

- Die nächsten beiden Bit bestimmen die Klasse der Option. Zur Zeit sind nur die beiden Klassen 00 (*control*) und 10 (*Debugging & Measurement*) bekannt. Da die *Traceroute* Option zur Klasse *Debugging & Measurement* zählt, sind die beiden relevanten Bits auf 10 gesetzt.
- Die restlichen fünf Bit des ersten Oktetts einer IP Option geben die eindeutige Nummer der IP Option an. Tabelle 3.4 listet einige Typen für IP Optionen auf. Die Nummer der *Traceroute* Option beträgt 18, so dass sich für die fünf Bit der Wert 10010 ergibt.

Tabelle 3.4. Einige bekannte IP Optionen.

Nummer	Option
0	Ende der Optionen Liste
1	No Operation (NOP)
2	Sicherheitsoptionen
3	Lockere Quellvermittlung („Loose Source Routing")
4	Internet Zeitstempel
7	Routen Protokollierung („Record Route")
8	Datenstrom Identifikationsnummer („Stream ID")
9	Strikte Quellvermittlung („Strict Source Routing")
18	Traceroute

Jedes Gateway auf dem Weg vom Sender zum Empfänger, das dieses IP Paket mit der *Traceroute* Option erhält und weiterleitet, sendet nun ein Antwortpaket an den ursprünglichen Absender, der im entsprechenden Datenfeld der IP Option angegeben ist. Als Antwortpaket wird ein ICMP *Traceroute* Paket ohne zusätzliche *Traceroute* Option IP Paketkopf verwendet. Abbildung 3.18

verdeutlicht den schematischen Aufbau des ICMP Nachrichtentyps *Traceroute*. Abhängig vom Erfolg des Weiterleitens vom ursprünglichen IP Paket kann das Gateway mit einem von zwei verschiedenen ICMP *Traceroute* Paketen antworten.

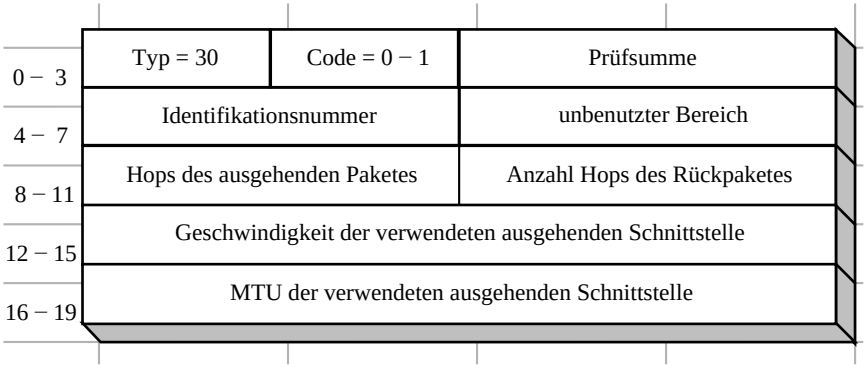

Abb. 3.18. Schematische Darstellung eines ICMP *Traceroute* Paketes. Unter den Hops ist die Anzahl der passierten Router auf dem Weg der Pakete zu verstehen.

traceroute successfully forwarded (Typ 30, Code 0)

Nachdem ein Gateway ein eingegangenes IP Paket mit gesetzter *Traceroute* Option erfolgreich in Richtung des Empfängers weitergeleitet hat, sendet es eine Bestätigungsmeldung an den ursprünglichen Absender zurück. Dazu wird das Antwortpaket *traceroute successfully forwarded* an die im Datenfeld der IP Option für den Absender gespeicherten IP Adresse gerichtet. In das Datenfeld des ICMP Paketes für die Anzahl der Hops des Hinwegs wird der entsprechende Wert aus dem IP Optionsfeld kopiert. Das ICMP Datenfeld für die Anzahl der Hops des Rückpaketes wird analog mit Null initialisiert. Um dem Absender noch zusätzliche Informationen über die Verbindungsstrecke mitzuteilen, trägt das Gateway auch noch die korrekten Werte für die MTU und die Verbindungsgeschwindigkeit derjenigen Schnittstelle in die vorgesehenen ICMP Datenfelder ein, die vom Gateway für den Rückweg verwendet wird. Auf diese Weise kann der Absender auch Informationen über die Bandbreite zwischen diesem Gateway und dem auf der Strecke vorhergehenden Gateway sammeln.

Enthält ein Gateway ein ICMP *traceroute successfully forwarded* Paket, so handelt es sich folgerichtig um ein Rückpaket an den Absender. In diesem Fall erhöht das Gateway den Zähler für die Anzahl der Hops des Rückpaketes und leitet es weiter in Richtung Empfänger.

no route to traceroute target (Typ 30, Code 1)

Kann ein Gateway ein eingehendes IP Paket mit gesetzter IP *Traceroute* Option aus irgendeinem Grund nicht weiterleiten, so sendet es die Fehlermeldung *no route to traceroute target* an den Absender zurück. Mit Ausnahme des Wertes für die Unterkategorie („Code“) werden jedoch alle anderen Datenfelder analog zur positiven Bestätigungsmeldung *traceroute successfully forwarded* ausgefüllt. Auch das Weiterleiten von *no route to traceroute target* Paketen anderer Gateways an den ursprünglichen Empfänger wird ebenfalls wie bei der positiven Bestätigung abgehandelt: Nur der Wert des Zählers für die Anzahl der Hops des Rückpaketes wird um Eins erhöht und das Paket dann weitergeleitet.

3.4 Auf ICMP basierende Werkzeuge

Die wichtigsten auf ICMP basierenden Werkzeuge sind zweifelsohne der PING und der TRACEROUTE Befehl. Obwohl der Name es vielleicht vermuten lässt, handelt es sich bei dem letzteren nicht um ein Werkzeug, das auf dem gleichnamigen ICMP Nachrichtentyp *Traceroute* basiert, sondern wie der PING Befehl auch auf den beiden Nachrichtentypen *Echo* und *Echo Reply*.

3.4.1 PING

Der PING Befehl verwendet die beiden ICMP Nachrichtentypen *Echo* und *Echo Reply*, um ein Maß für die Erreichbarkeit und Verfügbarkeit eines beliebigen Gerätes im Netzwerk zu liefern. Typischerweise ermittelt das PING Werkzeug neben dem erfolgreichen Eintreffen der Rückantworten auch noch die Laufzeit der Pakete, indem es die Zeit zwischen Absenden der ICMP *Echo* Nachricht und Empfangen der ICMP *Echo Reply* Nachricht misst. Als ein Maß für die Erreichbarkeit steht dann schließlich neben dem prozentualen Verhältnis der empfangenen Antworten auch die durchschnittliche Antwortzeit des Zielsystems. Abbildung 3.19 zeigt ein Beispiel für die Ausgabe eines PING Befehls auf einem Linux Rechner.

Anhand der Ausgabe kann man die Laufzeit jedes einzelnen Paketes erkennen. Außerdem führt das *ping* Programm eine Statistik über die Anzahl der gesendeten Pakete und erhaltenen Antworten. Schließlich wird aus allen gemessenen Laufzeiten noch eine durchschnittliche Round Trip Time (RTT) ermittelt, welche der durchschnittlichen Zeit für den Hinweg eines *Echo* Paketes plus der Zeit für den Rückweg des *Echo Reply* Paketes zum ursprünglichen Absender darstellt.

Die Implementierung des PING Befehls ist auf verschiedenen Plattformen und in verschiedenen Betriebssystemen unterschiedlich ausgeprägt. Dies bezieht sich auf die zur Verfügung stehenden Kommandozeilenparameter ebenso

```
nms$ ping -n -c3 172.17.2.1
PING 172.17.2.1 (172.17.2.1) from 172.17.2.85 : 56(84) bytes.
64 bytes from 172.17.2.1: icmp_seq=1 ttl=64 time=0.492 ms
64 bytes from 172.17.2.1: icmp_seq=2 ttl=64 time=0.307 ms
64 bytes from 172.17.2.1: icmp_seq=3 ttl=64 time=0.302 ms
64 bytes from 172.17.2.1: icmp_seq=2 ttl=64 time=0.313 ms
64 bytes from 172.17.2.1: icmp_seq=3 ttl=64 time=0.301 ms

--- 172.17.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% loss, time 4246ms
rtt min/avg/max/mdev = 0.301/0.343/0.492/0.075 ms
nms$
```

Abb. 3.19. Beispiel für die Ausgabe eines PING Befehls auf einem Linux Rechner. Der Parameter -n verhindert eine Namensauflösung der IP Adressen und der Parameter -c5 limitiert die Anzahl der gesendeten ICMP *Echo* Pakete auf drei.

wie auf das Normalverhalten bezüglich der gesendeten Anzahl von *Echo* Paketen. Typisch sind aber der Parameter -c *n*, mit dessen Hilfe die Anzahl der zu sendenden ICMP Pakete bestimmt werden kann, sowie der Parameter -n, mit dem eine Namensauflösung der IP Adressen unterdrückt werden kann.

3.4.2 TRACEROUTE / TRACERT

Der Befehl TRACEROUTE – oder manchmal auch TRACERT genannt – geht über die Messung der bloßen Erreichbarkeit eines einzelnen Systems im Netzwerk hinaus. Mit dem TRACEROUTE Befehl können sogar die einzelnen Gateways auf dem Weg zum Ziel bestimmt und deren Erreichbarkeit ermittelt werden. Ist eine Netzwerkkomponente mittels PING Befehl nicht erreichbar, so kann beispielsweise mittels des TRACEROUTE Befehls das genaue Gateway ermittelt werden, bis zu welchem die Kommunikation noch einwandfrei funktioniert. Auf diese Weise können Fehler im Netzwerk deutlich besser isoliert werden.

ICMP TRACEROUTE

Unabhängig vom weiter oben beschriebenen *Traceroute* Mechanismus verwendet der TRACEROUTE Befehl in den meisten Fällen zwar ICMP Pakete, aber nicht vom Typ *Traceroute*. Vielmehr werden eine ganze Reihe von ICMP *Echo* Paketen erzeugt. Der Trick besteht nun darin, die im IP Paketkopf angegebene maximale Lebensdauer des ICMP Paketes auf ein Minimum zu setzen. Beginnt man mit einem TTL Wert von eins, so wird das unmittelbar folgende Gateway das Paket verwerfen müssen, da die maximale Lebensdauer beim Eintreffen im Gateway bereits erreicht worden ist. Als Folge davon antwortet das Gateway mit einer entsprechenden Fehlermeldung, für die im ICMP

Protokoll der Nachrichtentyp *time to live exceeded* (Typ 11, Code 0) zur Verfügung steht. Somit sind bereits zwei Messwerte aufgenommen worden: die IP Adresse des ersten Gateways auf dem Weg zum Ziel sowie die Laufzeit der Pakete bis dahin und zurück. Anschließend wird ein zweites ICMP *Echo* Paket versendet, bei dem dieses Mal allerdings die TTL um eins erhöht ist. Dieses Paket wird vom ersten und mittlerweile bekannten Gateway erfolgreich weitergeleitet werden können zum zweiten Gateway, bei dem dann die maximale Lebensdauer abläuft. Folglich sendet nun das zweite Gateway eine ICMP *time to live exceeded* Nachricht an den ursprünglichen Absender, um diesen über den Verlust des Paketes zu informieren. Mit diesem Paket kann nun die IP Adresse des zweiten Gateways und die Laufzeit der Pakete dorthin ermittelt werden. Diese Prozedur wird so lange wiederholt, bis die *Echo* Pakete ihr Ziel erreichen und gültige *Echo Reply* Antwortpakete an den Absender schicken. Zu diesem Zeitpunkt hat man alle IP Adressen auf dem Weg zum Ziel ermittelt und gleichzeitig die Laufzeit der Pakete zu diesen Systemen gemessen.

Abbildung 3.20 zeigt ein Beispiel für die Verwendung des TRACEROUTE Befehls auf einem Linux Rechner. Da sich die Implementierungen in den einzelnen Betriebssystemen teilweise sehr stark unterscheiden, sind die im Beispiel angegebenen Kommandozeilenparameter nicht unbedingt auf andere Systeme übertragbar. In diesem Fall steht der Parameter `-n` wieder für das Unterdrücken der Namensauflösung der ermittelten IP Adressen, während der Parameter `-I` die Verwendung von ICMP *Echo* Nachrichten aktiviert.

Schaut man sich die Ausgabe des TRACEROUTE Befehls etwas genauer an, so erkennt man direkt seine Ähnlichkeit zum PING Befehl. Jede Zeile in der Ausgabe entspricht einem Hop auf dem Weg zum Ziel. Für jeden Host auf dem Weg zum Ziel werden insgesamt drei Messungen durchgeführt, die sich in Form der Laufzeiten zu diesem Host ausdrücken und nach dessen IP Adresse ausgegeben werden.

Die Verwendung von ICMP *Echo* Nachrichten zur Durchführung eines *Traceroute* hat klare Nachteile. Zwar lässt sich mit Hilfe des Kommandozeilenparameters `-q1` die Anzahl der gesendeten Pakete pro Gateway auf eins reduzieren, jedoch werden in diesem Fall immer noch doppelt so viele Pakete ausgesendet wie Hops auf dem Weg zum Ziel liegen. Bei der Verwendung des *Traceroute* Mechanismus über die ICMP *Traceroute* Nachrichten würde nur ein einziges Paket auf dem Hinweg und für jeden Hop nur ein Paket für den Rückweg benötigt. Die Anzahl der ICMP *Echo* Pakete ist somit beinahe doppelt so groß. Es besteht jedoch ein klarer Vorteil gegenüber einer Verwendung von ICMP Nachrichten des Typs *Traceroute*. Die *Traceroute* Pakete wurden erst in der RFC 1393 definiert und kamen damit erst zu einem sehr späten Zeitpunkt auf. Bis dahin existierte bereits eine große Anzahl von Gateways, welche alle die beiden Nachrichtentypen *Echo* und *Echo Reply* beherrschten. Der neue Nachrichtentyp wurde von nur wenigen Gateways unterstützt, so dass sich in der Praxis der bereits bewährte Weg über die *Echo* Pakete durchgesetzt hat.

```
nms$ traceroute -n -I www.google.de
traceroute: Warning: www.google.de has multiple addresses;
            using 66.249.85.104
traceroute to www.l.google.com (66.249.85.104),
           30 hops max, 38 byte packets
 1  172.17.2.1  0.451 ms  0.308 ms  0.309 ms
 2  172.19.33.9  25.764 ms  1.520 ms  0.362 ms
 3  172.19.35.1  0.383 ms  0.381 ms  8.694 ms
 4  192.0.2.5  0.401 ms 192.0.2.2  24.022 ms  192.0.2.5  4.992 ms
 5  212.28.33.39  9.905 ms  0.404 ms  9.468 ms
 6  212.88.131.41  9.850 ms  0.426 ms  14.698 ms
 7  217.24.235.113  16.020 ms  20.070 ms  8.404 ms
 8  217.24.235.3  11.633 ms  8.174 ms  9.801 ms
 9  217.24.235.13  9.979 ms  22.802 ms  6.997 ms
10  80.81.192.108  21.516 ms  8.160 ms  22.267 ms
11  216.239.46.47  18.530 ms 66.249.94.136  22.424 ms
    216.239.46.49  14.296 ms
12  66.249.85.104  16.096 ms  13.260 ms  14.833 ms
nms$
```

Abb. 3.20. Beispiel für die Ausgabe eines TRACEROUTE Befehls auf einem Linux Rechner. Der Parameter -I aktiviert die Verwendung von *Echo* Paketen zur Messung des Übertragungsweges. Der Parameter -n verhindert gleichzeitig die Namensauflösung der IP Adressen.

Ein weiterer Nachteil für die auf dem PING Befehl basierende Methode ist die „Beweglichkeit" der Verbindungsstrecke. Das Internet ist ein dynamisches System, bei dem nicht alle Pakete denselben festen Wege zum Ziel nehmen müssen. Jedes ausgesendete ICMP Paket kann demnach einen anderen Weg im Netzwerk einschlagen. Die erhaltenen Antworten zu den Paketen mit vorgegebener TTL stammen daher auch nicht zwangsweise von denselben Gateways, die außerdem auch keine direkte Verbindung zum Ziel darstellen müssen. Dieser Umstand wird auch in Abbildung 3.20 deutlich. Werden von den ersten drei Gateways noch einheitliche Antworten empfangen, gibt es eine deutliche Abweichung beim vierten Gateway. Hier scheint offensichtlich eine redundante Anbindung zwischen dem dritten und dem fünften Gateway zu bestehen, die vermutlich nach dem Round-Robin[7] Verfahren genutzt wird. Es ist deutlich zu erkennen, dass die *time to live exceeded* Fehlermeldung der zugehörigen ICMP *Echo* Nachricht zunächst vom Gateway 192.0.2.5 zurückgesendet wird (nach $0,401ms$). Das nächste ICMP *Echo* Paket landet allerdings beim Gateway 192.0.2.2, welches nach $24,022ms$ über den Ablauf der maximalen Lebensdauer des ausgesendeten Paketes informiert. Schließlich meldet sich

[7] Der Begriff „Round-Robin" stammt aus dem Englischen und bezeichnet einen Wettkampf, bei dem jeder gegen jeden kämpft.

beim dritten Versuch wieder das zuerst identifizierte Gateway 192.0.2.5 (nach $4,992ms$). Die Vermutung liegt nun nahe, dass die beiden Gateways parallele Verbindungen zwischen dem dritten und dem fünften Gateway realisieren, und dass diese redundanten Wege abwechselnd verwendet werden. Im weiteren Verlauf bleibt die Verbindungsstrecke zwischen dem Sender und dem Empfänger zunächst statisch. Erst beim elften Gateway, welches unmittelbar vor dem Ziel liegt, lässt sich wieder eine Besonderheit ausmachen. Zur Erklärung dieser Tatsache sei erwähnt, dass es sich beim Netzwerk von „Google“[8] um eine ganz besondere Konstruktion handelt. Die bereits zu Beginn vom TRACEROUTE Befehl ausgegebene Warnung liefert einen Teil der Erklärung für das ungewöhnliche Verhalten des letzten Gateways. Die Meldung

```
traceroute: Warning: www.google.de has multiple addresses;
            using 66.249.85.104
```

deutet auf eine der Besonderheiten des Google-Netzwerkes hin. Unter einem einzigen Domainnamen werden bei Google hunderte von Rechnern zusammengefasst, die wiederum nach einem ausgeklügelten System abwechselnd angesprochen werden. Bei dem in Abbildung 3.20 durchgeführten TRACEROUTE wurde durch Zufall der Domainname `www.google.de` mit der IP Adresse 66.249.85.104 aufgelöst. Außerdem sind die vielen Endgeräte mehrfach mit den verschiedenen Gateways von Google vernetzt, so dass viele verschiedene Pfade zu den Endgeräten existieren. Im Beispiel aus Abbildung 3.20 wurde bei jedem der drei ausgesendeten ICMP *Echo* Pakete ein anderes Gateway als Vermittler zum Zielsystem ausgewählt (216.239.46.47, 66.249.94.136 und 216.239.46.49). Auf diese Weise wird beinahe jedes Mal beim Aufruf der Internetseite `www.google.de` ein anderer Weg zu einem jeweils anderen Endgerät gewählt[9]. Aus den Informationen des TRACEROUTE Befehls lässt sich abschließend die in Abbildung 3.21 dargestellte Netzwerkstruktur ableiten.

UDP TRACEROUTE

Dieses Kapitel handelt zwar von ICMP Werkzeugen, jedoch ist der Befehl TRACEROUTE manchmal auch in der Lage, andere Pakete zu verwenden. Neben der beschriebenen Methode über ICMP *Echo* Nachrichten können einige Implementierungen des TRACEROUTE Befehls auch einen ganz anderen Weg über Pakete des verbindungslosen Protokolls UDP einschlagen. Dazu werden ähnlich der ICMP Methode Pakete mit unterschiedlicher maximaler Lebensdauer erzeugt. Beginnend mit einer TTL von eins wird diese sukzessive erhöht, bis man nicht mehr Antworten von Gateways erhält, welche die Pakete verworfen haben, sondern bis eine Antwort vom Zielsystem empfangen wird. Die

[8] www.google.de

[9] Tatsächlich ist das auch der Trick, mit dem Google eine derart hohe Erreichbarkeit und Verfügbarkeit realisieren kann.

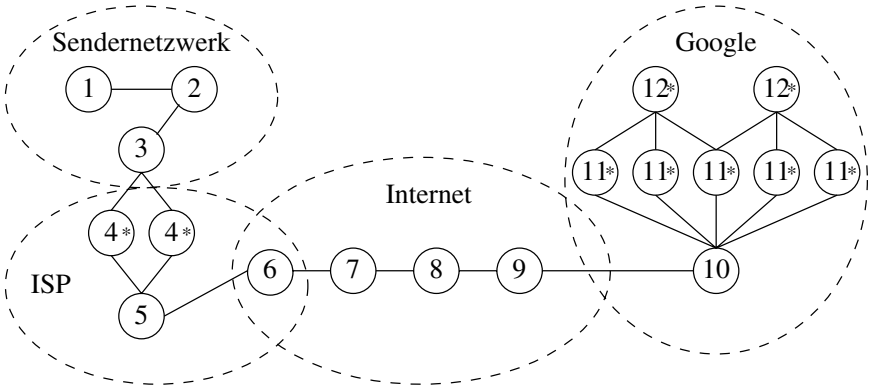

Abb. 3.21. Aus den Ergebnissen des TRACEROUTE Befehls abgeleitete Netzwerkstruktur zwischen Sender und Empfänger (Google). Mit einem Sternchen versehene Gateways sind parallel angelegt und müssen nicht bei jeder Anfrage durchlaufen werden. Die abgebildete schematische Struktur liefert nur eine unvollständige und ungenaue Momentaufnahme der tatsächlichen Struktur.

UDP Pakete werden an einen Port des Zielsystems geschickt, an dem möglichst kein Server auf eingehende Pakete wartet. So lässt sich durch die ICMP Fehlermeldung *port unreachable* (*Typ 3, Code 3*) das Erreichen des Zielsystems erkennen. Die Ausgabe des TRACEROUTE Befehls unter Verwendung von UDP Paketen ist identisch zur Ausgabe bei Verwendung von ICMP Paketen. Es könnte jedoch vorkommen, dass die UDP Pakete einer genaueren Kontrolle durch eine Firewall unterliegen und daher geblockt werden. In diesem speziellen Fall darf der Kommandozeilenparameter `-I` nicht vergessen werden, da dieser eine Umstellung von UDP Paketen auf ICMP Pakete bewirkt.

Prinzipiell sind auch noch Pakete anderer Protokolle zur Realisierung des *Traceroute* Mechanismus denkbar. Beispielsweise könnten TCP Pakete mit gesetztem Synchronisierungs-Flag (SYN) für den Mechanismus verwendet werden. Allerdings nimmt mit steigender Komplexität der Protokolle nicht nur die Wahrscheinlichkeit einer Filterung in entsprechenden Kontrollgeräten zu, sondern auch die Antwortzeiten erhöhen sich durch den steigenden Bearbeitungsaufwand.

3.4.3 TRACEPATH

Auch der Befehl TRACEPATH fällt streng genommen nicht unter die Kategorie ICMP-basierender Werkzeuge. Jedoch ist er eng verwandt mit dem TRACEROUTE Befehl und soll daher in diesem Zusammenhang vorgestellt werden. Aufgabe des TRACEPATH Befehls ist es, die Größe der Maximum Transfer Unit (MTU) auf der gesamten Verbindungsstrecke zwischen Sender und Empfänger zu überprüfen. Auch hier hätte man wieder auf den neuen ICMP Nachrichtentyp *Traceroute* aufbauen können, da dieser explizit ein Datenfeld für die MTU

und die Verbindungsgeschwindigkeit für alle Rückpakete vorsieht, so dass mit einem ausgesendeten Paket diese Informationen für die gesamte Strecke ermittelt werden können. Auch hier hat sich in der Praxis schließlich ein anderer Mechanismus durchgesetzt. Ähnlich dem TRACEROUTE Befehl verwendet der TRACEPATH Befehl Pakete des verbindungslosen Protokolls UDP. Der Mechanismus beruht auf dem Senden von UDP Paketen mit maximal möglicher Größe, welche an einen möglichst unbenutzten Port des Zielrechners gesendet werden. Wie beim TRACEROUTE Befehl auch, wird der Wert für die maximale Lebensdauer TTL sukzessive erhöht und die erhaltenen Fehlermeldungen werden auf ihre Größenangabe und den Status ihrer Fragmentierung hin untersucht. Auf diese Weise lässt sich für jede Teilstrecke auf dem Weg zum Zielsystem ermitteln, ob sie eine kleinere MTU besitzt als die vorherigen Verbindungswege. Als Ergebnis erhält man die MTU in Bezug auf die gesamte Strecke zwischen Sender und Empfänger, die auch Path Maximum Transmission Unit (PMTU) genannt wird. Abbildung 3.22 zeigt die beispielhafte Ausgabe eines TRACEPATH Befehls. Mit dem Kommandozeilenparameter -n kann wie bei den Befehlen PING und TRACEROUTE auch die Auflösung der IP Adressen in Domainnamen unterdrückt werden.

```
nms$ tracepath -n www.google.de
 1:  192.0.2.5          0.230ms pmtu 1500
 1:  192.0.2.1          0.513ms
 2:  212.227.35.193     0.595ms
 3:  212.227.121.198    0.486ms
 4:  212.227.120.38     3.007ms
 5:  no reply
 6:  no reply
 7:  no reply
 8:  no reply
 9:  no reply
10:  no reply
11:  no reply
12:  no reply
nms$
```

Abb. 3.22. Ausgabe des TRACEPATH Befehls zwischen Sender und Empfänger (Google). Da UDP Pakete häufiger einer Filterung an einer Firewall unterliegen, kann es durchaus vorkommen, dass der TRACEPATH Befehl nicht bis zum Ziel vordringen kann. Mit dem Parameter -n wird die Namensauflösung der IP Adressen verhindert.

Wie an der Ausgabe zu erkennen ist, unterliegen die UDP TRACEPATH Pakete den gleichen Problemen, wie die UDP TRACEROUTE Pakete. Während der Senderrechner selbst und die nachfolgenden 4 Gateways noch eine Antwort senden, bricht die Kette auf halbem Weg ab. Das sechste UDP Paket wird

zwar mit der korrekten TTL Zeit ausgesendet, jedoch sendet das entsprechende Gateway entweder nicht mehr die *time to live exceeded* Fehlermeldung, oder aber diese Meldung wird auf dem weiteren Rückweg blockiert. Theoretisch könnte das Problem auch auf ein Gateway mit einer fehlerhaften Implementierung des ICMP Protokolls zurückzuführen sein. In der Vergangenheit gab es häufiger Implementierungsprobleme, bei denen beispielsweise der TTL Wert des Rückpaketes nicht auf den Standardwert gesetzt wurde, sondern auf den verbliebenen Wert des eingegangenen UDP Paketes. In diesem Fall beträgt dieser Wert allerdings bereits Null, so dass die Antwortpakete vom ICMP Typ *time to live exceeded* den Absender nicht mehr erreichen.

Wäre andererseits der Weg zum Ziel ohne eine blockierende Firewall und andere mögliche Problembereiche erreichbar gewesen, so hätte der TRACEPATH Befehl am Ende seiner Messungen eine Zusammenfassung der Ergebnisse inklusive der Angabe der maximalen MTU für die gesamte Wegstrecke ausgegeben. Die folgende Ausgabe weist beispielsweise auf eine PMTU von 1472 Byte hin, welche für die über zwölf Gateways laufende Verbindungsstrecke Gültigkeit besitzt:

```
Resume: pmtu 1472 hops 12 back 12
```

3.4.4 CLOCKDIFF

Der Befehl CLOCKDIFF dient zur Feststellung der Zeitdifferenz zwischen zwei Geräten im Netzwerk. Im engeren Sinne handelt es sich bei CLOCKDIFF nicht um einen Befehl, der rein auf dem Protokoll ICMP aufsetzt. Vielmehr wird mit der IP Option *Internet Timestamp* eines zunächst beliebigen IP Paketes gearbeitet. In der Praxis wird als Nutzlast fast immer eine Kombination aus ICMP *Echo* und *Echo Reply* Paketen gearbeitet. Der Sender schickt also ein „PING“ Paket an den Empfänger und fügt dem IP Paketkopf die spezielle Option *Internet Timestamp* mit der Nummer 4 hinzu. Diese IP Option kann zunächst mit unterschiedlicher Länge spezifiziert werden, jedoch wird beim CLOCKDIFF Befehl im Normalfall entweder mit drei oder mit vier Paaren aus jeweils einer vorgegebenen IP Adresse und einem 32-Bit Wort für den Zeitstempel gearbeitet. Bei Angabe des Kommandozeilenparameters `-o` sind dies vier, bei Angabe des Parameters `-o1` sind es nur drei Paare mit jeweils einer Länge von 8 Oktetten. Abbildung 3.23 verdeutlicht den schematischen Aufbau der IP Option *Internet Timestamp*.

Wie beim ICMP Nachrichtentyp *Traceroute* auf Seite 59 bereits beschrieben, setzt sich das erste Oktett der Option aus drei Feldern zusammen. Die *Internet Timestamp* IP Option trägt die Nummer 4; außerdem dürfen Pakete mit dieser Option nicht fragmentiert werden und zählen zur Klasse *Debugging & Measurement*. Demzufolge erhält man für das erste Oktett den Wert 68. Der Zeiger ist eine redundante Angabe, da er nur ein Zeiger auf das Ende der Option ist, die sich ebenfalls durch die Angabe der Länge für diese Option

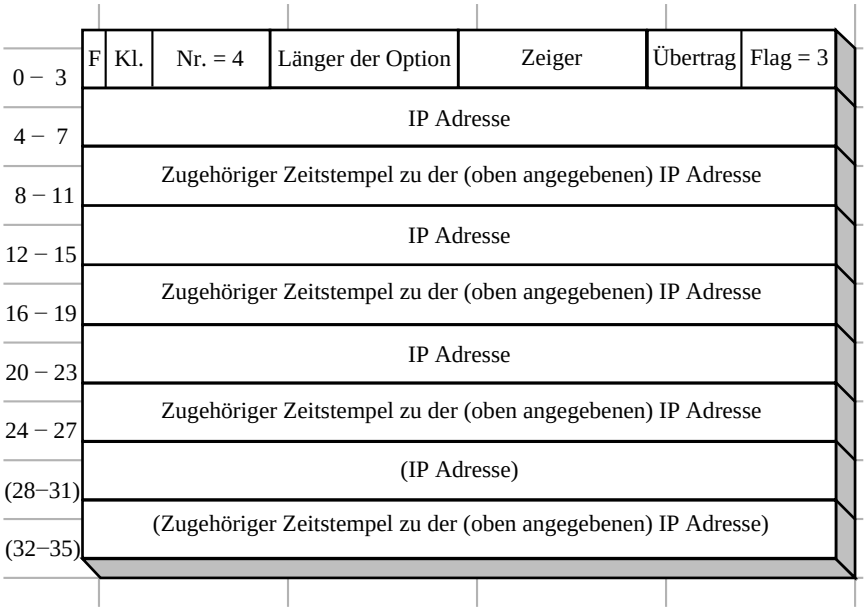

Abb. 3.23. Schematische Darstellung der *Internet Timestamp* Option des Protokolls IP. Das allererste Bit „F" legt fest, dass diese IP Option nicht in Fragmente von Paketen übertragen werden darf, und ist daher auf Null gesetzt. Die nächsten beiden Bit geben Auskunft über die Klasse *Debugging & Measurement* der Option und sind auf 10 gesetzt. Die letzten fünf Bit des ersten Oktetts geben die eindeutige Nummer (00100) der Option an, so dass in allen IP Paketen mit dieser Option das erste Oktett eine 01000100 (68) enthält. Das Flag im vierten Oktett spezifiziert die genaue Art der *Internet Timestamp* Option, wobei in diesem Fall mit vordefinierten IP Adressen gearbeitet wird (Flag = 3). Anschließend folgen drei oder vier Paare aus einer IP Adresse und des dazugehörigen Zeitstempels.

berechnen lässt. Mit dem Datenfeld für den Übertrag können IP Module verfolgt werden, welche der *Internet Timestamp* Option keine Rechnung tragen können. Das Flag im vierten Oktett schließlich steht für die genaue Art der *Internet Timestamp* Option. In Tabelle 3.5 sind die drei möglichen Arten von Optionen aufgeführt, von denen die *Internet Timestamp* Option die dritte Varianten mit vordefinierten IP Adressen verwendet.

Der Sender verschickt nun ein ICMP *Echo* Paket an den Empfänger und gibt in der *Internet Timestamp* Option des IP Paketkopfes drei IP Adressen vor: Zuerst spezifiziert er seine eigene Adresse, dann die Adresse des Empfängers und zum Schluss wieder seine eigene. Den ersten Zeitstempel für den Eintrag mit seiner eigenen IP Adresse füllt der Sender ebenfalls aus. Der zweite Zeitstempel entstammt dem Empfänger, welcher das dazugehörige Datenfeld im Antwortpaket mit dem ICMP *Echo Reply* ausfüllt. Der dritte Zeitstempel wird wieder vom Sender eingefügt. Sind die Werte bekannt, so lässt sich aus

Tabelle 3.5. Die drei möglichen Klassen von IP Optionen des Typs *Internet Timestamp*.

Nummer	Beschreibung
0	Die restlichen Felder bestehen ausschließlich aus Zeitstempeln
1	Die restlichen Felder bestehen aus Paaren IP Adresse/Zeitstempel
3	Die IP Adressen für die folgenden Paare sind vordefiniert

ihnen die durchschnittliche Zeitdifferenz zwischen den beiden Geräten ableiten, die als Ergebnis vom CLOCKDIFF Befehl ausgegeben wird. Abbildung 3.24 zeigt zwei Beispiele für die Ausgabe des CLOCKDIFF Befehls. Abhängig vom Typ des Zielsystems wurde einmal die Variante mit drei Zeitstempeln (Kommandozeilenparameter -o1) und einmal die Variante mit vier Zeitstempeln (Kommandozeilenparameter -o) verwendet. Die verschiedenen Betriebssysteme reagieren unterschiedlich auf den jeweiligen Kommandozeilenparameter, was vor allem am vierten Beispiel aus Abbildung 3.24 deutlich wird.

```
nms$ clockdiff -o1 192.0.2.115
.
host=linux rtt=750(187)ms/0ms delta=-22607ms/-22607ms
                              Sun Jul 3 15:54:10 2005
nms$ clockdiff -o 192.0.2.115
.
host=linux rtt=750(187)ms/0ms delta=-22607ms/-22607ms
                              Sun Jul 3 15:54:10 2005
nms$ clockdiff -o1 192.0.2.202
.
host=win rtt=750(187)ms/0ms delta=480ms/480ms
                             Sun Jul 3 15:54:10 2005
nms$ clockdiff -o 192.0.2.202
wrong timestamps
measure: unknown failure
nms$
```

Abb. 3.24. Ausgabe des CLOCKDIFF Befehls mit zwei Zielsystemen unterschiedlicher Architektur. Mit den Parametern -o und -o1 wird die Anzahl der im Datenfeld vorgegebenen IP Adressen bestimmt.

Zur Berechnung der Zeitdifferenz werden die Werte aus den Datenfeldern der IP Option lediglich subtrahiert. Da die Angaben in Millisekunden erfolgen, wird ein vergleichsweise hohes Auflösungsvermögen erreicht.

4

Simple Network Management Protocol

Grundlage jedes Netzwerkmanagements ist die Kommunikation zwischen den überwachten Geräten und den Managementstationen. Dabei muss prinzipiell zwischen den Kommunikationswegen der Netzwerküberwachung und der Netzwerkkonfiguration unterschieden werden. Bei der Überwachung werden typischerweise nur Informationen von den überwachten Geräten zu den Managementstationen übertragen. Auslöser dafür kann entweder eine konkrete Anfrage der Managementstation sein oder auch ein Ereignis auf der überwachten Komponente, welches eine unaufgeforderte Benachrichtigung der Managementstation bewirkt. Wird das Netzwerk nicht nur überwacht, sondern auch konfiguriert, so fließen zusätzlich Informationen von der Managementstation zu den überwachten Komponenten.

Der wohl bekannteste Kommunikationsmechanismus aus dem Bereich des Netzwerkmanagements ist das Simple Network Management Protocol (SNMP), das sowohl das Überwachen als auch das Konfigurieren von Netzwerkkomponenten erlaubt. SNMP ist seit seiner Einführung im Jahr 1988 um viele Funktionalitäten und Attribute erweitert worden, so dass heute mehrere Versionen von SNMP existieren. Der Fokus in diesem Buch liegt auf SNMPv3, da in dieser Version einige der in den Vorgängerversionen fehlenden wichtigen Sicherheitsmechanismen implementiert wurden. Einen tieferen Einblick vor allem in die älteren SNMP Versionen findet der interessierte Leser in den beiden Referenzwerken [153, 113].

Das Simple Network Management Protocol (SNMP) ist im Grunde ein einfaches und zugleich sehr umfangreiches und mächtiges Rahmenwerk. SNMP basiert auf vergleichsweise einfachen Grundregeln, jedoch wurden in den vergangenen Jahren mittlerweile sehr viele unterschiedliche Regeln aufgestellt, welche die verschiedenen Versionen von SNMP ausmachen. Eine umfangreiche Quelle für detaillierte Informationen zu SNMP finden sich in den vielen Request for Comments (RFCs) der Internet Engineering Task Force (IETF). Eine ausführliche Auflistung aller RFCs zu SNMP findet sich in Anhang A. Zur Veranschaulichung soll Abbildung 4.1 die schematische Struktur des SNMP

Rahmenwerkes skizzieren. Die verwendeten Begriffe werden in den weiteren Abschnitten dieses Kapitels näher erläutert.

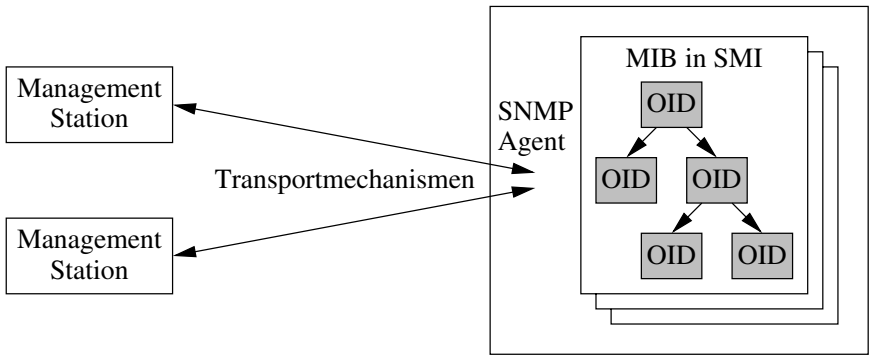

Abb. 4.1. Schematische Darstellung des SNMP Rahmenwerkes. Eine oder mehrere Netzwerkmanagementstationen kommunizieren über definierte Transportmechanismen mit den Netzwerkmanagement Agenten. Die Agenten beinhalten eine oder mehrere in der Definitionssprache SMI erstellte MIBs, die jeweils aus einem hierarchischen Objektbaum aus einzelnen OIDs bestehen.

4.1 Transportmechanismen

SNMP wurde mit dem klaren Ziel entwickelt, die immer komplexer werdenden Netzwerke mit ihrer ständig steigenden Anzahl an Netzwerkkomponenten entfernt über das Netzwerk wartbar zu machen. Der wichtigste Aspekt lag dabei vor allem in der Möglichkeit, von einer entfernten Managementstation aus häufig benötigte Statusinformationen über eine einheitliche Schnittstelle von den Komponenten abfragen zu können. Der zweite wichtige Aspekt lag in der Schaffung einer Möglichkeit, Netzwerkkomponenten aus der Entfernung über das Netzwerk mittels derselben einheitlichen Schnittstelle konfigurieren zu können. Zum damaligen Zeitpunkt war Sicherheit noch kein bedeutendes Thema, so dass entsprechende Mechanismen, die heute unverzichtbar geworden sind, in der ersten Version von SNMP[1] noch nicht enthalten waren.

Die grundlegende Funktionsweise von SNMP basiert im Wesentlichen auf zwei verschiedenen Kommunikationsformen (siehe Abbildung 4.2):

1. Die erste Kommunikationsform verläuft bidirektional und arbeitet nach dem Frage-Antwort-Prinzip. Eine Überwachungsstation stellt bei einer überwachten Netzwerkkomponente eine Anfrage, die von dieser beantwortet wird. Diese Kommunikationsform ermöglicht sowohl das Überwachen

[1] Die erste Version von SNMP wird häufig auch als SNMPv1 bezeichnet.

als auch das Verwalten von Komponenten. Im ersten Fall erhält die Frage als Antwort eine Angabe über den Zustand des abgefragten Teils der Komponente; im zweiten Fall ist es die Bestätigung der Durchführung der angeforderten Zustandsänderung. Ein Beispiel für eine Überwachung wäre die Frage ‚Wieviel Prozent beträgt deine aktuelle Prozessorauslastung?' und die mögliche Antwort ‚23'. Im Fall des Managements könnte die Anfrage ‚Setze den Paketzähler für deine Schnittstelle `eth0` zurück auf Null!' lauten und die Antwort darauf ‚0'. Die erfolgreiche Ausführung der angefragten Managementaufgabe wird dabei oft – wie im obigen Beispiel – einfach dadurch quittiert, dass der angeforderte Wert (0) für das vorgegebene Attribut (Paketzähler für die Schnittstelle `eth0`) zurückgegeben wird.

2. Bei der zweiten Kommunikationsform werden Nachrichten[2] unidirektional – und zwar ausschließlich von der überwachten Komponente hin zur Managementstation – gesendet. Dieser Kommunikationsweg wurde deshalb zusätzlich eingerichtet, damit eine Netzwerkkomponente seine Managementstation über außergewöhnliche Ereignisse zeitnah informieren kann.

Damit SNMP fehlerfrei funktionieren kann, müssen sowohl die Managementstationen als auch die beteiligten Netzwerkkomponenten dieselbe Sprache sprechen. Auf Seiten der unteren Schichten des OSI Referenzmodells ist dies relativ einfach erreicht. SNMP verwendet für seine Kommunikation im Normalfall das verbindungslose User Datagram Protocol (UDP), wobei die Fragen und Antworten der bidirektionalen Kommunikation über den Port 161 und die unidirektionalen Nachrichten über den Port 162 gesendet werden. Die Verwendung eines verbindungslosen Protokolls mag vielleicht auf den ersten Blick seltsam erscheinen, da somit das korrekte und zuverlässige Ausliefern der Pakete nicht garantiert werden kann. Anders ausgedrückt: Stellt die Managementstation eine Anfrage an eine Netzwerkkomponente, so ist weder der Erhalt der Frage für das überwachte System noch der Erhalt der Antwort durch den ursprünglichen Fragesteller gewährleistet. Für diesen Fall ist der Managementstation beziehungsweise der Management-Software zusätzliche Intelligenz zu verleihen. Eine übliche Methode besteht im Festlegen eines Zeitintervalls, innerhalb dessen die Managementstation auf die Antwort der gestellten Anfrage wartet. Erfolgt keine Antwort, so wird die Anfrage oftmals nur wenige Male wiederholt, bis die Managementstation von der Unterbrechung des Kommunikationsweges oder der Nichterreichbarkeit der Netzwerkkomponente ausgeht. Tatsächlich kann die Netzwerkmanagementstation

[2] Diese Nachrichten sind im englischen Sprachgebrauch als sogenannte „traps" bekannt. Vermutlich stammt diese Namensgebung von derjenigen englischen Bedeutung des Wortes „trap", welche für die Wurfmaschine beim Tontaubenschießen steht. Tatsächlich finden sich hier gute Parallelen: Sowohl beim Tontaubenschießen als auch bei einer SNMP-trap Nachricht wird ohne eine äußere Anregung etwas „abgefeuert". Der exakte Zeitpunkt ist dabei für den Schützen bzw. die Managementstation nicht vorhersehbar.

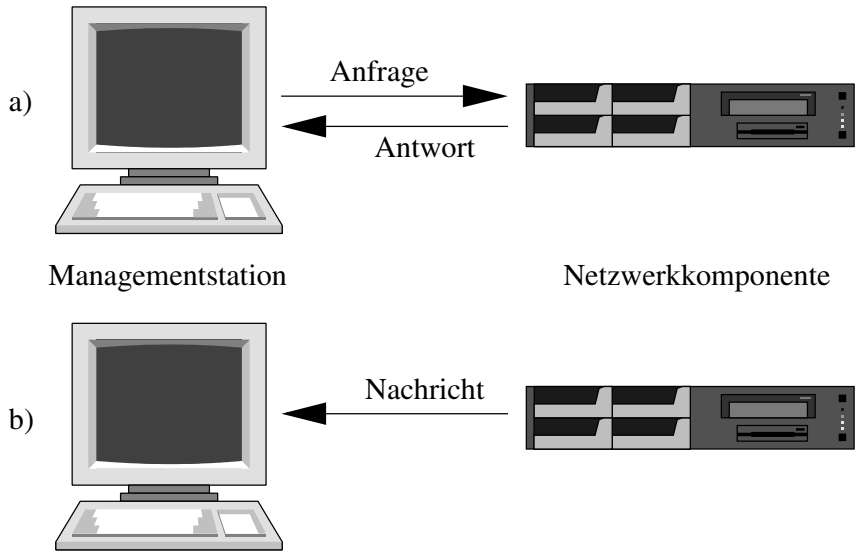

Abb. 4.2. Kommunikationswege beim SNMP Protokoll: a) Von der Managementstation ausgehende Überwachung und Verwaltung nach dem Frage-Antwort-Prinzip. Die Frage kann dabei auch eine Schreibaufforderung darstellen. b) Von der überwachten Netzwerkkomponente ausgehende Nachricht über besondere Ereignisse.

niemals mit Gewissheit feststellen, ob die gesendeten Anfragen bereits auf dem Weg zur Netzwerkkomponente verloren gegangen sind oder ob lediglich die Antworten den Rückweg nicht erfolgreich zurücklegen konnten. Noch gravierender wirkt sich dieser Umstand bei den SNMP Nachrichten aus. Eine Netzwerkkomponente erhält auf eine solche Nachricht laut SNMP-Standard keine Antwort. Somit besitzt die Komponente auch keinerlei Möglichkeit, eine korrekte Auslieferung der Nachricht festzustellen. Da die Managementstation diese Benachrichtigung über ein besonderes Ereignis im Normalfall nicht erwartet, kann diese sehr leicht unbemerkt verloren gehen.

Der Einsatz eines verbindungslosen Protokolls zur Netzwerküberwachung und zur Netzwerkverwaltung hat aber auch klare Vorteile. Die Hauptaufgabe eines Netzwerkes liegt im Regelfall nicht in der Selbstüberwachung, sondern im Transportieren und Ausliefern von Nutzdaten. Zwar ist das Management für den reibungslosen Betrieb des Netzwerkes ausgesprochen wichtig, dennoch sollte es dessen Hauptaufgabe möglichst wenig beeinflussen. Insbesondere bei stark ausgelasteten Netzwerken würde ein verbindungsorientiertes Netzwerkmanagementprotokoll für eine erhebliche Mehrbelastung und negative Beeinträchtigung des Netzwerkes sorgen. Vor allem der Verwaltungsmehraufwand für die Sicherstellung der korrekten Auslieferung der einzelnen Pakete und Nachrichten würde dann bereits ein Vielfaches der eigentlichen SNMP-Inhalte betragen. Dies wird recht schnell deutlich, wenn man das verbindungslose Protokoll UDP mit dem verbindungsorientierten Protokoll TCP vergleicht:

- Der UDP Paketkopf enthält exakt vier Angaben mit jeweils 2 Byte Länge[3]. Eine einzelne Nachricht einer Netzwerkkomponente an die Managementstation besteht auch aus einem einzelnen UDP Paket und erzeugt somit einen Gesamt-Overhead von 8 Byte.
- Der TCP Paketkopf besteht aus wesentlich mehr Informationen als der UDP Paketkopf und kommt auf eine Länge von 24 Byte. Außerdem sorgen sowohl der Verbindungsaufbau als auch der Verbindungsabbau für eine größere Anzahl von Paketen. Beim Drei-Wege-Verbindungsaufbau werden die Daten erst im dritten Paket gesendet, welches dann auch als erstes Paket des Drei-Wege-Verbindungsabbaus zählt. Bei insgesamt fünf Paketen beträgt somit allein der Verwaltungs-Overhead in der Transportschicht schon 120 Byte.

Um die Kommunikation zwischen der Managementstation und den Netzwerkkomponenten zu vereinfachen, sind auf den zu überwachenden Geräten so genannte Agenten installiert. Die begriffliche Anlehnung an die Agenten eines Geheimdienstes liefert gleichzeitig einen anschaulichen Vergleich: Die Überwachungsbehörde (Managementstation) postiert ihre Agenten an wichtigen Knotenpunkten (Netzwerkkomponenten), damit diese vor Ort Informationen sammeln können und entweder auf Anfrage oder bei besonderen Ereignissen auch autark die gesammelten Informationen an die Zentrale übermitteln. Die SNMP-Agenten werden allerdings nicht von der Managementstation ausgesendet, sondern sind von den Herstellern der Netzwerkkomponenten in diese zu implementieren. Die Agenten dienen dann als Kommunikationspartner der Managementstation.

4.2 Object Identifier

Wenn man von Netzwerkmanagement spricht, dann ist damit im Normalfall ein Satz von Objekten und Attributen von Netzwerkkomponenten gemeint, die von einer Managementstation aus überwacht und verwaltet werden. Zwei weiter oben bereits genannte Beispiele für derartige Attribute sind die prozentuale Prozessorauslastung oder der Paketzähler einer Schnittstelle. Jedes einzelne per SNMP verwaltungsfähige Attribut erhält zur Verdeutlichung eine eindeutige Identifikationsnummer, den so genannten Object Identifier (OID). OIDs werden nur genau einmal vergeben, und einmalig vergebene OIDs können im Regelfall auch nicht mehr geändert werden.

[3] UDP und TCP sind Protokolle der Transportschicht (4. Schicht) des OSI Referenzmodells. Die darunterliegenden Schichten kapseln die Pakete der jeweils höheren Schicht und fügen ihre eigenen Zusatzangaben hinzu. Beispielsweise haben die Angaben im Kopf eines IP Paketes der Vermittlungsschicht (3. Schicht des OSI Referenzmodells) eine Länge von 24 Byte, die zur Paketgröße des UDP oder TCP Paketes hinzugefügt werden müssen.

Die verschiedenen Objekte und ihre OIDs sind zur besseren Übersichtlichkeit in einer hierarchischen Struktur angeordnet. Daraus ergibt sich auch die Struktur einer OID: Sie besteht aus einer Kette von Zahlen, die durch einzelne Punkte voneinander getrennt sind. Ein Beispiel für eine OID ist

1.3.6.1.2.1.1.1

Damit die OIDs für den Menschen besser lesbar werden, hat man den Zahlen noch textuelle Pendants zugewiesen, die sich gleichbedeutend verwenden lassen. Ausgehend von der Wurzel der hierarchischen Struktur hat die erste Zahl (1) den Namen *iso*. In der Ebene direkt unterhalb des Elementes *iso* befinden sich weitere Elemente, von denen das dritte (3) den Namen *org* trägt. Schließlich löst sich die im obigen Beispiel angegebene OID *1.3.6.1.2.1.1.5* auf zu der textuellen Form

iso(1).org(3).dod(6).internet(1).mgmt(2).mib(1).system(1).sysName(5)

oder

iso.org.dod.internet.mgmt.mib.system.sysName

Das obige Beispiel beinhaltet also eine OID, welche den administrativen Namen des Systems enthält. Darunter ist der Full Qualified Domain Name (FQDN) des Systems zu verstehen. Damit aber nicht immer der voll ausgeschriebene Hierarchiename der OIDs anzugeben ist, sind OIDs in Gruppen zusammengefasst. Auf das obige Beispiel angewendet findet sich ein Standard, in dem der *iso(1).org(3).dod(6).internet(1).mgmt(2).mib(1)* Zweig des hierarchischen OID Baumes definiert ist. Unterstützt ein SNMP Agent einen derartigen Standard, dann sind ihm auch die Namen der OIDs bekannt. In diesem Fall lässt sich auf dasselbe Objekt ganz einfach über den in der MIB-I (siehe Abschnitt 4.4) definierten Namen *sysName* zugreifen.

4.2.1 Tabellen

Auch Tabellen lassen sich in der hierarchischen Baumstruktur der OIDs abbilden. Dabei erhält allerdings nicht nur jede einzelne Tabellenzelle eine eigene OID, sondern zur Darstellung in einem Baum werden noch weitere OIDs benötigt. Die Tabelle selbst erhält eine OID, die gleichzeitig die Wurzel des Teilbaumes darstellt. Direkt unterhalb der OID für die Tabelle steht ein einzelnes Objekt, in welchem sich die Definitionen für die einzelnen Spalten der Tabelle befinden. Noch eine Stufe weiter unten in der Hierarchie befinden sich dann mehrere Objekte, welche ihrerseits den einzelnen Spalten der Tabelle entsprechen. In der vierten Hierarchiestufe am Ende des MIB Baumes liegt schließlich für jede in der Tabelle vorhandene Zeile ein weiteres Objekt. Diese Objekte symbolisieren die einzelnen Zellen der Tabelle und beinhalten deren Werte. Zugriff auf den Inhalt einer Zelle erhält man demnach über die vierstufige OID-Kette

Tabellenobjekt.Spaltendefinition.Spalte.Zeile

Abbildung 4.3 veranschaulicht noch einmal die vierstufige, hierarchische Darstellung von Tabellen in SNMP MIBs.

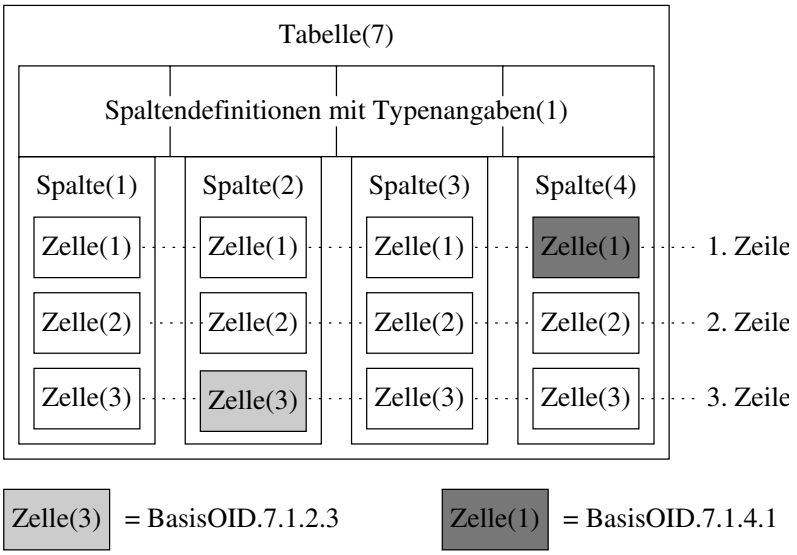

Abb. 4.3. Schematische Darstellung für die hierarchische Repräsentation einer Tabelle in MIB Bäumen. Zur Verdeutlichung wurde ein Beispiel ausgewählt, bei dem die Tabelle als das siebte Objekt unter einer fiktiven *BasisOID* liegt. Für jede der abgebildeten Zellen berechnet sich demnach die OID jeweils aus *BasisOID.Tabelle.Spaltendefinitionen.Spalte.Zelle*.

Die Angabe zur Zeilennummer in der vierten Hierarchiestufe kann sich in manchen Fällen deutlich von den anderen drei Werten unterscheiden. Die OIDs für die Tabelle und das Spaltendefinitionsobjekt direkt darunter sind feste Konstanten. Da weiterhin die einzelnen Spalten bei ihrer Definition von eins beginnend durchnummeriert werden, besteht auch die dritte Hierarchiestufe aus einer Ganzzahl. Bei der Identifizierung der korrekten Zeile aus der Tabelle wird jedoch gänzlich anders verfahren. Die an vierter Stelle stehende Identifikation der Zeile wird nicht einfach durchnummeriert, sondern sie setzt sich aus dem Inhalt aller als Index deklarierten Spalten der gewünschten Zeile zusammen. Zur Verdeutlichung sollen an dieser Stelle drei verschiedene Tabellen eines Netzwerkgerätes beispielhaft beschrieben werden. Neben der logischen „Loopback" Schnittstelle soll dieses Gerät zwei weitere physikalische Schnittstellen aufweisen.

Die erste Tabelle *ifTable* in Abbildung 4.4 aus dem *interfaces* Unterzweig der MIB-I entspricht dem einfachen Beispiel, bei dem in der Tabelle lediglich eine einzige Index-Spalte mit durchlaufend nummerierten Ganzzahlen exis-

tiert (*ifIndex*). In diesem Fall wird das Element der vierten Hierarchiestufe zur Identifikation der Zeile wie die anderen Stufen auch durch die bereits in Abbildung 4.3 gezeigte Ganzzahl repräsentiert.

Mit Ausnahme von Ganzzahlen können Indizes von SNMP Tabellen auch durch komplexere Datentypen repräsentiert werden. Im Falle von IP Adressen beispielsweise besteht jeder Index aus einer Kette von vier Zahlen. Um eine einzelne Zeile einer derartig indizierten Tabelle eindeutig identifizieren zu können, erfolgt als Angabe in der vierten Hierarchiestufe nicht eine einzelne Ganzzahl, sondern die vollständige, aus vier Oktetten bestehende IP Adresse. Tabelle 4.1 listet die möglichen Datentypen für Indizes auf und beschreibt, wie diese zur Spezifizierung der Zeilen dargestellt werden. Nähere Informationen zu den möglichen Datentypen finden sich in Tabelle 4.7.

Tabelle 4.1. Mögliche Datentypen, die als Index in einer SNMP Tabelle verwendet werden dürfen.

Datentyp	Indexdarstellung
Ganzzahlen	Ein zusätzlicher Wert mit der Zahl als Ganzzahl.
Zeichenketten fester Länge	Pro Zeichen der Zeichenkette ein zusätzlicher Wert als Ganzzahl, der das jeweilige Zeichen repräsentiert.
Zeichenketten variabler Länge	Identisch zu den Zeichenketten fester Länge mit dem Unterschied, dass vor die zusätzlichen Oktette ein weiteres Zeichen eingefügt wird, welches die Länge der Zeichenkette angibt.
OIDs	Für jeden Wert der OID ein zusätzlicher Wert mit einem weiteren vorgelagerten Wert, der die Länge der OID Kette spezifiziert.
Netzwerkadressen	Ein Zeichen für den Typ der Netzwerkadresse gefolgt von den einzelnen Zeichen für die jeweilige Netzwerkadresse. Die Anzahl der zusätzlichen Zeichen hängt vom Typ der Netzwerkadresse ab.
IP Adressen	Vier zusätzliche Zeichen, welche die vier Oktette der IP Adresse symbolisieren.

Ein gutes Beispiel für die Verwendung von komplexeren Datentypen für die Indizierung findet sich in der Tabelle *ipAddrTable* aus dem *ip* Unterzweig der MIB-I. Die Index-Spalte *ipAdEntAddr* weist dabei den Datentyp `IpAddress` auf (siehe hierzu auch Tabelle 4.2). Ein Ausschnitt der Tabelle für die Beispiel-Netzwerkkomponente ist in Abbildung 4.5 dargestellt. Es ist deutlich zu erkennen, dass die unterhalb der OIDs für die einzelnen Spalten befindliche Indizierung jeweils aus IP Adressen besteht. Es existieren genau drei Zeilen in der Tabelle *ipAddrTable*, welche über die Indizes `127.0.0.1`, `10.1.54.7` und `172.19.1.8` angesprochen werden. Die Indizes ergeben sich aus dem Inhalt

```
ifTable.ifEntry.ifIndex.1        = 1
ifTable.ifEntry.ifIndex.2        = 2
ifTable.ifEntry.ifIndex.3        = 3
ifTable.ifEntry.ifDescr.1        = "lo0"
ifTable.ifEntry.ifDescr.2        = "eri0"
ifTable.ifEntry.ifDescr.3        = "eri1"
ifTable.ifEntry.ifType.1         = softwareLoopback(24)
ifTable.ifEntry.ifType.2         = ethernetCsmacd(6)
ifTable.ifEntry.ifType.3         = ethernetCsmacd(6)
ifTable.ifEntry.ifMtu.1          = 8232
ifTable.ifEntry.ifMtu.2          = 1500
ifTable.ifEntry.ifMtu.3          = 1500
ifTable.ifEntry.ifSpeed.1        = 127000000
ifTable.ifEntry.ifSpeed.2        = 100000000
ifTable.ifEntry.ifSpeed.3        = 100000000
ifTable.ifEntry.ifPhysAddress.1  = ""
ifTable.ifEntry.ifPhysAddress.2  = "0:3:ba:27:76:20"
ifTable.ifEntry.ifPhysAddress.3  = "0:3:ba:27:76:21"
ifTable.ifEntry.ifAdminStatus.1  = up(1)
ifTable.ifEntry.ifAdminStatus.2  = up(1)
ifTable.ifEntry.ifAdminStatus.3  = up(1)
ifTable.ifEntry.ifOperStatus.1   = up(1)
ifTable.ifEntry.ifOperStatus.2   = up(1)
ifTable.ifEntry.ifOperStatus.3   = up(1)
ifTable.ifEntry.ifLastChange.1   = (0) 0:00:00.00
ifTable.ifEntry.ifLastChange.2   = (0) 0:00:00.00
ifTable.ifEntry.ifLastChange.3   = (0) 0:00:00.00
ifTable.ifEntry.ifInOctets.1     = 0
ifTable.ifEntry.ifInOctets.2     = 492655038
ifTable.ifEntry.ifInOctets.3     = 736585826
ifTable.ifEntry.ifInUcastPkts.1  = 3985612
ifTable.ifEntry.ifInUcastPkts.2  = 3429527
ifTable.ifEntry.ifInUcastPkts.3  = 4184652
 ...
ifTable.ifEntry.ifOutErrors.1    = 0
ifTable.ifEntry.ifOutErrors.2    = 0
ifTable.ifEntry.ifOutErrors.3    = 0
ifTable.ifEntry.ifQLen.1         = 0
ifTable.ifEntry.ifQLen.2         = 0
ifTable.ifEntry.ifQLen.3         = 0
ifTable.ifEntry.ifSpecific.1     = SNMPv2-SMI::zerodotzero
ifTable.ifEntry.ifSpecific.2     = SNMPv2-SMI::zerodotzero
ifTable.ifEntry.ifSpecific.3     = SNMPv2-SMI::zerodotzero
```

Abb. 4.4. Ausschnitt aus der *ifTable* der *interfaces* MIB (siehe Abschnitt 4.4.2) für eine beispielhafte Netzwerkkomponente. Index der Tabelle ist die erste Spalte (*ifIndex*).

der Index-Spalte *ipAdEntAddr*, bei der deshalb auch Index und Inhalt jeder einzelnen Tabellenzelle identisch sind.

```
ipAddrTable.ipAddrEntry.ipAdEntAddr.127.0.0.1          = 127.0.0.1
ipAddrTable.ipAddrEntry.ipAdEntAddr.10.1.54.7          = 10.1.54.7
ipAddrTable.ipAddrEntry.ipAdEntAddr.172.19.1.8         = 172.19.1.8
ipAddrTable.ipAddrEntry.ipAdEntIfIndex.127.0.0.1       = 1
ipAddrTable.ipAddrEntry.ipAdEntIfIndex.10.1.54.7       = 2
ipAddrTable.ipAddrEntry.ipAdEntIfIndex.172.19.1.8      = 3
ipAddrTable.ipAddrEntry.ipAdEntNetMask.127.0.0.1       = 255.0.0.0
ipAddrTable.ipAddrEntry.ipAdEntNetMask.10.1.54.7       = 255.0.0.0
ipAddrTable.ipAddrEntry.ipAdEntNetMask.172.19.1.8      = 255.255.0.0
ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.127.0.0.1     = 0
ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.10.1.54.7     = 0
ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.172.19.1.8    = 0
ipAddrTable.ipAddrEntry.ipAdEntReasmMaxSize.127.0.0.1  = 65535
ipAddrTable.ipAddrEntry.ipAdEntReasmMaxSize.10.1.54.7  = 65535
ipAddrTable.ipAddrEntry.ipAdEntReasmMaxSize.172.19.1.8 = 65535
```

Abb. 4.5. Ausschnitt aus der *ipAddrTable* des *ip* Unterzweigs der MIB-I (siehe Abschnitt 4.4.2) für eine beispielhafte Netzwerkkomponente. Index der Tabelle ist die erste Spalte (*ipAdEntAddr*).

Zu guter letzt darf eine SNMP Tabelle auch mehr als nur einen Index aufweisen. In einem solchen Fall werden die Zeilen durch eine Aneinanderreihung der Inhalte aller Index-Spalten spezifiziert. Ein gutes Beispiel hierfür findet sich in der Tabelle *ipNetToMediaTable* des *ip* Unterzweiges der MIB-I. In dieser Tabelle existieren die beiden Indizes *ipNetToMediaIfIndex* und *ipNetToMediaNetAddress*, die zusätzlich noch unterschiedliche Datentypen aufweisen. Abbildung 4.6 liefert einen Ausschnitt aus der Tabelle *ipNetToMediaTable* für die beispielhaft ausgewählte Netzwerkkomponente. Die Indizierung der einzelnen Tabellenzellen setzt sich aus den letzten fünf Ziffern der OID zusammen, die sich aus den Inhalten der beiden Index-Spalten ableiten. Die erste Ziffer entstammt der ersten Index-Spalte *ipNetToMediaIfIndex*, deren Datentyp eine Ganzzahl ist und auf den Index einer Schnittstelle hinweist, die bereits in Abbildung 4.4 definiert worden ist. Da die Tabelle *ipNetToMediaTable* Zuordnungen von IP Adressen und physikalischen Adressen enthält, welche die Netzwerkkomponente über ihre Schnittstellen in Erfahrung bringen konnte, finden sich dort auch nur Einträge der beiden physikalischen Schnittstellen. Die restlichen vier Ziffern der gezeigten Indizierung werden aus dem Inhalt der zweiten Index-Spalte *ipNetToMediaNetAddress* gewonnen, welche aus der

über das Netzwerk gelernten IP Adresse besteht[4]. Auch in diesem Beispiel lässt sich wieder eindeutig der Zusammenhang zwischen den Indizes und dem Inhalt der Index-Spalten erkennen: In der ersten Spalte *ipNetToMediaIfIndex* sind der erste Teil der Indizes und der Spalteninhalt identisch, in der dritten Spalte *ipNetToMediaNetAddress* sind der zweite Teil der Indizes und der jeweilige Spalteninhalt identisch.

4.3 Structure of Management Information

Um die einzelnen OIDs klar und verständlich definieren zu können, bedarf es einer Sprache mit eindeutiger Syntax. Im Rahmenwerk von SNMP existiert aus diesem Grund die Structure of Management Information (SMI), in der alle OIDs definiert werden[5]. Im Wesentlichen enthält die SMI die grundlegenden Datentypen und Makros, auf die in der Definition einer MIB zurückgegriffen werden kann. Außerdem werden in der SMI Hinweise für eine einheitliche Vergabe von Namen gegeben.

In direktem Zusammenhang mit der SMI stehen zwei weitere Definitionssprachen, die als Grundlage bei der Definition von MIBs und OIDs verwendet werden. Es handelt sich hierbei um die Abstract Syntax Notation One (ASN.1) [99] und die Basic Encoding Rules (BER) [100]. Diese beiden ISO Standards befinden sich vom technischen Standpunkt aus gesehen auf einem noch unterhalb der SMI befindlichen Niveau. Beide Standards stellen im Wesentlichen sicher, dass Managementstationen und Netzwerkkomponenten unabhängig vom Hersteller und der verwendeten Hardware fehlerfrei miteinander kommunizieren können. Die verschiedenen Versionen der SMI sind aus diesem Grund durch einen Ausschnitt der ASN.1 definiert. Damit auf dem Übertragungsweg bei der Zerlegung der Oktette in einzelne Bits und der beim Empfänger erfolgenden Reassemblierung zu Oktetten keine Missverständnisse aufkommen, verwendet ASN.1 für den Transport die BER. Diese regelt beispielsweise auch die Byte-Folge[6] der übertragenen Daten, so dass sie gleichermaßen von „littleendian“ und „big-endian“ Systemen verstanden wird.

[4] Die IP Adresse der Indizierung entspricht nicht der IP Adresse der Schnittstelle. Diese ist ja bereits über den ersten Index eindeutig spezifiziert.

[5] In der Praxis werden nicht einzelne OIDs spezifiziert und definiert, sondern mehrere OIDs sind jeweils zu einer Management Information Base (MIB) zusammengefasst (siehe Abschnitt 4.4). Diese MIBs werden dann in der Definitionssprache SMI spezifiziert.

[6] Im englischen Sprachgebrauch wird hier das Wort „endianess“ verwendet. Es wird behauptet, dass dieser Name von Jonathan Swift's Werk *Gulliver's Reisen* [208] entstammt. Dort werden zwei verfeindete Volksgruppen beschrieben, die sich darüber streiten, von welchem „Ende“ aus ein Ei aufzuschlagen sei. Dabei schwört das eine Lager – die „Big-endians“ – auf die alte Tradition, Eier am größeren Ende aufzuschlagen, während das andere Lager unter Androhung schwerster Strafen darauf besteht, Eier am kleineren Ende aufzuschlagen. Im Bereich der Informationstechnologie gibt es eine ähnliche Unterteilung der Systeme bezüglich der Reihenfolge,

```
ipNetToMediaTable.ipNetToMediaEntry. ...

    ... ipNetToMediaIfIndex.2.10.1.54.7       = 2
    ... ipNetToMediaIfIndex.2.10.47.166.3     = 2
    ... ipNetToMediaIfIndex.2.10.132.4.79     = 2
    ... ipNetToMediaIfIndex.2.10.205.6.23     = 2
    ... ipNetToMediaIfIndex.3.172.19.1.8      = 3
    ... ipNetToMediaIfIndex.3.172.19.9.1      = 3
    ... ipNetToMediaIfIndex.3.172.19.51.3     = 3
    ... ipNetToMediaIfIndex.3.172.19.98.7     = 3
    ... ipNetToMediaPhysAddress.2.10.1.54.7   = "0:3:ba:27:76:20"
    ... ipNetToMediaPhysAddress.2.10.47.166.3 = "0:3:ba:27:77:1a"
    ... ipNetToMediaPhysAddress.2.10.132.4.79 = "0:0:86:4f:8f:00"
    ... ipNetToMediaPhysAddress.2.10.205.6.23 = "0:e0:b6:1:80:00"
    ... ipNetToMediaPhysAddress.3.172.19.1.8  = "0:3:ba:27:76:21"
    ... ipNetToMediaPhysAddress.3.172.19.9.1  = "0:2:a5:b8:79:e5"
    ... ipNetToMediaPhysAddress.3.172.19.51.3 = "0:2:a5:b8:7d:9c"
    ... ipNetToMediaPhysAddress.3.172.19.98.7 = "0:0:cc:4b:39:af"
    ... ipNetToMediaNetAddress.2.10.1.54.7    = 10.1.54.7
    ... ipNetToMediaNetAddress.2.10.47.166.3  = 10.47.166.3
    ... ipNetToMediaNetAddress.2.10.132.4.79  = 10.132.4.79
    ... ipNetToMediaNetAddress.2.10.205.6.23  = 10.205.6.23
    ... ipNetToMediaNetAddress.3.172.19.1.8   = 172.19.1.8
    ... ipNetToMediaNetAddress.3.172.19.9.1   = 172.19.9.1
    ... ipNetToMediaNetAddress.3.172.19.51.3  = 172.19.51.3
    ... ipNetToMediaNetAddress.3.172.19.98.7  = 172.19.98.7
    ... ipNetToMediaType.2.10.1.54.7          = static(4)
    ... ipNetToMediaType.2.10.47.166.3        = dynamic(3)
    ... ipNetToMediaType.2.10.132.4.79        = dynamic(3)
    ... ipNetToMediaType.2.10.205.6.23        = dynamic(3)
    ... ipNetToMediaType.3.172.19.1.8         = static(4)
    ... ipNetToMediaType.3.172.19.9.1         = dynamic(3)
    ... ipNetToMediaType.3.172.19.51.3        = dynamic(3)
    ... ipNetToMediaType.3.172.19.98.7        = dynamic(3)
```

Abb. 4.6. Ausschnitt aus der *ipNetToMediaTable* des *ip* Unterzweigs der MIB-I (siehe Abschnitt 4.4.2) für eine beispielhafte Netzwerkkomponente. Indizes der Tabelle sind die erste und die dritte Spalte (*ipNetToMediaIfIndex* und *ipNetToMediaNetAddress*).

SMI ist zur Zeit in drei Versionen definiert: SMIv1, SMIv2 und SMI der nächsten Generation (SMIng). SMIv2 stellt eine Erweiterung des ursprünglichen Standards dar und wird weitgehend als der aktuelle Standard betrachtet. SMIng besitzt zur Zeit noch experimentellen Charakter und wird noch nicht verwendet.

4.3.1 SMIv1

Die ursprüngliche Structure of Management Information SMIv1 ist 1988 in RFC 1065 [123] aufgestellt worden und ist heute in einer durch RFC 1155 [184] fehlerkorrigierten sowie durch RFC 1212 [185] und RFC 1215 [182] erweiterten Version definiert. Die wichtigsten Angaben in der SMIv1 beschränken sich auf die Definition von Datentypen, die für die Werte der einzelnen verwalteten SNMP Objekte und SNMP Nachrichten verwendet werden können. Zwar ist bereits die neuere Version SMIv2 eingeführt worden, die momentan zur Definition von MIBs verwendet werden soll, jedoch sind noch immer einige MIBs in der alten SMIv1 definiert. Aus diesem Grund soll an dieser Stelle eine detaillierte Beschreibung der SMIv1 folgen.

Definition von SNMP Objekten nach SMIv1

Jedes verwaltete SNMP Objekt besitzt neben seiner eindeutigen OID auch einen eindeutigen Namen, der auch „Object Descriptor“ genannt wird. Um Verwechselungen zu vermeiden, sollte der Name eines SNMP Objektes eindeutig sein. Zur weiteren Erleichterung des Umgangs mit OIDs sollten die vergebenen Namen möglichst gut den beschriebenen Inhalt widerspiegeln. Die Zuweisung von Objektnamen zu OIDs erfolgt in der SMIv1 durch das Makro `OBJECT-TYPE`. Innerhalb dieses Makros können und müssen auch teilweise weitere Spezifizierungen des Objektes vorgenommen werden. Dazu stehen insgesamt sieben verschiedene Angaben zur Verfügung, von denen einige optional sind. Zwingend erforderlich ist jedoch die Spezifizierung des Datentyps für das Objekt über das Schlüsselwort `SYNTAX`.

SYNTAX. Zunächst wird jedem Objekt ein Datentyp (`SYNTAX`) zugewiesen, der seinen Inhalt näher beschreibt. Neben einigen ausgewählten bereits in ASN.1 definierten Grundtypen definiert SMIv1 weitere spezielle Datentypen, die als Syntax verwendet werden dürfen. Tabelle 4.2 listet die in der SMIv1 erlaubten einfachen Datentypen auf; Tabelle 4.3 enthält die beiden für strukturierte Daten erlaubten Typen.

---

in der Bits und Bytes gespeichert werden. Die „Little-endians“ speichern und übertragen das am wenigsten signifikante Byte zuerst, die „Big-endians“ speichern und übertragen das signifikanteste Byte zuerst.

Tabelle 4.2. Einfache Datentypen der Structure of Management Information SMIv1.

Datentyp	Beschreibung
NULL	NULL ist ein spezieller Datentyp, der keinen Inhalt zulässt. In der Praxis wird der NULL-Typ nur als Platzhalter verwendet.
OCTET STRING	Zeichenkette aus beliebig vielen Oktetten (8-Bit Werten). Dieser Datentyp wird häufig für textuelle Beschreibungen eingesetzt. In der Praxis wird dieser Datentyp sehr selten verwendet.
Opaque	Bei Opaque handelt es sich um einen Datentyp, der vor allem für maximale Kompatibilität eingeführt wurde. Mit Opaque lassen sich beliebige ASN.1 kodierte Angaben machen. Es handelt sich also im Wesentlichen um einen speziell kodierten OCTET STRING.
INTEGER	(a) Vorzeichenbehaftete Ganzzahl im Wertebereich von -2^{31} (-2147483648) bis 2^{31}-1 (2147483647). (b) Aufzählungstyp, bei dem der Index aus einer Liste von benannten positiven Ganzzahlen besteht. Die Namen können jeweils aus bis zu 64 Buchstaben, Ziffern und Bindestrichen bestehen. Der Index mit dem Wert 0 sollte nach Möglichkeit vermieden werden, ist aber in der Praxis häufig vergeben worden.
Counter	Der Datentyp Counter (Zähler) basiert auf dem Datentyp INTEGER. Die Hauptunterschiede liegen in der Beschränkung auf positive ganze Zahlen (0 bis 2^{32}-1, 0 bis 4294967295) und in der Tatsache, dass der Zähler nur für monoton steigende Werte verwendet werden darf. Eine besondere Eigenschaft des Zählers ist demnach auch die Möglichkeit eines Überlaufs. Würde der Wert des Zählers auf einen Wert von 2^{32} (4294967296) oder mehr steigen, so findet ein Überlauf statt, bei dem der Zähler wieder bei Null beginnt. Der Startwert des Datentyps Counter ist nicht definiert, so dass nach der Deaktivierung und anschließender Reaktivierung eines Zählers sein Wert unvorhersagbar ist. Um Inkonsistenzen zu vermeiden, sollte der Datentyp Counter nur in Zusammenhang mit einem weiteren Attribut verwendet werden, welches den Zeitpunkt der letzten Initialisierung des Counters beinhaltet. Oftmals kann hierfür der MIB Wert für die Laufzeit des Systems verwendet werden: *sysUpTime* (*1.3.6.1.2.1.1.3*).

(Fortsetzung auf nächster Seite)

Tabelle 4.2. Einfache Datentypen der Structure of Management Information SMIv1 (Fortsetzung)

Datentyp	Beschreibung
Gauge	Wie auch beim Counter liegt der Wertebereich des Datentyps Gauge bei positiven Ganzzahlen (0 bis 2^{32}-1, 0 bis 4294967295). Anders als beim Zähler kann der Wert eines Gauge sowohl steigen als auch sinken; es findet aber niemals ein Überlauf statt. Würde der Wert auf unter Null oder auf 2^{32} (4294967296) und mehr steigen, so wird der Inhalt des Typs Gauge begrenzt und auf das Minimum oder Maximum beschnitten. Die beiden Grenzwerte stellen somit streng genommen keine echten Werte dar, sondern jeweils einen Zustand, bei dem der Grenzwert erreicht oder bereits überschritten ist.
TimeTicks	Wie der Name bereits vermuten lässt, handelt es sich beim Datentyp TimeTicks um einen streng monoton steigenden Zähler. Als Basis dient wie bei Counter und Gauge der Datentyp INTEGER mit einer Beschränkung des Wertebereiches auf positive ganze Zahlen (0 bis 2^{32}-1, 0 bis 4294967295). Ein Objekt vom Typ TimeTicks misst die Anzahl der Hundertstelsekunden seit dem Eintreten eines bestimmten Ereignisses. Dies könnte beispielsweise die letzte Initialisierung des Systems sein. Wie sich leicht ausrechnen lässt, muss ein TimeTicks Zähler ohne Neuinitialisierung zwangsweise nach etwas mehr als 497 Tagen überlaufen.
IpAddress	Auf Grund der anfänglichen Spezialisierung von SNMP auf das IPv4 Protokoll war es durchaus von Vorteil, einen eigenen Datentyp speziell für die Darstellung von IP Adressen einzuführen. Realisiert ist IpAddress durch einen OCTET STRING der Länge 4.
NetworkAddress	Um auch andere Netzwerk-Adressen als IP Adressen zu unterstützen, wurde der generische Datentyp NetworkAddress definiert. Ihm zugrunde liegt der ASN.1 Auswahltyp CHOICE. Es befindet sich lediglich ein einziger Datentyp in dieser Liste, nämlich IpAddress.

DESCRIPTION. Die Angabe `DESCRIPTION` enthält eine Beschreibung des Objektes in Form einer beliebigen Text-Zeichenkette. Zwar ist die Angabe einer Definition in der SMIv1 optional, zur besseren Verständlichkeit sollte jedoch für jedes definierte SNMP Objekt auch eine detaillierte Beschreibung angegeben werden.

REFERENCE. Zu manchen SNMP Objekten finden sich weiterführende Literaturquellen, die nähere Angaben und Erläuterungen enthalten. Die Angabe `REFERENCE` ermöglicht die Definition eines textuellen Verweises, der nicht auf RFCs oder SNMP Literatur beschränkt sein muss. Wie auch die Beschrei-

Tabelle 4.3. Strukturierte Datentypen der Structure of Management Information SMIv1.

Datentyp	Beschreibung
SEQUENCE $< Typ1 >, \ldots, < TypN >$	Mit Hilfe des Datentyps SEQUENCE können in der SMIv1 Listen aus verschiedenen Grundtypen aufgebaut werden. Beispielsweise ist in der MIB-I der Wert *IfEntry* (*1.3.6.1.2.1.2.2.1*) als eine Liste aus insgesamt 21 Einzelwerten definiert: zwei Elemente vom Typ OCTET STRING, fünf Elemente vom Typ INTEGER, elf Elemente vom Typ Counter, zwei Elemente vom Typ Gauge und ein Element vom Typ TimeTicks.
SEQUENCE OF $< List >$	Durch den Datentyp SEQUENCE OF können die mittels SEQUENCE erzeugten Listen zu einer Tabelle erweitert werden. Beim obigen Beispiel aus der MIB-I werden die *IfEntry* Listen zu einer *IfTable* (*1.3.6.1.2.1.2.2*) zusammengefasst.

bung ist die Angabe einer Literaturquelle optional. Es findet sich aber nicht zu jedem SNMP Objekt eine geeignete Referenz, so dass im Gegensatz zur `DESCRIPTION` nicht immer davon Gebrauch gemacht werden kann.

ACCESS. Durch die zwingend vorgeschriebene Angabe von `ACCESS` werden die Zugriffsmöglichkeiten auf ein SNMP Objekt spezifiziert. Im SNMP Rahmenwerk unterscheidet man zwischen drei verschiedene Arten eines Zugriffs: Leseanfragen, Schreibanfragen sowie SNMP Nachrichten, die von einer überwachten Komponente autark versandet werden und den Werteinhalt einer oder mehrerer OIDs enthält. Beim Versenden von SNMP Nachrichten muss der SNMP Agent auf der überwachten Komponente zwar auch den Inhalt von Objekten lesen können, diese Lesezugriffe werden aber von den Leseanfragen der Managementstationen unterschieden. In der Tabelle 4.4 sind die vier möglichen Werte für die Zugriffsbeschränkung erklärt.

STATUS. Die ebenfalls verbindliche Angabe `STATUS` für ein SNMP Objekt definiert im Wesentlichen seine Gültigkeit. Hauptgrund für die Einführung des Attributs `STATUS` war die Schaffung eines Versionierungssystems für die MIBs. SMIv1 erlaubt genau vier Statuszustände, die in Tabelle 4.5 beschrieben sind.

DEFVAL. Die Angabe `DEFVAL` ist nur in Zusammenhang mit Tabellen möglich. Mit `DEFVAL` wird ein Standardwert für ein einzelnes Element einer Tabelle gesetzt, welche die Erzeugung von Tabellenzeilen zulässt. Wird beim Anlegen einer neuen Zeile kein initialer Wert für das entsprechende Unterobjekt mitgeliefert, so wird dieser auf den in `DEFVAL` angegebenen Wert gesetzt.

INDEX. Auch das Attribut `INDEX` kann nur bei Tabellenobjekten angegeben werden. Die Angabe ist zwingend erforderlich und gibt dasjenige oder dieje-

Tabelle 4.4. Zugriffsbeschränkungen auf SNMP Objekte in der Structure of Management Information SMIv1.

Zugriffsbeschränkung	Beschreibung
read-only	Der Wert des Objektes darf gelesen werden. Dies bedeutet auch, dass beim Senden einer SNMP Nachricht auf den Inhalt dieses Objektes zugegriffen werden darf. Das SNMP Objekt darf jedoch nicht über einen SNMP Befehl verändert werden.
read-write	Das Objekt darf ohne Einschränkung sowohl ausgelesen als auch verändert werden.
write-only	Der Inhalt des Objektes darf verändert werden. Die Definition eines ausschließlich beschreibbaren Objektes, dessen Wert noch nicht einmal ausgelesen werden darf, hat sich jedoch als unpraktisch erwiesen und gilt als veraltet.
not-accessible	Ein Objekt mit dieser Zugriffsbeschränkung darf weder gelesen noch beschrieben werden. Dies bietet sich vor allem bei der Verwendung von SNMP Tabellen an. Hier müssen lediglich die einzelnen Tabellenzellen zugreifbar sein, nicht aber das Tabellenobjekt oder das Objekt zur Spaltendefinition.

Tabelle 4.5. Statuszustände für Objekte der Structure of Management Information SMIv1.

Statuszustand	Beschreibung
mandatory	Die Implementierung eines Objektes mit dem Status mandatory ist für eine Konformität mit der jeweiligen MIB zwingend erforderlich.
obsolete	Besitzt ein SNMP Objekt den Status obsolete, so gilt es als veraltet und eine Implementierung ist weder vorgeschrieben noch empfohlen.
deprecated	Objekte mit dem Status deprecated gelten zwar als veraltet und überholt, sind aber für eine Konformität mit einer MIB dennoch zu implementieren.
optional	Der Statuszustand optional beschreibt gültige Objekte, die aber für eine volle Konformität zur jeweiligen MIB nicht notwendigerweise implementiert werden müssen. Da dieser Status leicht zu Problemen bei der Kompatibilität führen kann, wurde er später in der SMIv2 wieder verworfen und sollte daher auch nicht verwendet werden.

nigen Listenelemente an, welche den Index einer Tabelle und der einzelnen Zeilen repräsentieren.

Beispiel für die Definition eines SMIv1 konformen SNMP Objektes

Abbildung 4.7 zeigt ein Beispiel für die Definition eines SNMP Objektes aus der MIB-I unter Verwendung aller erlaubten Attribute. Das abgedruckte Beispiel zeigt einen kurzen Ausschnitt aus der MIB für das dynamische Routing Protokoll OSPF in der Version 2 (OSPF 2) aus RFC 1248 [10]. Man lasse sich nicht vom Schreibfehler bei der Angabe des `INDEX` verwirren (der in der aktuellen OSPF 2 MIB in RFC 1850 [11] selbstverständlich bereits von `ospfAreaID` zu `ospfAreaId` korrigiert ist).

Definition von SNMP Nachrichten nach SMIv1

Zur Definition von SNMP Nachrichten wird das Makro `TRAP-TYPE` verwendet. Ähnlich dem `OBJECT-TYPE` Makro können auch hier verschiedene Angaben zu der SNMP Nachricht gemacht werden. Dem Makro `TRAP-TYPE` wird allerdings im Gegensatz zu den Objekten keine OID sondern eine positive Ganzzahl zugewiesen.

ENTERPRISE. Die einzige zwingend vorgeschriebene Angabe einer SNMP Nachricht ist das Attribut `ENTERPRISE`. In diesem Feld steht eine OID, die auf den Sender dieser Nachricht hinweist. Es existiert eine Ausnahme: Wenn für das `ENTERPRISE` Attribut die OID *snmp* (*1.3.6.1.2.1.11*) vergeben wird, so wird dieser Wert ausgetauscht durch den Wert der OID *sysObjectID* (*1.3.6.1.2.1.1.2*). Gleichzeitig gilt die SNMP Nachricht in diesem Spezialfall als eine generische Nachricht („generic-trap"), während sie in allen anderen Fällen als spezielle Nachricht („specific-trap") interpretiert wird. Dies hat direkte Auswirkungen auf die erzeugte Nachricht:

- Handelt es sich um eine generische Nachricht (`ENTERPRISE` = *snmp*), so wird die dem `TRAP-TYPE` zugewiesene Ganzzahl in das `generic-trap` Feld kopiert. Gleichzeitig wird das `specific-trap` Feld auf den Wert 0 gesetzt.
- Handelt es sich um eine spezifische Nachricht (`ENTERPRISE` $\neq$ *snmp*), so wird die dem `TRAP-TYPE` zugewiesene Ganzzahl in das `specific-trap` Feld kopiert. Gleichzeitig wird das `generic-trap` Feld auf den Wert 6 (`enterpriseSpecific`) gesetzt.

VARIABLES. Alle im optionalen `VARIABLES` Attribut aufgeführten OIDs mit ihren Werten werden in das `variable-bindings` Feld der SNMP Nachricht überführt. Auf diese Art lassen sich auch mehrere Werte in einer SNMP Nachricht übergeben.

DESCRIPTION. Die Angabe `DESCRIPTION` enthält eine Beschreibung der SNMP Nachricht in Form einer beliebigen Text-Zeichenkette. Zu beachten

```
ospfAreaTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF OspfAreaEntry
    ACCESS   not-accessible
    STATUS   mandatory
    DESCRIPTION
       "Information describing the configured parameters and
       cumulative statistics of the router's attached areas."
    REFERENCE
       "OSPF Version 2, Section 6  The Area Data Structure"
    ::= { ospf 2 }

ospfAreaEntry OBJECT-TYPE
    SYNTAX   OspfAreaEntry
    ACCESS   not-accessible
    STATUS   mandatory
    DESCRIPTION
       "Information describing the configured parameters and
       cumulative statistics of one of the router's attached
       areas."
    INDEX { ospfAreaID }
    ::= { ospfAreaTable 1 }

OspfAreaEntry ::=
    SEQUENCE {
        ospfAreaId
            AreaID,
        ospfAuthType
            INTEGER,
        ospfImportASExtern
            TruthValue,
        ospfSpfRuns
            Counter,
        ospfAreaBdrRtrCount
            Gauge,
        ospfASBdrRtrCount
            Gauge,
        ospfLSACount
            Gauge,
        ospfAreaLSACksumSum
            INTEGER
    }
```

Abb. 4.7. Ausschnitt aus der MIB-I Definition des dynamischen Routing Protokolls Open Shortest Path First in der Version 2.

ist an dieser Stelle, dass in der SMIv1 die Angabe einer Definition optional ist. Lediglich wenn an keiner anderen Stelle Informationen über die Nachricht vorhanden sind, wird die Beschreibung verpflichtend.

REFERENCE. `REFERENCE` ist wie auch die Angabe einer `DESCRIPTION` optional. Hinter diesem Attribut findet sich ein textueller Verweis auf eine andere Literaturquelle, in der nähere Angaben zur Nachricht oder dem Ereignis gemacht werden.

Beispiel für die Definition einer SMIv1 konformen SNMP Nachricht

Abbildung 4.8 zeigt ein Beispiel für die Definition einer SNMP Nachricht aus der MIB-II unter Verwendung aller erlaubten Attribute. Das abgedruckte Beispiel zeigt einen gekürzten Ausschnitt aus der MIB für Repeater aus RFC 1368 [126].

4.3.2 SMIv2

Fünf Jahre nach der Veröffentlichung der SMIv1 wurde eine neue Version der Structure of Management Information (SMIv2) in RFC 1442 [29] aufgestellt, die anschließend noch durch RFC 1902 [34] und RFC 2578 [120] korrigiert sowie durch RFC 2579 [121] und RFC 2580 [119] erweitert wurde. Dabei wurden eine Reihe von Änderungen und Neuerungen gemacht, von denen die wichtigsten

- zwei neue MIB-Zweige unterhalb der OID *internet* (*1.3.6.1*), insbesondere der Zweig *snmpV2* (*internet.6*),
- mehrere neue Datentypen für das `SYNTAX` Attribut eines SNMP Objektes und
- eine Redefinition und Verbesserung verschiedener anderer Attribute von Objekten und SNMP Nachrichten sind.

Die Änderungen der SMIv2 sind außerdem so angelegt, dass sich die meisten der nach SMIv1 definierten MIBs vergleichsweise einfach in die neue SMIv2 übersetzen lassen. Im Folgenden sollen deshalb die Unterschiede der beiden Standards der Structure of Management Information durch eine entsprechende Übersetzungsanleitung beschrieben werden.

Definition von SNMP Objekten nach SMIv2

Wie bei SMIv1 wird auch bei der SMIv2 zur näheren Spezifizierung der SNMP Objekte das Makro `OBJECT-TYPE` verwendet. Wichtige grundsätzliche Neuerung ist die Tatsache, dass SMIv2 keinen Bindestrich („-“) in den Objektnamen mehr erlaubt. Außerdem wurden einige der möglichen Angaben zu einem Objekt geändert und neue hinzugefügt.

```
snmpDot3RptrMgt OBJECT IDENTIFIER ::= { mib-2 22 }

rptrOperStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                other(1),           -- undefined or unknown status
                ok(2),              -- no known failures
                rptrFailure(3),     -- repeater-related failure
                groupFailure(4),    -- group-related failure
                portFailure(5),     -- port-related failure
                generalFailure(6)   -- failure, unspecified type
            }
    ACCESS    read-only
    STATUS    mandatory
    DESCRIPTION
           "The rptrOperStatus object indicates the
           operational state of the repeater.  The
           rptrHealthText object may be consulted for more
           specific information about the state of the
           repeater's health..."
    REFERENCE
            "Reference IEEE 802.3 Rptr Mgt, 19.2.3.2,
            aRepeaterHealthState."
    ::= { rptrRptrInfo 2 }

rptrHealth TRAP-TYPE
    ENTERPRISE  snmpDot3RptrMgt
    VARIABLES   { rptrOperStatus }
    DESCRIPTION
            "The rptrHealth trap conveys information related
            to the operational status of the repeater.  This
            trap is sent only when the oper status of the
            repeater changes.

            The rptrHealth trap must contain the
            rptrOperStatus object.  The agent may optionally
            include the rptrHealthText object in the varBind
            list..."
    REFERENCE
            "Reference IEEE 802.3 Rptr Mgt, 19.2.3.4,
            hubHealth notification."
    ::= 1
```

Abb. 4.8. Ausschnitt aus der MIB-II Definition für Repeater.

MODULE-IDENTITY. In der SMIv2 ist für jede MIB ein neues Konstrukt vorgeschrieben, nämlich die Angabe von `MODULE-IDENTITY`. In diesem Block sollen vorrangig Informationen hinterlegt werden, welche die MIB, ihre Autoren und die Version der MIB betreffen. Die zugehörigen Informationsfelder sind in Tabelle 4.6 aufgelistet.

Tabelle 4.6. Angaben innerhalb des MODULE-IDENTITY Konstrukts der Structure of Management Information SMIv2.

Schlüsselwort	Beschreibung
LAST-UPDATED	Dieses Feld enthält die Zeitangabe der letzten Änderung des gesamten Moduls. Wenn die MIB und damit auch dieses Feld geändert wird, so sollte ebenfalls ein neuer REVISION Eintrag mit gleichem Zeitstempel folgen.
REVISION	Angabe eines Zeitstempels im UTC Format, der für die aktuelle oder eine historische Version der MIB steht. Jedem REVISION Eintrag folgt eine DESCRIPTION mit besonderen Angaben über die jeweilige Version.
DESCRIPTION	(a) Detaillierte Beschreibung des MIB-Moduls (b) Nach einer REVISION Angabe enthält der DESCRIPTION Block Informationen über die jeweilige Version.
ORGANIZATION	Angabe über den Autor und Besitzer der MIB.
CONTACT-INFO	Verschiedene Angaben, die eine Kontaktaufnahme mit dem Autor oder dem Besitzer der MIB ermöglichen.

Innerhalb einer `MODULE-IDENTITY` Angabe können beliebig viele `REVISION` Blöcke definiert werden, die jedoch jeweils von einer eigenen `DESCRIPTION` Angabe gefolgt werden müssen. Die beiden Einträge `LAST-UPDATED` und `REVISION` erfolgen im Universal Time Constant (UTC) Format und werden in einem OCTET STRING kodiert. Die ersten vier Zeichen[7] geben das Jahr an; danach folgen jeweils 2 Zeichen für den Monat, den Tag des Monats, die Stunde und die Minute des Zeitpunkts. Das letzte Zeichen ist ein „Z“ und symbolisiert die Zeitzone „Greenwich Mean Time“ (GMT).

SYNTAX. Sowohl bei SMIv1 als auch bei SMIv2 wird jedem Objekt zunächst ein Datentyp (`SYNTAX`) zugewiesen, der seinen Inhalt näher beschreibt. Art und Anzahl der erlaubten Datentypen haben sich von SMIv1 zu SMIv2 zum Teil stark geändert. Tabelle 4.7 beschreibt alle in der aktuellen Structure of Management Information erlaubten Datentypen. Beim einzigen in SMIv2 nicht mehr erlaubten Datentyp handelt es sich um `NetworkAddress`, der auf Grund seiner zu engen Limitierungen verworfen wurde.

[7] Bei Jahreszahlen vor 2000 reichen gemäß der ursprünglichen Definition der SMI auch die letzten beiden Zeichen des Jahres als Angabe.

Tabelle 4.7. Einfache Datentypen der Structure of Management Information SMIv2.

Datentyp	Beschreibung
NULL	NULL wird zwar in der SMIv2 nicht mehr explizit erwähnt (mit Ausnahme der Erwähnung in der Liste der reservierten Schlüsselworte), ist aber immer noch als der spezielle Datentyp ohne Inhalt zu verstehen. In der Praxis wird der NULL-Typ als Platzhalter oder in Fehlerfällen verwendet.
OCTET STRING	Zeichenketten aus Oktetten (8-Bit Werten) werden nach wie vor mit dem Datentyp OCTET STRING dargestellt. In der SMIv2 wird zusätzlich ein Hinweis auf mögliche Probleme von Zeichenketten mit mehr als 255 Zeichen hingewiesen.
Opaque	Opaque ist unverändert als ein Datentypen für beliebige ASN.1 kodierte Angaben definiert. In der SMIv2 gilt Opaque als veraltet und sollte nicht mehr verwendet werden.
BITS	Anders als der Datentyp OCTET STRING ermöglicht BITS nicht nur die Kodierung von beliebigen Oktetten, sondern beim BITS Typ kann jedes einzelne der Bits auch mit einem eigenen Namen versehen werden. Die interne Darstellung läuft dann wieder über den Datentyp OCTET STRING.
INTEGER	Der Basistyp INTEGER besitzt weiterhin zwei mögliche Interpretationen: (a) Vorzeichenbehaftete Ganzzahl im Wertebereich von -2^{31} (-2147483648) bis 2^{31}-1 (2147483647) (b) Aufzählungstyp, bei dem der Index aus einer Liste von benannten positiven Ganzzahlen besteht. Die Namen können jeweils aus bis zu 64 Buchstaben oder Ziffern bestehen, wobei auch hier der Bindestrich („-“) nicht mehr erlaubt ist.
Integer32	Zusätzlich zum Datentyp INTEGER wurde der Typ Integer32 eingeführt, der vorzeichenbehaftete Ganzzahlen im Wertebereich von -2^{31} (-2147483648) bis 2^{31}-1 (2147483647) aufnehmen kann.
Counter32	Der Datentyp Counter32 ist ein vollwertiger Ersatz für den in SMIv2 nicht mehr vorhandenen Datentyp Counter. Die Definition des Counter32 ist identisch zum Counter und beschreibt eine positive ganze Zahl (0 bis 2^{32}-1, 0 bis 4294967295), die nur für monoton steigende Werte verwendet werden darf. Auch die Möglichkeit eines Überlaufs existiert beim Counter32.

(Fortsetzung auf nächster Seite)

Tabelle 4.7. Einfache Datentypen der Structure of Management Information SMIv2 (Fortsetzung)

Datentyp	Beschreibung
Counter64	Counter64 ist eine Erweiterung des Datentyps Counter32 auf die doppelte Bit-Tiefe. Die Definition ist identisch zum Counter32, nur dass der Wertebereich von Null bis 2^{64}-1 (18.446.744.073.709.551.615 $\approx$ 1.8e19) liegt. Aus Kompatibilitätsgründen sollte Counter64 ausschließlich dort verwendet werden, wo unter normalen Bedingungen der Counter32 in weniger als einer Stunde überlaufen kann.
Gauge32	Analog zu Counter und Counter32 ist der Datentyp Gauge aus der SMIv1 durch den Typ Gauge32 vollwertig ersetzt worden. Der Wertebereich von Gauge32 liegt daher ebenfalls bei positiven Ganzzahlen (0 bis 2^{32}-1, 0 bis 4294967295), die sowohl steigen als auch sinken können, ohne dass ein Überlauf stattfindet.
Unsigned32	Trotz eines separaten Namens ist der neue Datentyp Unsigned32 identisch zu den Datentypen Gauge (SMIv1) und Gauge32 (SMIv2).
TimeTicks	Dieser Datentyp gibt nach wie vor die Anzahl der Hundertstelsekunden seit dem Eintreten eines bestimmten Ereignisses wieder.
IpAddress	Auch in der SMIv2 ist der Datentyp IpAddress als ein OCTET STRING mit einer Länge 4 definiert. Objekte des in der SMIv2 nicht mehr verwendeten Datentyps NetworkAddress müssen soweit möglich ebenfalls durch IpAddress ausgedrückt werden.

Die strukturierten Datentypen `SEQUENCE` und `SEQUENCE OF` sind auch in der SMIv2 noch vorhanden.

DESCRIPTION. Die Angabe `DESCRIPTION` enthält eine Beschreibung des Objektes in Form einer beliebigen Text-Zeichenkette. In der SMIv2 ist diese Angabe nicht mehr optional und muss für jede aus der SMIv1 übersetzte Definition hinzugefügt werden.

REFERENCE. Zwischen SMIv1 und SMIv2 bestehen keinerlei Unterschiede beim Attribut `REFERENCE`. In beiden Standards beschreibt es einen textuellen Verweis auf eine andere Literaturquelle, in der nähere Angaben zum Objekt gemacht werden. Die Angabe einer `REFERENCE` ist unverändert optional.

MAX-ACCESS. Durch die nicht ganz eindeutige Definition der Angabe `ACCESS` in der SMIv1 wurde eine Neueinführung des Attributs `MAX-ACCESS` notwendig. Tabelle 4.8 beschreibt die in der SMIv2 gültigen Werte für die Zugriffsbeschränkung. Für eine Übersetzung von SMIv1 MIBs nach SMIv2 sollte

man darauf achten, dass alle Tabellen und alle Listenelemente eine Zugriffsbeschränkung von `not-accessible` erhalten.

Tabelle 4.8. Zugriffsbeschränkungen auf SNMP Objekte in der Structure of Management Information SMIv2.

Zugriffsbeschränkung	Beschreibung
read-only	Der Wert des Objektes darf gelesen werden. Dies bedeutet auch, dass eine SNMP Nachricht mit dem Wert als Inhalt gesendet werden darf. Das Objekt darf aber nicht via SNMP Befehl verändert werden.
accessible-for-notify	Im Unterschied zur Berechtigung read-only sind alle Werte mit der Zugriffsbeschränkung accessible-for-notify ausschließlich für die Versendung von SNMP Nachrichten lesbar, nicht aber für einfache Leseanfragen.
read-write	Das Objekt ist für Leseanfragen lesbar und außerdem für Schreibanfragen beschreibbar.
read-create	Die Zugriffsbeschränkung read-create wird nur auf Tabellen angewendet. Die create Berechtigung erlaubt daher das Erstellen von neuen Tabellenzeilen.
not-accessible	Ein Objekt mit dieser Zugriffsbeschränkung darf weder gelesen noch beschrieben werden. Dies ist für Tabellen und Listen vorgeschrieben.

STATUS. Die verbindliche Angabe `STATUS` für ein SNMP Objekt definiert im Wesentlichen seine Gültigkeit. Im Vergleich zu SMIv1 wurden einige Änderungen zu den möglichen Zuständen gemacht. Tabelle 4.9 liefert Informationen über die drei möglichen Werte. Für eine Übersetzung von SMIv1 nach SMIv2 reicht es in erster Näherung aus, den nicht mehr unterstützten Wert `mandatory` durch `current` und den ebenfalls nicht mehr erlaubten Wert `optional` durch einen der beiden Zustände `current` oder `obsolete` zu ersetzen.

Tabelle 4.9. Statuszustände für Objekte der Structure of Management Information SMIv2.

Statuszustand	Beschreibung
current	Aktuell gültige Definition.
deprecated	Objekte mit dem Status deprecated gelten als veraltet und überholt.
obsolete	Besitzt ein SNMP Objekt den Status obsolete, so gilt es als veraltet und eine Implementierung ist nicht empfohlen.

DEFVAL. Die Angabe DEFVAL ist identisch zur Implementierung in SMIv1. Mit DEFVAL wird ein Standardwert für ein einzelnes Element einer Tabelle gesetzt, die die Erzeugung von Tabellenzeilen zulässt. Wird beim Anlegen einer neuen Zeile kein initialer Wert für das entsprechende Unterobjekt mitgeliefert, so kann dieser auf den in DEFVAL angegebenen Wert gesetzt werden. SMIv2 erlaubt eine Angabe von DEFVAL auch für skalare Objekte, die dynamisch erzeugt werden.

INDEX. Wie schon bei der SMIv1 kann auch das Attribut INDEX nur bei Tabellenobjekten angegeben werden. Für Basistabellen, die keine direkte Abhängigkeit zu einer anderen Tabelle besitzen, ist die Angabe zwingend erforderlich. Mit dem INDEX Attribut werden dasjenige oder diejenigen Listenelemente angegeben, welche den Index einer Tabelle und der einzelnen Zeilen repräsentieren. Handelt es sich um eine Erweiterungstabelle, welche die vorhandenen Angaben einer Basistabelle direkt erweitert, so wird in der SMIv2 die Verwendung von AUGMENTS empfohlen.

AUGMENTS. Oftmals enthalten MIBs mehrere Tabellen, die einen gemeinsamen Satz an Zeilenelementen teilen. Eine solche Relation, bei der jede Zeile der ersten Tabelle direkt mit genau einer Zeile der zweiten Tabelle zusammenhängt und umgekehrt, lässt sich nach der SMIv2 mittels des AUGMENTS Attributs vereinfachen. Eine Basistabelle, die grundlegende Informationen zu den jeweiligen Elementen enthält, wird mit einem oder mehreren Indizes über entsprechende INDEX Einträge definiert. Die zweite Tabelle, die auch optionale Angaben enthalten kann, ist als konsequente Erweiterung der ersten Tabelle anzusehen. Für diese Erweiterungstabelle gibt man als Operanden für das AUGMENTS Attribut das Zeilenobjekt der Basistabelle an. Beide Tabellen verbindet derselbe Index aus der Basistabelle.

UNITS. Mit der optionalen Angabe von UNITS kann für ein Objekt eine textuelle Beschreibung der zum Wert gehörenden Einheit angegeben werden.

Beispiel für die Definition eines SMIv2 konformen SNMP Objektes

Abbildung 4.9 zeigt ein Beispiel für die Definition eines SNMP Objekts aus der MIB-II unter Verwendung aller erlaubten Attribute. Das abgedruckte Beispiel zeigt einen gekürzten und für eine bessere Anzeige leicht abgeänderten Ausschnitt aus der MIB für das Virtual Router Redundancy Protocol (VRRP) aus RFC 2787 [103].

Definition von SNMP Nachrichten nach SMIv2

Das in der SMIv1 zur Definition von SNMP Nachrichten verwendete Makro TRAP-TYPE ist in der SMIv2 grundlegend überarbeitet worden. Neben den zahlreichen Änderungen und Verbesserungen wurde der Eintrag deshalb auch umbenannt und verbirgt sich hinter dem Schlüsselwort NOTIFICATION-TYPE.

```
vrrpMIB  MODULE-IDENTITY
    LAST-UPDATED "200003030000Z"
    ORGANIZATION "IETF VRRP Working Group"
    CONTACT-INFO
           "Brian R. Jewell
    Postal: Copper Mountain Networks, Inc.
            2470 Embarcadero Way
            Palo Alto, California 94303
    Tel:    +1 650 687 3367
    E-Mail: bjewell@coppermountain.com"

    DESCRIPTION
        "This MIB describes objects used for managing Virtual
         Router Redundancy Protocol (VRRP) routers."
    REVISION "200003030000Z"    -- 03 Mar 2000
    DESCRIPTION "Initial version as published in RFC 2787."
    ::= { mib-2 68 }

vrrpOperEntry OBJECT-TYPE
    SYNTAX       VrrpOperEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry in the vrrpOperTable containing the operational
         characteristics of a virtual router. On a VRRP router,
         a given virtual router is identified by a combination
         of the IF index and VRID..."

    INDEX    { ifIndex, vrrpOperVrId }
    ::= { vrrpOperTable 1 }

vrrpRouterStatsEntry OBJECT-TYPE
    SYNTAX       VrrpRouterStatsEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry in the table, containing statistics information
        about a given virtual router."
    AUGMENTS    { vrrpOperEntry }
    ::= { vrrpRouterStatsTable 1 }
```

Abb. 4.9. Ausschnitt aus der MIB-II Definition des Virtual Router Redundancy Protocols.

```
VrrpOperEntry ::=
    SEQUENCE {
        vrrpOperVrId
            VrId,
        vrrpOperVirtualMacAddr
            MacAddress,
        vrrpOperState
            INTEGER,
        vrrpOperAdminState
            INTEGER,
        vrrpOperPriority
            Integer32,
        vrrpOperIpAddrCount
            Integer32,
        vrrpOperMasterIpAddr
            IpAddress,
        vrrpOperPrimaryIpAddr
            IpAddress,
        vrrpOperAuthType
            INTEGER,
        vrrpOperAuthKey
            OCTET STRING,
        vrrpOperAdvertisementInterval
            Integer32,
        vrrpOperPreemptMode
            TruthValue,
        vrrpOperVirtualRouterUpTime
            TimeStamp,
        vrrpOperProtocol
            INTEGER,
        vrrpOperRowStatus
            RowStatus
}
```

Abb. 4.9. Ausschnitt aus der MIB-II Definition des Virtual Router Redundancy Protocols (Fortsetzung).

Einem Objekt vom Typ `NOTIFICATION-TYPE` wird nicht wie dem `TRAP-TYPE` Objekt der SMIv1 eine Ganzzahl zugewiesen, die dann über den `ENTERPRISE` Eintrag an eine OID gebunden wird. Die SMIv2 verwendet das `ENTERPRISE` Attribut nicht mehr; im Gegenzug wird dem `NOTIFICATION-TYPE` Objekt direkt die passende OID des Ereignisses zugewiesen.

OBJECTS. Unverändert existiert bei den SMIv2 die Möglichkeit, zusätzliche Objekte und deren Wert in einer SNMP Nachricht zu übermitteln. Dies kann durch das optionale `OBJECTS` Attribut geschehen.

DESCRIPTION. Der DESCRIPTION Eintrag enthält weiterhin eine textuelle Beschreibung der SNMP Nachricht. Allerdings ist die DESCRIPTION Angabe in der SMIv2 nicht mehr optional sondern vorgeschrieben.

REFERENCE. Der REFERENCE Eintrag, der einen Verweis zu einer anderen Literaturquelle beinhaltet, die nähere Angaben zur SNMP Nachricht macht, ist wie bei der SMIv1 unverändert optional.

STATUS. Analog zur Definition von Objekten nach der SMIv2 ist – anders als bei der SMIv1 – auch für SNMP Nachrichten die Angabe eines STATUS Attributs verpflichtend. Mögliche Werte sind ebenfalls die drei Angaben current, deprecated und obsolete (siehe auch Tabelle 4.9).

Beispiel für die Definition einer SMIv2 konformen SNMP Nachricht

Abbildung 4.10 zeigt ein Beispiel für die Definition einer SNMP Nachricht aus der MIB-II unter Verwendung aller erlaubten Attribute. Das abgedruckte Beispiel zeigt einen gekürzten Ausschnitt aus der MIB für Repeater aus der aktuellen RFC 2108 [52]. Das Beispiel aus der hier gezeigten Abbildung 4.10 zeigt denselben aktualisierten Ausschnitt wie Abbildung 4.8, welche die ursprüngliche, nach SMIv1 in RFC 1368 [126] definierte MIB enthält.

Definition von Implementierungsanforderungen an SNMP Agenten

Durch eine MODULE-COMPLIANCE Angabe kann der Autor einer MIB Anforderungen an die Implementierung von SNMP Agenten stellen. In der MODULE-COMPLIANCE Angabe sind diejenigen Objekte, Nachrichten und Gruppen aufgelistet, die zur Konformität mit der entsprechenden MIB von einem SNMP Agenten zwingend unterstützt werden müssen. Dabei können auch Abhängigkeiten zu anderen MIBs aufgebaut werden. Für Standard MIBs ist die MODULE-COMPLIANCE Angabe verpflichtend.

STATUS. Die Angabe des Status der Anforderungen ist erforderlich und erfolgt über das bereits bekannte Attribut STATUS.

DESCRIPTION. Ebenfalls zwingend erforderlich ist die Angabe einer textuellen Beschreibung der Anforderungen an die SNMP Agenten. Die Realisierung erfolgt – wie bekannt – durch den Eintrag DESCRIPTION.

REFERENCE. Neben den notwendigen Angaben STATUS und DESCRIPTION kann auch optional ein Verweis auf eine andere Literaturquelle mit näheren Informationen zu den Anforderungen gemacht werden. Das entsprechende Attribut REFERENCE ist ebenfalls bereits bekannt.

MODULE. Die zentrale Angabe innerhalb eines MODULE-COMPLIANCE Eintrags ist zweifelsohne das MODULE Attribut, welches mindestens einmal vorhanden sein muss. Innerhalb des MODULE Attributs wiederum werden diejenigen

```
snmpDot3RptrMgt OBJECT IDENTIFIER ::= { mib-2 22 }

rptrInfoOperStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                  other(1),
                  ok(2),
                  failure(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The rptrInfoOperStatus object indicates the
            operational state of the repeater."
    REFERENCE
            "[IEEE 802.3 Mgt], 30.4.1.1.5, aRepeaterHealthState."
    ::= { rptrInfoEntry 3 }

rptrInfoHealth NOTIFICATION-TYPE
    OBJECTS     { rptrInfoOperStatus }
    STATUS      current
    DESCRIPTION
            "In a system containing multiple managed repeaters,
            the rptrInfoHealth notification conveys information
            related to the operational status of a repeater.
            It is sent either when the value of rptrInfoOperStatus
            changes, or upon completion of a non-disruptive test.

            The agent must throttle the generation of
            consecutive rptrInfoHealth notifications for
            the same repeater so that there is at least
            a five-second gap between notifications of this type.
            When notifications are throttled, they are dropped,
            not queued for sending at a future time.  (Note
            that 'generating' a notification means sending
            to all configured recipients.)"
    REFERENCE
            "[IEEE 802.3 Mgt], 30.4.1.3.1, nRepeaterHealth
            notification."
    ::= { snmpDot3RptrMgt 0 4 }
```

Abb. 4.10. Ausschnitt aus der MIB-II Definition für Repeater.

Gruppen für Objekte und SNMP Nachrichten des angegebenen Moduls über das Unterattribut `MANDATORY-GROUPS` definiert, welche von einem Entwickler zur Wahrung der Konformität zwingend umgesetzt werden müssen. Fehlt der Modulname, so beziehen sich die weiteren Angaben auf das aktuelle Modul. Zusätzlich zu den uneingeschränkt vorgeschriebenen Gruppen können noch einzelne bedingt abhängige Gruppen über das Unterattribut `GROUP` sowie einzelne Objekte über das Unterattribut `OBJECT` angegeben werden, zu denen mögliche Abweichungen zwischen Definition und minimaler Umsetzung definiert werden können.

MANDATORY-GROUPS. Eine Angabe von Gruppen, die der Entwickler eines SNMP Agenten für eine Konformität zur jeweiligen MIB unbedingt implementieren muss, erfolgt über das optionale Attribut `MANDATORY-GROUPS`. Es können sowohl Gruppen für Objekte als auch für SNMP Nachrichten definiert werden. Alle Gruppen, die nur bei bestimmten Abhängigkeiten implementiert sein müssen, dürfen nicht bei `MANDATORY-GROUPS` definiert werden, sondern sind unter dem separaten Schlüsselwort `GROUP` anzugeben.

GROUP. Manchmal lassen sich ein oder mehrere Gruppen von Objekten oder SNMP Nachrichten im Agenten nicht anwenden, weil Teile der in der Gruppe enthaltenen Elemente nicht in das System integriert sind, welches die MIB unterstützen möchte. In solchen Fällen kann der Autor der MIB dennoch dafür sorgen, dass dieses Gerät theoretisch volle Konformität zu der MIB erlangen kann. Zu diesem Zweck sind die entsprechenden Gruppen nicht bei den uneingeschränkt vorgeschriebenen Einträgen der `MANDATORY-GROUPS` aufzulisten, sondern bei den eingeschränkt vorgeschriebenen `GROUP` Attributen. Diese können bei Nichtanwendbarkeit vom Entwickler des SNMP Agenten ausgelassen werden, ohne die Konformität zur MIB zu gefährden. Für jeden dieser bedingt abhängigen `GROUP` Einträge muss außerdem eine Beschreibung mittels des üblichen `DESCRIPTION` Attributs vorhanden sein, in welcher die jeweiligen Abhängigkeiten aufgeführt sind.

OBJECT. Sowohl zu Objekten der uneingeschränkt vorgeschriebenen Gruppen als auch zu Objekten der bedingt abhängigen Gruppen lassen sich mittels des `OBJECT` Attributs Ausnahmen definieren. Im Regelfall beschränken sich die Ausnahmen auf eine geänderte Zugriffserlaubnis, was innerhalb des `OBJECT` Blocks mit `MIN-ACCESS` angegeben wird. Es sind aber auch andere Abweichungen möglich: Die Angaben `SYNTAX` und `WRITE-SYNTAX` definieren einen anderen Typ für das jeweilige Objekt für Leseoperationen oder Schreiboperationen. Tabelle 4.10 liefert eine detailliertere Beschreibung der drei optionalen Attribute. In jedem Falle aber ist die Angabe einer Beschreibung per `DESCRIPTION` Attribut für jedes Objekt vorgeschrieben.

Tabelle 4.10. Abweichungsmöglichkeiten bei der Implementierung von Objekten der uneingeschränkt vorgeschriebenen Gruppen und der bedingt abhängigen Gruppen der Structure of Management Information SMIv2.

Art der Abweichung	Beschreibung
SYNTAX	Der Typ des Objektes ist ein anderer. Es kann nur eine Untermenge des ursprünglichen Typs gewählt werden, also beispielsweise *Integer32 (1..10)* anstatt *Integer32*. Bei gleichzeitiger Angabe einer WRITE-SYNTAX gilt diese Abweichung nur für Lesezugriffe.
WRITE-SYNTAX	Der Typ des Objektes ist ausschließlich für Schreibzugriffe ein anderer. Es kann nur eine Untermenge des ursprünglichen Typs gewählt werden.
MIN-ACCESS	Für eine Konformität zur MIB reicht eine Zugriffsberechtigung aus, die höchstens dieser Angabe entspricht.

Beispiel für die Definition von Implementierungsanforderungen an SNMP Agenten

Abbildung 4.11 zeigt ein Beispiel für die Definition von Implementierungsanforderungen an SNMP Agenten. Das abgedruckte Beispiel zeigt einen gekürzten Ausschnitt aus der MIB für Medium Attachment Units (MAU) aus der RFC 2668 [199].

Definition von Implementierungscharakteristiken einzelner SNMP Agenten

Bei der Implementierung von MIBs kann es durchaus vorkommen, dass Teile der MIB nicht vollständig oder gar nicht implementiert wurden. Um diese Abweichungen von der ursprünglichen MIB definieren zu können, wurde in der SMIv2 der `AGENT-CAPABILITIES` Block geschaffen. Dort lassen sich alle unterstützten Objekte mit ihren eventuell vorhandenen Abweichungen auflisten. Außerdem ist es möglich, `AGENT-CAPABILITIES` für einzelne Teilbereiche einer MIB zu definieren. Umfangreiche und modular gestaltete MIBs lassen sich so bequem in einzelne Bibliotheken aufteilen, die jeweils eigene Implementierungscharakteristiken aufweisen können.

PRODUCT-RELEASE. Die verpflichtende Angabe `PRODUCT-RELEASE` enthält eine textuelle Beschreibung der Version des SNMP Agenten und seiner Implementierung. Für modular aufgebaute MIBs kann sich die Versionsangabe auch auf die entsprechende Teilbibliothek beziehen.

STATUS. Die Angabe des `STATUS` ist wie bei den meisten SMIv2 Elementen zwingend erforderlich und kann einen der drei bekannten Werte `current`, `deprecated` oder `obsolete` enthalten (siehe auch Tabelle 4.9).

```
mauModIfCompl2 MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION "Compliance for MAUs attached to interfaces."

    MODULE -- this module
        MANDATORY-GROUPS { mauIfGrpBasic }

        GROUP       mauIfGrpHighCapacity
        DESCRIPTION "Implementation of this optional group is
                     recommended for MAUs which have 100Mb/s
                     or greater capability."

        GROUP       mauIfGrpJack
        DESCRIPTION "Implementation of this optional group is
                     recommended for MAUs which have one or more
                     external jacks."

        GROUP       mauIfGrpAutoNeg2
        DESCRIPTION "Implementation of this group is mandatory
                     for MAUs which support managed
                     auto-negotiation."

        GROUP       mauIfGrpAutoNeg1000Mbps
        DESCRIPTION "Implementation of this group is mandatory
                     for MAUs which have 1000Mb/s or greater
                     capability and support managed
                     auto-negotiation."

        GROUP       ifMauNotifications
        DESCRIPTION "Implementation of this group is recommended
                     for MAUs attached to interfaces."

        OBJECT      ifMauStatus
        MIN-ACCESS  read-only
        DESCRIPTION "Write access is not required."
    ::= { mauModCompls 3 }
```

Abb. 4.11. Ausschnitt aus der MIB-II Definition für Medium Attachment Units.

DESCRIPTION. Die Beschreibung der Implementierungscharakteristiken ist in der SMIv2 vorgeschrieben und wird durch Angabe des bereits bekannten Attributs `DESCRIPTION` vorgenommen.

REFERENCE. Optional kann auch wieder das Attribut `REFERENCE` angegeben werden, das auf eine andere Literaturquelle hinweist, die nähere Informationen zu den Implementierungscharakteristiken liefert.

SUPPORTS. Innerhalb eines `AGENT-CAPABILITIES` Blocks können mehrere Angaben zu den jeweils vom SNMP Agenten unterstützten MIB Moduls gemacht werden. Jedes implementierte MIB Modul wird über einen separaten `SUPPORTS` Eintrag definiert. Zusätzlich muss für jeden `SUPPORTS` Eintrag auch ein `INCLUDES` Attribut folgen, über welches die jeweils unterstützten Objektgruppen und SNMP Nachrichtengruppen dieses Moduls angegeben werden.

INCLUDES. Da ein SNMP Agent ein MIB Modul auch nur teilweise implementieren kann, ist eine genaue Angabe der unterstützten Objekte notwendig. Die SMIv2 erlaubt diese Detaillierung über das Attribut `INCLUDES`, welches jedem `SUPPORTS` Eintrag folgen muss. Angegeben werden können jedoch nicht einzelne Objekte oder SNMP Nachrichten, sondern lediglich ganze MIB Gruppen. Sind auch innerhalb des MIB Moduls einzelne Objekte nicht in der vorgesehenen Weise implementiert, so kann eine weitere Detaillierung der Angaben über zusätzliche `VARIATION` Blöcke im Zusammenhang mit den `SUPPORTS` Einträgen gemacht werden.

VARIATION. Das Unterattribut `VARIATION` kann den Angaben `SUPPORTS` und `INCLUDES` folgen, wenn auf Ausnahmen bei der Implementierung der einzelnen Objekte einer MIB Gruppe hingewiesen werden soll. Jede der Variationen muss zumindest eine Beschreibung besitzen, die im Attribut `DESCRIPTION` untergebracht ist. Tabelle 4.11 zeigt mögliche Variationen, von denen eine oder mehrere pro `VARIATION` angegeben werden darf.

Beispiel für die Definition von Implementierungscharakteristiken einzelner SNMP Agenten

Ein Beispiel für die Definition von Implementierungscharakteristiken ist in Abbildung 4.12 dargestellt. Das abgedruckte Beispiel zeigt einen gekürzten Ausschnitt aus einer MIB des Geräteherstellers Cisco. Die den Namen `CISCO-AAA-SERVER-CAPABILITY` tragende MIB beschreibt Implementierungscharakteristiken in Bezug auf den Authentication, Authorization, and Accounting (AAA) Server.

Andere Neuerungen in der SMIv2

In der SMIv2 wurden zu den bisher aufgeführten Änderungen noch weitere Neuerungen eingeführt. Vor allem zur Verbesserung der Übersichtlichkeit und Verständlichkeit der MIBs wurden Gruppen eingeführt. Seit SMIv2 muss jedes

```
ciscoAAAServerCapabilityV10R00 AGENT-CAPABILITIES

    PRODUCT-RELEASE "Cisco IOS 12.0(4)XJ"
    STATUS          current
    DESCRIPTION     "Cisco AAA Server MIB capabilities"

    SUPPORTS        CISCO-AAA-SERVER-MIB
    INCLUDES        {casStatisticsGroup, casConfigGroup,
                     casServerNotificationGroup }

    VARIATION       casConfigEntry
                    CREATION-REQUIRES       casAddress

    VARIATION       casAddress
                    ACCESS read-only
                    DESCRIPTION
                            "create is not yet supported"

    VARIATION       casAuthenPort
                    ACCESS read-only
                    DESCRIPTION
                            "create is not yet supported"

    VARIATION       casAcctPort
                    ACCESS read-only
                    DESCRIPTION
                            "create is not yet supported"

    VARIATION       casKey
                    ACCESS read-only
                    DESCRIPTION
                            "create is not yet supported"

    VARIATION       casConfigRowStatus
                    ACCESS read-only
                    DESCRIPTION
                            "create is not yet supported"

    ::= { ciscoAAAServerCapability 1 }
```

Abb. 4.12. Ausschnitt aus der CISCO-AAA-SERVER-CAPABILITY MIB Definition für AAA Server. Zur besseren Lesbarkeit wurde die Einrückung den anderen gezeigten MIBs in diesem Buch abgepasst.

Tabelle 4.11. Variationsmöglichkeiten bei der Implementierung von Objekten, die in der Structure of Management Information SMIv2 definiert sind.

Art der Variation	Beschreibung
SYNTAX	Der Typ des Objektes ist ein anderer. Es kann nur eine Untermenge des ursprünglichen Typs gewählt werden, also beispielsweise *Integer32 (1..10)* anstatt *Integer32*. Bei gleichzeitiger Angabe einer WRITE-SYNTAX gilt diese Abweichung nur für Lesezugriffe.
WRITE-SYNTAX	Der Typ des Objektes ist ausschließlich für Schreibzugriffe ein anderer. Es kann nur eine Untermenge des ursprünglichen Typs gewählt werden.
ACCESS	Es wurde eine niedrigere Zugriffsberechtigung implementiert.
CREATION-REQUIRES	Diese Angabe ist nur für die OIDs einer Tabellezeile sinnvoll und möglich. Sie gibt diejenigen Objekte der Zeile an, die für eine Erstellung einer neuen Zeile zwingend durch die Managementstation initialisiert werden müssen.
DEFVAL	Es wurde ein abweichender Standardwert für Elemente einer Tabellenzeile implementiert.

SNMP Objekt verpflichtend in einer `OBJECT-GROUP` und jede SNMP Nachricht verpflichtend in einer `NOTIFICATION-GROUP` enthalten sein.

OBJECT-GROUP. Innerhalb einer `OBJECT-GROUP` Angabe werden unter dem Attribut `OBJECTS` alle zu dieser Gruppe gehörenden SNMP Objekte aufgelistet. Außerdem sind die bereits bekannten Attribute `STATUS` und `DESCRIPTION` verpflichtend sowie das Attribut `REFERENCE` optional anzugeben. Jedes SNMP Objekt muss in mindestens einer `OBJECT-GROUP` enthalten sein, es darf aber auch in mehreren Gruppen definiert werden.

NOTIFICATION-GROUP. Parallel zur `OBJECT-GROUP` muss auch jede SNMP Nachricht bei der SMIv2 in einer `NOTIFICATION-GROUP` enthalten sein. Die Angabe selbst erfolgt mittels des Schlüsselwortes `NOTIFICATIONS`. Die Attribute `STATUS`, `DESCRIPTION` und `REFERENCE` sind analog zur `OBJECT-GROUP` zu verwenden.

TEXTUAL-CONVENTION. Um einmal definierte Makros, die eine bessere Lesbarkeit in MIB Definitionen bringen, besser dokumentieren zu können, wurde in der SMIv2 die `TEXTUAL-CONVENTION` eingeführt. Mittels des verpflichtenden Attributs `SYNTAX` wird der neue Typ des Makros definiert. Weitere Angaben sind erneut die Einträge `STATUS`, `DESCRIPTION` und das optionale Attribut `REFERENCE`. Zusätzlich kann über das ebenfalls optionale Attribut `DISPLAY-HINT` ein Vorschlag für eine optimale Wiedergabe und Präsentation des Inhaltes gemacht werden. In der Tabelle 4.12 sind die Defi-

nitionsmöglichkeiten für das Attribut DISPLAY-HINT von Objekten des Typs INTEGER aufgelistet. Bei allen Formatierungen werden führende Nullen nicht berücksichtigt und deshalb bei der Ausgabe unterdrückt.

Tabelle 4.12. Definitionsmöglichkeiten für Angaben von DISPLAY-HINT bei Objekten des Typs INTEGER in der Structure of Management Information SMIv2.

Syntax	Beschreibung	Darstellung des Wertes 42
b	Der Inhalt wird als Binärzahl angezeigt	101010
d	Der Inhalt wird als dezimale Ganzzahl dargestellt	42
d-n	Der Inhalt wird als reelle Zahl angezeigt. Das Komma wird um die mit n angegebene ganze Zahl nach links verschoben, was einer Multiplikation mit 10^{-n} entspricht.	4.2 (bei $n = 1$)
o	Der Inhalt wird als Oktalzahl angezeigt	52
x	Der Inhalt wird als Hexadezimalzahl dargestellt	2A

Die Darstellung von Objekten des Typs OCTET STRING ist wesentlich komplizierter. Der Wert des Attributs DISPLAY-HINT besteht aus beliebig vielen Darstellungsformaten für Zeichenketten. Jedes Darstellungsformat besteht wiederum aus bis zu fünf Elementen:

1. Der optionale Multiplikator „*“ gibt an, dass das aktuelle Oktett des OCTET STRING Objektes als Multiplikator für die folgende Formatierung und nicht als Inhalt verwendet wird.
2. Die Anzahl der Zeichen vom aktuellen Wert des OCTET STRING Objektes, die angezeigt und für die weitere Bearbeitung aus der Zeichenkette entfernt werden muss. Die Anzahl ist eine positive Ganzzahl, die auch Null sein darf, aber nicht größer als die noch verbleibende Anzahl an Zeichen im OCTET STRING Objekt.
3. Die Formatvorgabe für die Darstellung der ausgewählten Zeichen.
   a Darstellung als ASCII Zeichen
   d Darstellung als Dezimalzahl
   o Darstellung als Oktalzahl
   t Darstellung als UTF-8 Zeichen [230]
   x Darstellung als Hexadezimalzahl
4. Ein optionales Trennzeichen, welches zwischen den aktuell ausgegebenen Zeichen und den Zeichen des nächsten Darstellungsformats dargestellt wird.
5. Ein optionales Wiederholungstrennzeichen, welches zwischen den durch Angabe des Multiplikators mehrfach dargestellten Zeichen ausgegeben wird. Das Wiederholungstrennzeichen kann nur angegeben werden, wenn

sowohl der Multiplikator „*“ als auch das normale Trennzeichen ebenfalls angegeben sind.

Tabelle 4.13 zeigt einige Beispiele für die formatierte Ausgabe eines SNMP Objektes vom Typ `OCTET STRING`. Die erste Spalte enthält den auszugebenden `OCTET STRING` in hexadezimaler Form. Die zweite Spalte gibt den Wert des `DISPLAY-HINT` Eintrags an, der aus einem oder mehreren Darstellungsformaten bestehen kann. Die Spalte „Formatelemente“ bezieht sich auf die vorhergehende Spalte und gibt an, welche der oben definierten fünf möglichen Elemente in den einzelnen Darstellungsformaten aus der zweiten Spalte angegeben wurden. Zur Veranschaulichung sind die Darstellungsformate jeweils einzeln in runde Klammern eingefasst. Die letzte Spalte gibt an, wie das `OCTET STRING` Objekt idealerweise auszugeben ist.

Tabelle 4.13. Beispiele für eine formatierte Ausgabe von Objekten des Typs OCTET STRING in der Structure of Management Information SMIv2.

OCTET STRING	Darstellungsformate	Formatelemente	Anzeige
„417274687572“H	„100a“	(23)	„Arthur“
„417274687572“H	„1d 5a“	(234)(23)	„65 rthur“
„03414243“H	„*1a“	(123)	„ABC“
„03414243“H	„*1a -“	(12345)	„A-B-C“
„7F000001“H	„1d.1d.1d.1d“	(234)(234)(234)(23)	„127.0.0.1“
„H“	„0aM0aI0aB0a“	(234)(234)(234)(23)	„MIB“
„417274687572“H	„100t“	(23)	„Arthur“

Beispiel für die neuen Angaben in der SMIv2

Abbildung 4.13 zeigt ein Beispiel für die Verwendung der weiteren Neuerungen der SMIv2. Das Beispiel zeigt einen gekürzten Ausschnitt aus der MIB für das Layer Two Tunneling Protocol (L2TP) aus der RFC 3371 [37].

4.3.3 SMIng

Im Jahr 1999 hat die Arbeit an einer neuen Structure of Management Information begonnen, die ein Jahr später an die Network Management Research Group (NMRG) der Internet Research Task Force (IRTF) übergeben wurde. Ziel des Vorhabens war es, die Unzulänglichkeiten der SMIv2, wie die Abhängigkeit von SNMP und ASN.1 zu beseitigen und in die neue Next Generation Structure of Management Information (SMIng) zu überführen. Anstatt sich auf SNMP zu konzentrieren, zielt SMIng auch auf beliebige andere Netzwerkmanagementprotokolle. Insbesondere zielt SMIng auch auf das Common Open

```
L2tpMilliSeconds ::=  TEXTUAL-CONVENTION
        DISPLAY-HINT    "d-3"
        STATUS          current
        DESCRIPTION
           "A period of time measured in units of .001 of seconds
            when used in conjunction with the DISPLAY-HINT will
            show seconds and fractions of second with a resolution
            of .001 of a second."
        SYNTAX          Integer32 (0..2147483646)

l2tpMappingGroup OBJECT-GROUP
        OBJECTS {
            l2tpTunnelMapIfIndex,
            l2tpSessionMapTunnelIfIndex,
            l2tpSessionMapLocalSID,
            l2tpSessionMapStatus
        }
        STATUS          current
        DESCRIPTION
           "A collection of objects providing index mapping."
        ::= { l2tpGroups 5 }

l2tpTrapGroup NOTIFICATION-GROUP
        NOTIFICATIONS {
            l2tpTunnelAuthFailure
        }
        STATUS          current
        DESCRIPTION
           "A collection of L2TP trap events as specified
            in NOTIFICATION-TYPE constructs."
        ::= { l2tpGroups 7 }
```

Abb. 4.13. Ausschnitt aus der MIB Definition für das Layer Two Tunneling Protocol.

Policy Service Provisioning (COPS-PR) Protokoll [41], welches in der Structure of Policy Provisioning Information (SPPI) [117] definiert wird. Beide Protokolle sind einander sehr ähnlich, jedoch lassen sie sich nur schwer miteinander verbinden. Die Basisdefinition von SMIng wurde in der RFC 3780 [206] festgehalten; zusätzlich enthält die RFC 3781 [205] Definitionen, wie sich SNMP auf SMIng abbilden lässt.

Eine der grundlegenden Neuerungen der SMIng ist der objektorientierte Ansatz. Die Definitionssprache arbeitet durchweg mit Klassen und Hierarchien, in denen sogar einfache Vererbung von Eigenschaften möglich ist. Auch im Detail erinnert SMIng eher an eine Programmiersprache als seine beiden Vor-

gänger. Anweisungsblöcke werden nun konsequenter von geschweiften Klammern „{“ und „}“ umschlossen und Kommentare werden nicht mehr mit zwei Bindestrichen „- -“ sondern mit zwei Schrägstrichen „//“ eingeleitet. Außerdem wird die Angabe einzelner Attribute nun überall durch ein Semikolon „;“ abgeschlossen. Die Abstammung von der SMIv2 ist dennoch in vielen Punkten wiederzuerkennen. Es sind viele Datentypen und Konstrukte teilweise sogar größtenteils unverändert übernommen worden. Abbildung 4.14 zeigt einige veranschaulichende Beispielfragmente aus der RFC 3781.

SMIng ist zum heutigen Stand noch im experimentellen Stadium und wird in der Praxis nahezu nicht verwendet. Die Arbeiten an der Weiterentwicklung von SMIng sind aber nicht beendet und es bleibt zu hoffen, dass SMIng nicht das Schicksal vom Next Generation Internet Protocol (IPng) teilen wird. Dieses ist auch über viele Jahre nie als endgültiger Standard verabschiedet worden und in der Praxis lange Zeit fast vollständig vernachlässigt worden.

4.4 Management Information Base

Wie bereits erwähnt wird die große Anzahl der OIDs nicht einzeln definiert, sondern sie werden in übergeordneten Objekten zusammengefasst. Die hierarchische Struktur, in welcher die einzelnen OIDs zusammengefasst sind, trägt den Namen Management Information Base (MIB). Mit der MIB-I und der MIB-II existieren zwei verschiedene grundlegende Standards für MIBs, wobei die MIB-II eine Erweiterung zum Vorgänger MIB-I darstellt und diesen de facto abgelöst hat. In den standardisierten MIBs sind vorrangig allgemeine Objekte definiert, die auf den verschiedensten Netzwerkkomponenten zu finden sind. Zusätzlich zu diesen Standard-MIBs ist eine größere Anzahl weiterer MIBs definiert worden, die sich jeweils mit spezielleren Teilbereichen eines Netzwerks beschäftigen. Tabelle A.2 in Anhang A listet alle MIBs auf, die zum Zeitpunkt der Erstellung dieses Buches von der Internet Engineering Task Force (IETF) in RFCs definiert worden sind.

Durch Standards lassen sich niemals alle Möglichkeiten abdecken, die sich in der Praxis ergeben. So kann also ein Hersteller von Netzwerkkomponenten ein Attribut über SNMP verfügbar machen wollen, das in keiner der Standard MIBs definiert ist. Für diesen Fall ist in der hierarchischen Struktur Platz für weitere private OIDs eingerichtet worden, unter denen jeder Hersteller unabhängig von den Standard-MIBs seine eigenen MIBs und damit seine eigenen Teilbäume einfügen kann. Aus der Beispiel-OID des vorigen Abschnitts lässt sich erkennen, dass die Standard-MIBs nicht etwa die Wurzel der hierarchischen Struktur repräsentieren, sondern sie sind wesentlich weiter unten – erst in der sechsten Hierarchie-Ebene – eingebunden. Abbildung 4.15 verdeutlicht die Position der Standard-MIBs (*mib/mib-2*) und die Stelle, an der herstellerspezifische MIBs eingebunden werden können (*private* oder genauer gesagt *enterprise*). Es gilt zu beachten, dass die MIB-II die MIB-I als Standard abgelöst hat und deshalb auch deren OID trägt. Demnach hat sowohl *mib* als auch

```
node iso                { oid 1;     status current; };
node   org              { oid iso.3; status current; };
node     dod            { oid org.6; status current; };
node       internet     { oid dod.1; status current; };

scalars ip {
  oid             mib-2.4;
  object ipForwarding { implements Ip.forwarding; };
  object ipDefaultTTL { implements Ip.defaultTTL; };
  // ...
  status          current;
  description
          "This scalar group implements the Ip class.";
};

table ifTable {
  oid             interfaces.2;
  index           (ifIndex);
  object ifIndex { implements Interface.index;       };
  object ifDescr { implements Interface.description; };
  // ...
  status          current;
  description
          "This table implements the Interface class.";
};

notification linkDown {
    oid         snmpTraps.3;
    signals     Interface.linkDown {
        object      ifIndex;
        object      ifAdminStatus;
        object      ifOperStatus;
    };
    status      current;
    description
          "This notification signals the linkDown event
           of the Interface class.";
};
```

Abb. 4.14. Ausschnitt aus der RFC 3781 zur Abbildung von SMIng auf SNMP.

mib-2 die OID *1.3.6.1.2.1*. Die MIB-I sowie die MIB-II müssen von jedem Hersteller eines SNMP-fähigen Gerätes implementiert werden. Alle anderen, von der IETF definierten MIBs sind optional und können bei Bedarf zusätzlich implementiert werden. Auch die Erweiterung der Agenten mit den Standard-MIBs um eigene MIBs und OIDs bleibt jedem Hersteller selbst überlassen.

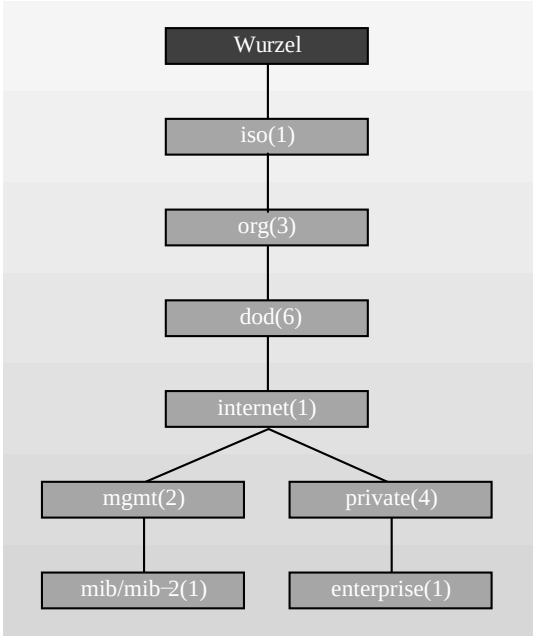

Abb. 4.15. Vereinfachte Struktur der OID-Hierarchie. Die Standard MIBs (MIB-I und MIB-II) befinden sich unter *mib/mib-2* (*.1.3.6.1.2.1*); herstellerspezifische MIBs finden sich unter *private* (*.1.3.6.1.4*).

4.4.1 MIB-I

Die MIB-I ist 1988 in RFC 1066 [122] aufgestellt worden und ist heute in einer fehlerkorrigierten Version in RFC 1156 [124] definiert. Sie befindet sich in der hierarchischen Struktur unterhalb des *mgmt* Objektes an der Position *1.3.6.1.2.1*. Die MIB-I besitzt genau acht Unterkategorien für grundlegende Objekte des Netzwerkmanagements. Da die MIB-I auf das Internet Protocol (IP) ausgerichtet wurde, ist es auch nicht verwunderlich, dass sich die meisten Kategorien auf die TCP/IP Protokollfamilie beziehen. Allem voran sind das

die MIB Teilbäume *ip* (*mib.4*), *icmp* (*mib.5*), *tcp* (*mib.6*) und *udp* (*mib.7*)[8]. Da die MIB-I durch die MIB-II ersetzt worden ist, welche die meisten Objekte aus ihrer Vorgängerversion übernommen hat, soll für eine nähere Beschreibung der MIB-I auf den nächsten Abschnitt 4.4.2 verwiesen werden. Das folgende Kapitel geht detailliert auf die einzelnen Objekte der MIB-II ein und erläutert auch, welche der Objekte bereits in der MIB-I vorhanden waren.

4.4.2 MIB-II

Die Erfahrungen mit der MIB-I haben gezeigt, dass einige ihrer OIDs ungeschickt formuliert waren, und dass andere wichtige Attribute nicht im Objektbaum der Standard-MIB enthalten waren. Aus diesem Grund wurde 1990 in RFC 1158 [181] die MIB-II als vollwertiger Ersatz für die MIB-I definiert. Kleinere Fehlerbeseitigungen sowie konkretere und verständlichere Formulierungen haben schließlich zur heute aktuellen Version in RFC 1213 [125] geführt. Es sind noch Erweiterungen zur MIB-II in der RFC 2011 [114], RFC 2012 [115] und RFC 2013 [116] formuliert, die als Erneuerung gelten. Sie ändern die Syntax der MIB-II jedoch in keiner Weise; es wird lediglich eine neuere Definitionssprache verwendet (SMIv2, siehe Seite 88).

Mit Ausnahme einiger kleiner Fehlerkorrekturen enthält die MIB-II sämtliche Objekte der MIB-I in unveränderter Form. Zusätzlich wurden eine Reihe weiterer OIDs definiert, welche den geänderten Anforderungen an eine Standard-MIB gerecht werden sollten. Hier ist insbesondere die Kategorie *snmp* (*mib-2.11*) zu erwähnen. Im Folgenden werden die OIDs der MIB-II im Einzelnen beschrieben[9].

system (*mib.1*)

Abbildung 4.16 veranschaulicht den *system* Zweig der MIB-I/MIB-II. In dieser Kategorie finden sich drei Objekte, die jeweils einen direkten Bezug zum gesamten System haben. Die MIB-II ergänzt weitere vier Objekte, so dass sich insgesamt folgende sieben Einträge ergeben:

***sysDescr* (*system.1*).** Enthält eine textuelle Beschreibung des Systems. Zu den angegebenen Informationen sollten der Name und eine eindeutige Identifikation der Hardware, des Betriebssystems und der laufenden Software gehören.

***sysObjectID* (*system.2*).** Beinhaltet einen Verweis auf eine andere OID, welche die genaue Herstellerbezeichnung des Systems enthält.

[8] *mib.4* steht für die vierte Unterkategorie der OID *mib*. Da *mib* gleichbedeutend mit *1.3.6.1.2.1* ist, steht *mib.4* für die OID *.1.3.6.1.2.1.4*, welches also die OID von *ip* ist. Analog dazu steht *.1.3.6.1.2.1.5* (*mib.5*) für *icmp*

[9] Da die MIB-I durch die MIB-II ersetzt worden ist und deren OID übernommen hat, gilt: *mib* = *mib-2*. Daraus folgt auch die Äquivalenz zwischen *system* = *mib.1* = *mib-2.1* = *.1.3.6.1.2.1.1*

***sysUpTime* (*system.3*).** Steht für die Zeit in Hundertstelsekunden seit der letzten Neuinitialisierung des Systems.

***sysContact* (*system.4*).** Liefert Informationen über einen Ansprechpartner zu diesem System sowie verschiedene Kontaktmöglichkeiten zu ihm.

***sysName* (*system.5*).** Enthält den administrativen Namen des Systems. Darunter ist der Full Qualified Domain Name (FQDN) des Systems zu verstehen.

***sysLocation* (*system.6*).** Beinhaltet genauere Angaben zum physikalischen Standort des Systems.

***sysServices* (*system.7*).** Steht für die Schichten des OSI Referenzmodells, in denen dieses System hauptsächlich arbeitet. Ein Zugangsrouter zum Internet arbeitet beispielsweise hauptsächlich in der dritten Schicht (Vermittlungsschicht).

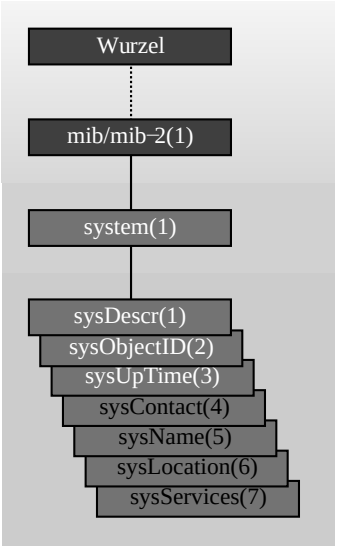

Abb. 4.16. *system*-Zweig des MIB-I/MIB-II Baumes. Dunkel beschriftete Objekte sind nur in der MIB-II definiert.

interfaces (*mib.2*)

Abbildung 4.17 veranschaulicht den *interfaces* Zweig der MIB-I/MIB-II. Diese Kategorie enthält genau zwei Objekte, welche die Anzahl und die Art der Schnittstellen des Systems beschreiben.

***ifNumber* (*interfaces.1*).** Liefert die Anzahl der Schnittstellen des Systems.

***ifTable* (*interfaces.2*).** Entspricht einem Unterbaum mit einer Tabelle, die weitere Beschreibungen zu jeder der Schnittstellen des Systems beinhaltet. Folglich findet sich exakt ein Objekt unterhalb der Tabelle mit dem Namen *ifEntry* (*ifTable.1*). Unterhalb dieses Elements finden sich die Einträge für die insgesamt 22 verschiedenen Angaben zu jeder der Schnittstellen. Neu in der MIB-II hinzugekommen ist der letzte Eintrag. Die jeweiligen untergeordneten OIDs lauten im Einzelnen:

- *ifIndex(1)* gibt den eindeutigen Index eines Eintrags an. Die Tabelle enthält exakt so viele Zeilen wie Schnittstellen in *interfaces.ifNumber* definiert sind. Die möglichen Werte für den Index liegen zwischen 1 und *interfaces.ifNumber*.
- *ifDescr(2)* enthält eine textuelle Beschreibung der Schnittstelle, in der auch Angaben zum Hersteller und der Produktbezeichnung enthalten sein können.
- *ifType(3)* gibt den genauen Typ der Schnittstelle an. In RFC 1213 waren noch genau 31 verschieden Typen spezifiziert. Durch die permanente Weiterentwicklung der Hardware sind in den vergangenen Jahren immer neue Schnittstellentypen entstanden, so dass in RFC 1573 [118] eine Aktualisierung der Liste auf insgesamt 53 verschiedene Typen vorgenommen wurde. Auch diese Liste ist nicht als endgültig zu betrachten, da die technischen Weiterentwicklungen unaufhaltsam sind.
- *ifMtu(4)* stellt eine Angabe zum größten Datagramm dar, das von dieser Schnittstelle gesendet und empfangen werden kann. Die Maximum Transfer Unit (MTU) wird in Oktetten angegeben.
- *ifSpeed(5)* liefert die Bandbreite der Schnittstelle in Bit pro Sekunde.
- *ifPhysAddress(6)* enthält die physikalische Adresse der Schnittstelle. Oftmals entspricht diese der Media Access Control (MAC) Adresse der jeweiligen Schnittstelle.
- *ifAdminStatus(7)* steht für den administrativen Status, den die Schnittstelle laut dem letzten Konfigurationsbefehl einnehmen soll.
- *ifOperStatus(8)* steht für den aktuellen operativer Status der Schnittstelle.
- *ifLastChange(9)* gibt den Zeitpunkt der letzten Statusänderung zum aktuellen Status aus *ifOperStatus(8)* an.
- *ifInOctets(10)* enthält einen Zähler über alle von dieser Schnittstelle empfangenen Oktette.
- *ifInUcastPkts(11)* liefert die Anzahl aller von dieser Schnittstelle empfangenen Unicast Pakete. Hierunter sind Pakete für definierte Einzelziele, nicht aber für Gruppenziele wie Broadcast-Pakete zu verstehen.
- *ifInNUcastPkts(12)* liefert in Ergänzung zu *ifInUcastPkts(11)* die Anzahl aller nicht Unicast Pakete, die von dieser Schnittstelle empfangen wurden.

- *ifInDiscards(13)* enthält einen Zähler aller empfangenen und verworfenen Pakete dieser Schnittstelle, die beispielsweise auf Grund eines Speicherüberlaufes gelöscht werden mussten.
- *ifInErrors(14)* entspricht einem Zähler, der alle fehlerhaften von dieser Schnittstelle empfangenen Pakete enthält.
- *ifInUnknownProtos(15)* zählt die von dieser Schnittstelle empfangenen Pakete, die als Ziel ein unbekanntes oder nicht unterstütztes Protokoll enthielten.
- *ifOutOctets(16)* enthält einen Zähler über alle von dieser Schnittstelle gesendeten Oktette.
- *ifOutUcastPkts(17)* liefert die Anzahl aller von dieser Schnittstelle gesendeten Unicast Pakete. Hierunter sind Pakete für definierte Einzelziele, nicht aber für Gruppenziele wie Broadcast-Pakete zu verstehen.
- *ifOutNUcastPkts(18)* liefert in Ergänzung zu *ifOutUcastPkts(17)* die Anzahl aller nicht Unicast Pakete, die von dieser Schnittstelle gesendet wurden.
- *ifOutDiscards(19)* enthält einen Zähler aller verworfenen und nicht gesendeten Pakete dieser Schnittstelle, die beispielsweise auf Grund eines Speicherüberlaufes gelöscht werden mussten.
- *ifOutErrors(20)* entspricht einem Zähler, der alle Pakete dieser Schnittstelle enthält, die auf Grund von Fehlern nicht gesendet werden konnten.
- *ifOutQLen(21)* enthält die Länge des ausgehenden Paketpuffers, was der Anzahl der auf Versendung wartenden Pakete entspricht.
- *ifSpecific(22)* verweist gegebenenfalls auf eine andere OID, in der nähere Angaben zum Medium definiert sind, das von dieser Schnittstelle verwendet wird.

Für jede Schnittstelle existiert nur genau eine Zeile in dieser Tabelle – also auch ein separates Objekt unterhalb der 22 Spalten.

at (mib.3)

Abbildung 4.18 veranschaulicht den *at* Zweig der MIB-I/MIB-II. Diese Kategorie enthält Angaben zur Beschreibung der vorhandenen physikalischen Schnittstellen. In der MIB-II wird diese Kategorie als veraltet definiert, da sie durch entsprechende Tabellen bei den Protokollen aus der Transportschicht des OSI Referenzmodells ersetzt wurde. Das einzige enthaltene Objekt dieser Kategorie besteht aus der Tabelle, die nach dem bekannten Schema aus Abbildung 4.3 definiert ist:

atTable (at.1). Die *atTable* ist *ifTable* aus der Kategorie *interfaces* sehr ähnlich und enthält nur das einzige Unterobjekt *atEntry* (*atTable.1*), unter dem sich die drei Angaben zu jeder der vorhandenen Schnittstellen befinden:

- *atIfIndex* (*atEntry.1*) ist die eindeutige Identifikationsnummer der Schnittstelle.

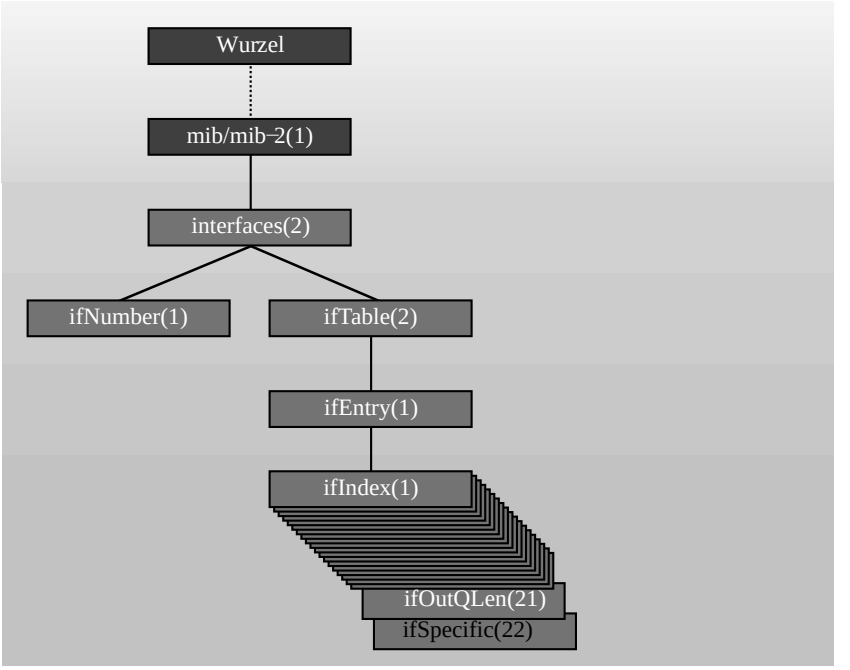

Abb. 4.17. *interfaces*-Zweig des MIB-I/MIB-II Baumes. Dunkel beschriftete Objekte sind nur in der MIB-II definiert.

- *atPhysAddress* (*atEntry.2*) steht für die physikalische Adresse der Schnittstelle.
- *atNetAddress* (*atEntry.3*) entspricht der logischen Adresse (IP Adresse) der Schnittstelle.

ip (*mib.4*)

Abbildung 4.19 veranschaulicht den *ip* Zweig der MIB-I/MIB-II. Diese Kategorie enthält 21 Unterkategorien mit verschiedensten Angaben zum System, die sich grundlegend auf das IP Protokoll und dessen Verwendung beziehen. Die MIB-II ergänzt die *ip* Kategorie um weitere zwei Elemente. Bei der Ausgliederung der *ip* und der *icmp* Gruppe in die separate RFC 2011 [114] wurde die Tabelle *ipRouteTable* als veraltet gekennzeichnet und durch die neue in RFC 1354 [7] sowie deren aktualisierte Fassung in RFC 2096 [9] definierte *ipForward* Tabelle ersetzt, so dass sich insgesamt 24 Einträge aus Skalaren und Tabellen ergeben.

***ipForwarding* (*ip.1*).** Gibt an, ob dieses System als ein Endgerät (Host) oder als ein Knotenpunkt (Gateway) im Netzwerk fungiert. Im ersten Fall kann dieses Gerät lediglich eigene Pakete senden und Pakete für sich selbst

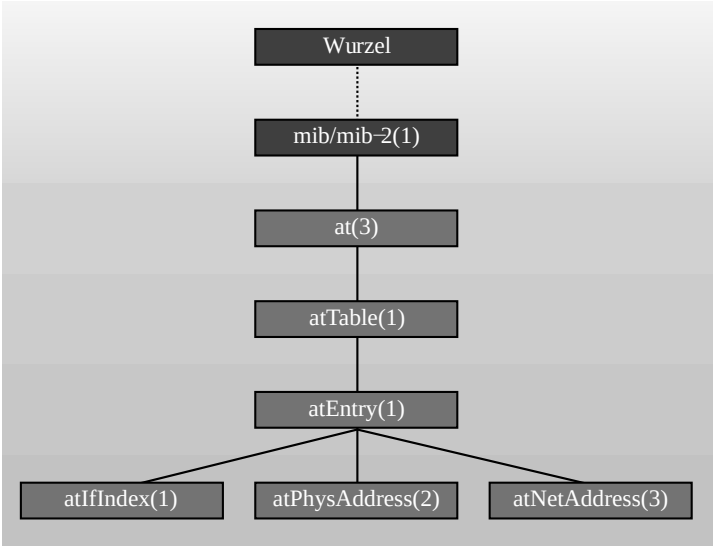

Abb. 4.18. *at*-Zweig des MIB-I/MIB-II Baumes.

entgegennehmen. Im zweiten Fall ist dieses Gerät zusätzlich in der Lage, Pakete zu anderen Komponenten weiterzuleiten und zu vermitteln.

***ipDefaultTTL* (*ip.2*).** Enthält den Standardwert für das Time-to-Live (TTL) Feld aller gesendeten IP Pakete. Der TTL Wert eines Paketes macht Angaben darüber, wie viele Knotenpunkte (Gateways) es auf dem Weg zum Bestimmungsort überqueren darf, bevor es als veraltet betrachtet wird und gelöscht werden muss.

***ipInRecieves* (*ip.3*).** Zähler, der die Summe aller empfangenen IP Pakete sämtlicher Schnittstellen enthält.

***ipInHdrErrors* (*ip.4*).** Liefert die Anzahl aller ungültigen eingegangenen IP Pakete mit fehlerhaften Angaben im Paketkopf.

***ipInAddrErrors* (*ip.5*).** Liefert die Summe aller eingegangenen IP Pakete, die auf Grund der Angabe eines lokal unbekannten oder ungültigen Ziels verworfen wurden.

***ipForwDatagrams* (*ip.6*).** Zähler, der die Summe aller von diesem System weitergeleiteten IP Pakete enthält. Für Systeme, die als Endpunkte im Netzwerk arbeiten, ist dieser Zähler immer gleich Null.

***ipInUnknownProtos* (*ip.7*).** Liefert die Summe aller eingegangenen IP Pakete, die auf Grund der Angabe eines unbekannten oder nicht unterstützten Protokolls verworfen werden mussten.

***ipInDiscards* (*ip.8*).** Entspricht der Summe aller gültigen eingegangenen IP Pakete, die auf Grund anderer Gründe verworfen werden mussten. Ein möglicher Grund dafür könnte beispielsweise eine zu hohe Systemauslastung sein.

***ipInDelivers* (*ip.9*).** Zähler, der die Summe aller IP Pakete enthält, die erfolgreich an höhere Protokolle wie TCP oder UDP weitergeleitet wurden.

***ipOutRequests* (*ip.10*).** Liefert die Summe aller von höheren Protokollen empfangenen Pakete, die anschließend vom System in IP Pakete umgewandelt und über das Netzwerk weitergeleitet wurden.

***ipOutDiscards* (*ip.11*).** Entspricht der Summe aller gültigen, von höheren Protokollen empfangenen Pakete, die auf Grund anderer Gründe verworfen werden mussten. Auch hier könnte eine zu hohe Systemauslastung eine mögliche Ursache sein.

***ipOutNoRoutes* (*ip.12*).** Zähler, der die Summe aller IP Pakete enthält, deren Weiterleitung auf Grund einer fehlenden Route zum vorgesehenen Ziel nicht durchgeführt werden konnte.

***ipReasmTimeout* (*ip.13*).** Enthält das Zeitintervall, in welchem das System auf weitere Paketfragmente wartet, um diese zu einem vollständigen IP Paket zusammenführen zu können.

***ipReasmReqds* (*ip.14*).** Liefert die Summe aller Fragmente, die noch nicht zu einem vollständigen IP Paket zusammengesetzt werden konnten.

***ipReasmOKs* (*ip.15*).** Entspricht der Summe aller von diesem System erfolgreich zusammengesetzten IP Pakete.

***ipReasmFails* (*ip.16*).** Liefert die Anzahl aller Fehler, die beim Zusammensetzen empfangener Fragmente zu IP Paketen aufgetreten sind.

***ipFragOKs* (*ip.17*).** Zähler, der die Summe aller IP Pakete enthält, die von diesem System zur weiteren Versendung erfolgreich in Fragmente aufgeteilt worden sind.

***ipFragFails* (*ip.18*).** Entspricht der Summe aller IP Pakete, welche das System zur weiteren Versendung nicht in Fragmente aufteilen konnte.

***ipFragCreates* (*ip.19*).** Liefert die Summe aller von diesem System erzeugten Fragmente.

***ipAddrTable* (*ip.20*).** Enthält eine Tabelle mit Einträgen zu den Eigenschaften der von diesem System verwendeten IP Adressen. Unterhalb der OID für die Tabelle findet sich nach dem bekannten Schema aus Abbildung 4.3 nur der Eintrag für die Spaltendefinitionen *ipAddrEnty* (*ipAddrTable.1*), unter dem sich insgesamt fünf Informationen befinden. Die fünfte und letzte Angabe wurde erst durch die MIB-II hinzugefügt.

- *ipAdEntAddr(1)* IP Adresse, auf die sich die jeweilige Tabellenzeile bezieht.
- *ipAdEntIfIndex(2)* Index der Schnittstelle, welche diese IP Adresse verwendet.
- *ipAdEntNetMask(3)* Netzwerkmaske zu dieser IP Adresse. Durch die Angabe einer Netzwerkmaske ist gleichzeitig das Subnetz definiert, in dem diese IP Adresse verwendet wird.
- *ipAdEntBcastAddr(4)* Angabe zur zugehörigen Broadcast-Adresse dieser IP Adresse, die entweder Null oder Eins sein kann. Der Wert entspricht dem niederwertigsten Bit der vollständigen Broadcast-Adresse.
- *ipAdEntReasmMaxSize(5)* Maximale Größe, die ein aus mehreren Fragmenten zusammengesetztes IP Paket erreichen darf.

***ipRouteTable* (*ip.21*).** Beinhaltet eine Tabelle mit Einträgen zur Routingtabelle des Systems. In der MIB-I lautete der Name dieses Objektes noch *ipRoutingTable*. Einziges Objekt unterhalb der Tabelle ist wieder nur die Spaltendefinition *ipRouteEntry* (*ipRouteTable.1*). Im Normalfall bauen nur Knotenpunkte im Netzwerk, die auch IP Pakete vermitteln, eine Routingtabelle auf. Jeder Eintrag der Tabelle besteht aus zehn Attributen, die in der MIB-II um weitere drei auf insgesamt 13 Attribute erweitert wurden. Aktuell gilt die *ipRouteTable* als veraltet und sollte nicht mehr genutzt werden. Größter Nachteil war die fehlende Unterstützung für Classless Inter-Domain Routing (CIDR), die durch RFC 2096 ergänzt wurde.

- *ipRouteDest(1)* Zieladresse der Route. Es kann zwar mehr als eine Route zum selben Ziel führen, jedoch ist die Repräsentation der Routen in der *ipRouteTable* dafür nicht ausgelegt. Grund dafür ist die Verwendung der Zieladresse als eindeutiger Index der Tabelle. Ziel einer Route kann sowohl eine einzelne IP Adresse als auch ein Subnetz sein. Die Angabe der IP Adresse 0.0.0.0 als Ziel entspricht der Definition einer Standard-Route (Default Route), die für alle Pakete gültig ist, für die keine spezifischere Route in der Routingtabelle vorhanden ist.
- *ipRouteIfIndex(2)* Index derjenigen Schnittstelle, über welche die angegebene Zieladresse der Route erreichbar ist.
- *ipRouteMetric1(3)* Metrik erster Ordnung für diesen speziellen Routingeintrag. Die genaue Bedeutung der Metrik einer Route ist durch das jeweils verwendete Routingprotokoll definiert. Beispiele für eine Metrik sind die Länge einer Route, die Kosten einer Route oder die Anzahl von Knotenpunkten auf dem Weg einer Route.
- *ipRouteMetric2(4)* Metrik zweiter Ordnung. Die genaue Bedeutung ist wieder durch das verwendete Routingprotokoll bestimmt. Nicht verwendete Metriken werden durch den Wert -1 symbolisiert.
- *ipRouteMetric3(5)* Metrik dritter Ordnung.
- *ipRouteMetric4(6)* Metrik vierter Ordnung.
- *ipRouteNextHop(7)* IP Adresse des nächsten Knotenpunktes dieses Routingeintrags.

- *ipRouteType(8)* Die wichtigsten Werte für den Typ der Route sind *direct* für direkt über eine Schnittstelle erreichbare Zieladressen, *indirect* für Zieladressen, die sich nur über mindestens einen weiteren Knotenpunkt erreichen lassen, sowie *invalid* für ungültige Routen. Für zukünftige Erweiterungsmöglichkeiten ist noch der Typ *other* definiert.
- *ipRouteProto(9)* Verwendetes Routingprotokoll. Zur Auswahl stehen bereits 13 vordefinierte Routingprotokolle.
- *ipRouteAge(10)* Alter des Routingeintrags. Gemeint ist die Anzahl der Sekunden, die seit der letzten Bestätigung der Gültigkeit des Routingeintrags vergangen sind.
- *ipRouteMask(11)* Netzmaske des Routingeintrags. Durch die Netzmaske wird die Größe des Subnetzes der Zieladresse definiert.
- *ipRouteMetric5(12)* Metrik fünfter Ordnung.
- *ipRouteInfo(13)* Verweis auf eine OID, die nähere Angaben zum verwendeten Routingprotokoll macht.

***ipNetToMediaTable* (*ip.22*).** Enthält eine Tabelle mit Einträgen zu Adressauflösungen. Aus dieser Tabelle lassen sich logische (IP) Adressen in physikalische (MAC) Adressen umsetzen. Die vier Untereinträge unterhalb des Objekts *ipNetToMediaEntry* (*ipNetToMediaTable.1*) für die Spaltendefinitionen lauten deshalb:

- *ipNetToMediaIfIndex(1)* Index der Schnittstelle, welche diese IP Adresse verwendet.
- *ipNetToMediaPhysAddress(2)* Zugehörige physikalische Adresse zur IP Adresse dieses Eintrags.
- *ipNetToMediaNetAddress(3)* IP Adresse dieses Tabelleneintrags.
- *ipNetToMediaType(4)* Art der Zuweisung der IP Adresse zur Schnittstelle. *static* entspricht der statischen Konfiguration der IP Adresse und *dynamic* entspricht der Zuweisung über ein dynamisches Protokoll wie das Dynamic Host Configuration Protocol (DHCP). Auch bei *ipNetToMediaType* existieren die beiden zusätzlichen Werte *invalid* für ungültige Zuweisungen und *other* für zukünftige Erweiterungsmöglichkeiten.

***ipRoutingDiscards* (*ip.23*).** Liefert die Anzahl der gültigen Einträge aus der Routingtabelle, die aus besonderen Gründen gelöscht werden mussten. Ein möglicher Grund könnte der Überlauf des Speicherplatzes für die Routingtabelle sein.

***ipForward* (*ip.24*).** Stellt ein in RFC 1354 und RFC 2096 definiertes Modul dar, in dem sich verschiedene Attribute zum CIDR befinden:

- *ipForwardNumber(1)* Enthält die Anzahl der gültigen Einträge in der *ipForwardTable*.

- *ipForwardTable(2)* Stellt eine Tabelle mit insgesamt 15 Attributen zu den einzelnen Routingeinträgen dar: *ipForwardDest*, *ipForwardMask*, *ipForwardPolicy*, *ipForwardNextHop*, *ipForwardIfIndex*, *ipForwardType*, *ipForwardProto*, *ipForwardAge*, *ipForwardInfo*, *ipForwardNextHopAS*, *ipForwardMetric1*, *ipForwardMetric2*, *ipForwardMetric3*, *ipForwardMetric4*, und *ipForwardMetric5*.
- *ipCidrRouteNumber(3)* Enthält analog zu *ipForwardNumber* die Anzahl der gültigen Einträge in der *ipCidrRouteTable*, die auch CIDR unterstützt.
- *ipCidrRouteTable(4)* Stellt analog zu *ipForwardTable* eine Tabelle mit 16 Attributen zu den einzelnen CIDR Routing-Einträgen dar: *ipCidrRouteDest*, *ipCidrRouteMask*, *ipCidrRouteTos*, *ipCidrRouteNextHop*, *ipCidrRouteIfIndex*, *ipCidrRouteType*, *ipCidrRouteProto*, *ipCidrRouteAge*, *ipCidrRouteInfo*, *ipCidrRouteNextHopAS*, *ipCidrRouteMetric1*, *ipCidrRouteMetric2*, *ipCidrRouteMetric2*, *ipCidrRouteMetric4*, *ipCidrRouteMetric5* und *ipCidrRouteStatus*.

icmp (mib.5)

Abbildung 4.20 veranschaulicht den *icmp* Zweig der MIB-I/MIB-II, der mit der Kategorie *ip* vergleichbar ist. Er enthält insgesamt 26 verschiedene Angaben zum System bezüglich des Protokolls ICMP.

***icmpInMsgs* (*icmp.1*).** Zähler, welcher die Summe aller empfangenen ICMP Nachrichten enthält.

***icmpInErrors* (*icmp.2*).** Entspricht der Summe aller ungültigen, von diesem System empfangenen ICMP Nachrichten.

***icmpInDestUnreachs* (*icmp.3*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Destination Unreachable".

***icmpInTimeExcds* (*icmp.4*).** Enthält die Summe aller empfangenen ICMP Nachrichten des Typs „Time Exceeded".

***icmpInParmProbs* (*icmp.5*).** Enthält die Summe aller empfangenen ICMP Nachrichten des Typs „Parameter Problem".

***icmpInSrcQuenchs* (*icmp.6*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Source Quench".

***icmpInRedirects* (*icmp.7*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Redirect".

***icmpInEchos* (*icmp.8*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Echo Request".

***icmpInEchoReps* (*icmp.9*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Echo Reply".

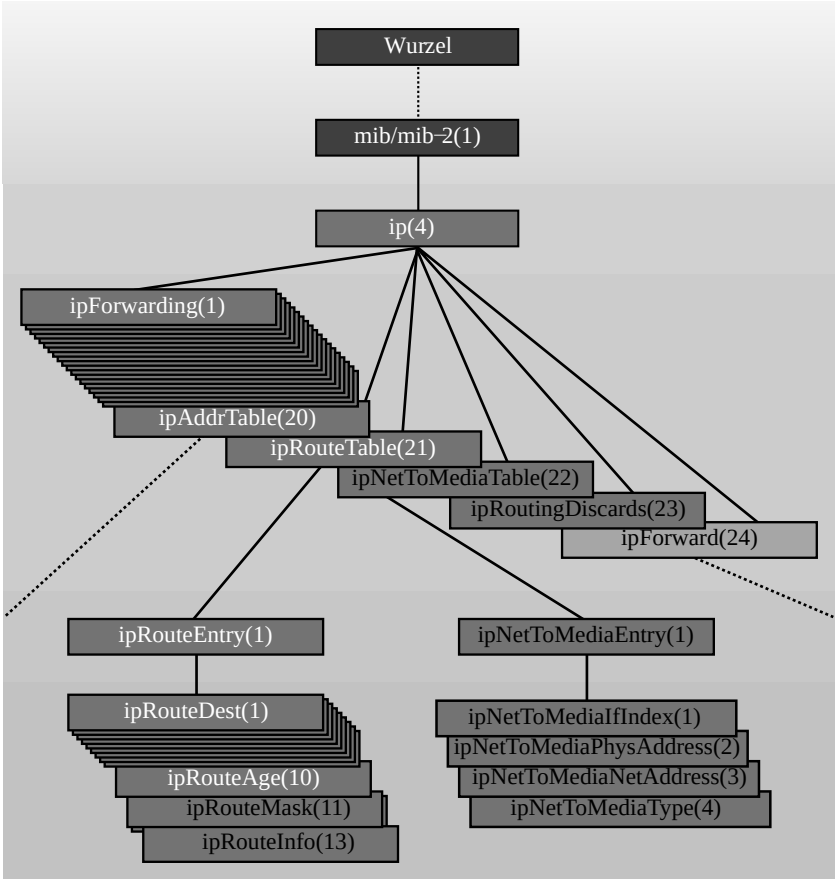

Abb. 4.19. *ip*-Zweig des MIB-I/MIB-II Baumes. Dunkel beschriftete Objekte sind nur in der MIB-II definiert. Das hellgrau hinterlegte Objekt (*ipForward*) symbolisiert einen ganzen Unterzweig, der in RFC 1354 zur MIB-II hinzugefügt wurde.

***icmpInTimestamps* (*icmp.10*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Timestamp Request".

***icmpInTimestampReps* (*icmp.11*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Timestamp Reply".

***icmpInAddrMasks* (*icmp.12*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Address Mask Request".

***icmpInAddrMaskReps* (*icmp.13*).** Beinhaltet die Summe aller empfangenen ICMP Nachrichten des Typs „Address Mask Reply".

***icmpOutMsgs* (*icmp.14*).** Zähler, der die Summe aller gesendeten ICMP Nachrichten enthält.

***icmpOutErrors* (*icmp.15*).** Liefert die Anzahl aller auf Grund von Fehlern nicht gesendeten ICMP Nachrichten.

***icmpOutDestUnreachs* (*icmp.16*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Destination Unreachable".

***icmpOutTimeExcds* (*icmp.17*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Time Exceeded".

***icmpOutParmProbs* (*icmp.18*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Parameter Problems".

***icmpOutSrcQuench* (*icmp.19*).** Enthält die Summe aller gesendeten ICMP Nachrichten des Typs „Source Quench".

***icmpOutRedirects* (*icmp.20*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Redirect".

***icmpOutEchos* (*icmp.21*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Echo Request".

***icmpOutEchoReps* (*icmp.22*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Echo Reply".

***icmpOutTimestamps* (*icmp.23*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Timestamp Request".

***icmpOutTimestampReps* (*icmp.24*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Timestamp Reply".

***icmpOutAddrMasks* (*icmp.25*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Address Mask Request".

***icmpOutAddrMaskReps* (*icmp.26*).** Beinhaltet die Summe aller gesendeten ICMP Nachrichten des Typs „Address Mask Reply".

tcp (*mib.6*)

Abbildung 4.21 veranschaulicht den *tcp* Zweig der MIB-I/MIB-II, der ebenfalls mit der Kategorie *ip* vergleichbar ist. Er enthält 13 verschiedene Angaben zum System bezüglich des verbindungsorientierten Protokolls TCP. Die MIB-II ergänzt weitere zwei Objekte, so dass sich insgesamt 15 Einträge ergeben. Später wurde der TCP Zweig der MIB-II ausgekoppelt und separat in RFC 2012 [115] unter Verwendung der Structure of Management Information SMIv2 veröffentlicht. Außerdem wurde in RFC 2452 [49] eine Ergänzung für das Protokoll IPv6 aufgestellt. Eine Aktualisierung der TCP-MIB wurde kürzlich in der RFC 4022 [170] herausgegeben.

***tcpRtoAlgorithm* (*tcp.1*).** Gibt den Algorithmus an, der zur Bestimmung des Timeout-Wertes für eine erneute Versendung von TCP Paketen verwendet wird, nachdem Probleme bei der Übertragung aufgetreten sind. Es sind

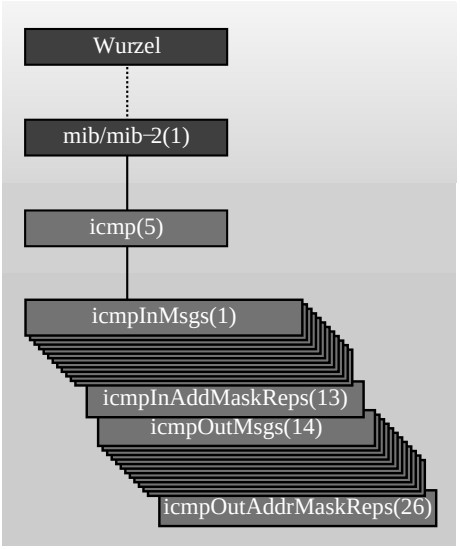

Abb. 4.20. *icmp*-Zweig des MIB-I/MIB-II Baumes.

die drei Algorithmen *constant*, *rsre* und *vanj* sowie die Erweiterungsmöglichkeit *other* für die Zukunft bereits vorgegeben. In der aktuellsten Version ist zusätzlich noch der neue Algorithmus *rfc2988* hinzugefügt worden.

***tcpRtoMin* (*tcp.2*).** Enthält den minimalen Timeout-Wert für eine erneute Versendung von TCP Paketen nach dem Auftreten von Problemen bei der Übertragung.

***tcpRtoMax* (*tcp.3*).** Enthält den maximalen Timeout-Wert für eine erneute Versendung von TCP Paketen nach dem Auftreten von Problemen bei der Übertragung.

***tcpMaxConn* (*tcp.4*).** Liefert die maximale Anzahl der vom System unterstützten parallelen TCP Verbindungen.

***tcpActiveOpens* (*tcp.5*).** Zähler, welcher die Summe aller vom System korrekt aufgebauten TCP Verbindungen enthält.

***tcpPassiveOpens* (*tcp.6*).** Liefert die Summe aller vom System empfangenen Versuche eines TCP Verbindungsaufbaus.

***tcpAttemptFails* (*tcp.7*).** Entspricht der Summe aller fehlgeschlagenen Versuche eines TCP Verbindungsaufbaus.

***tcpEstabReset* (*tcp.8*).** Enthält die Summe aller ordnungsgemäß durch ein Reset (RST) abgebauten TCP Verbindungen.

***tcpCurrEstab* (*tcp.9*).** Liefert die Anzahl der aktuell offenen TCP Verbindungen des Systems.

***tcpInSegs* (*tcp.10*).** Zähler, welcher die Summe aller empfangenen TCP Pakete enthält.

***tcpOutSegs* (*tcp.11*).** Liefert die Summe aller vom System gesendeten TCP Pakete.

***tcpRetransSegs* (*tcp.12*).** Enthält die Summe aller wiederholt gesendeten TCP Pakete. Dies betrifft im Wesentlichen Pakete, die wegen eines Übertragungsproblems erneut gesendet werden mussten.

***tcpConnTable* (*tcp.13*).** Beinhaltet eine Tabelle mit allen aktuellen TCP Verbindungen, die nicht zwangsweise bereits vollständig aufgebaut oder wieder abgebaut sind. Seit RFC 4022 gilt diese Tabelle als veraltet und wurde durch die neue Tabelle *tcpConnectionTable* ersetzt. Analog dem Schema aus Abbildung 4.3 befindet sich nur das Objekt *tcpConnEntry* (*tcpConnTable.1*) zur Spaltendefinition unterhalb der Tabelle. Wiederum eine Hierarchiestufe niedriger sind folgende fünf Angaben zu offenen TCP Verbindung gespeichert:

- *tcpConnState(1)* Status dieser TCP Verbindung. Es existieren insgesamt zwölf verschiedene Statuszustände, in denen sich eine Verbindung befinden kann: *closed(1)*, *listen(2)*, *sysSent(3)*, *sysReceived(4)*, *established(5)*, *finWait1(6)*, *finWait2(7)*, *closeWait(8)*, *lastAck(9)*, *closing(10)*, *timeWait(11)* und *deleteTCB(12)*.
- *tcpLocalAddress(2)* IP Adresse dieser TCP Verbindung auf lokaler Seite.
- *tcpRemAddress(3)* IP Adresse dieser TCP Verbindung auf der entfernten Seite.
- *tcpLocalPort(4)* Port-Nummer dieser TCP Verbindung auf lokaler Seite.
- *tcpRemPort(5)* Port-Nummer dieser TCP Verbindung auf der entfernten Seite.

***tcpInErrs* (*tcp.14*).** Zähler, der die Summe aller fehlerhaften empfangenen TCP Pakete mit ungültiger Prüfsumme enthält.

***tcpOutRsts* (*tcp.15*).** Liefert die Summe aller empfangenen TCP Pakete, die ein gesetztes RST Flag enthielten. Dieses Flag wird zum ordnungsgemäßen Trennen einer bestehenden Verbindung genutzt.

***ipv6TcpConnTable* (*tcp.16*).** Diese Tabelle, die Informationen zu allen aktuellen TCP Verbindungen enthält, welche auf dem Protokoll IPv6 aufbauen, wurde in der RFC 2452 [49] hinzugefügt. Die Unterscheidung zwischen verschiedenen IP Versionen ist aber bereits wieder veraltet, so dass die Tabelle nicht mehr verwendet wird. Zur Vollständigkeit wird sie an dieser Stelle dennoch aufgelistet. Analog dem Schema aus Abbildung 4.3 befindet sich nur das Objekt *ipv6TcpConnEntry* (*ipv6TcpConnTable.1*) zur Spaltendefinition unterhalb dieser Tabelle. Wiederum eine Hierarchiestufe niedriger sind folgende

sechs Angaben zu offenen TCP Verbindung gespeichert, die größtenteils schon in der analogen Tabelle *tcpConnTable* spezifiziert wurden:

- *ipv6TcpConnLocalAddress(1)* IPv6 Adresse dieser TCP Verbindung auf lokaler Seite.
- *ipv6TcpConnLocalPort(2)* Port-Nummer dieser TCP Verbindung auf lokaler Seite.
- *ipv6TcpConnRemAddress(3)* IPv6 Adresse dieser TCP Verbindung auf der entfernten Seite.
- *ipv6TcpConnRemPort(4)* Port-Nummer dieser TCP Verbindung auf der entfernten Seite.
- *ipv6TcpConnIfIndex(5)* Index der Schnittstelle, die mit dieser IPv6 TCP Verbindung auf lokaler Seite verbunden ist.
- *ipv6TcpConnState(6)* Status dieser IPv6 TCP Verbindung (vergleichbar zum *tcpConnState* der *tcpConnTable*). Es sind dieselben zwölf Statuszustände für Verbindungen möglich, wie sie bereits im *tcpConnState* Eintrag definiert worden sind: *closed(1)*, *listen(2)*, *sysSent(3)*, *sysReceived(4)*, *established(5)*, *finWait1(6)*, *finWait2(7)*, *closeWait(8)*, *lastAck(9)*, *closing(10)*, *timeWait(11)* und *deleteTCB(12)*.

***tcpHCInSegs* (*tcp.17*).** Entspricht dem Zähler aller empfangenen TCP Pakete *tcpInSegs*, wird jedoch durch einen 64 Bit langen Zähler repräsentiert, welcher die heutigen großen Paketmengen beherrschbar macht. Dieser Zähler ist parallel zum 32 Bit Zähler gültig.

***tcpHCOutSegs* (*tcp.18*).** Entspricht dem Zähler aller gesendeten TCP Pakete *tcpOutSegs*, wird jedoch durch einen 64 Bit langen Zähler repräsentiert, welcher die heutigen großen Paketmengen beherrschbar macht. Dieser Zähler ist parallel zum 32 Bit Zähler gültig.

***tcpConnectionTable* (*tcp.19*).** Der größte Unterschied zwischen der mittlerweile veralteten *tcpConnTable* und der *tcpConnectionTable* liegt in der zusätzlichen Angabe von Adresstypen für die beiden Kommunikationspartner. Auch diese Tabelle enthält analog zum Schema aus Abbildung 4.3 nur das einzelne Unterobjekt *tcpConnectionEntry* (*tcpConnectionTable.1*) zur Definition der Spalten, unterhalb dessen sich die folgenden acht Spaltenelemente befinden:

- *tcpConnectionLocalAddressType(1)* Adresstyp für die angegebene lokale Adresse dieser TCP Verbindung. Bislang definiert sind die drei Typen *ipv4* (IP Adresse), *ipv6* (IPv6 Adresse) und *dns* (Domainname).
- *tcpConnectionLocalAddress(2)* Adresse dieser TCP Verbindung auf lokaler Seite.
- *tcpConnectionLocalPort(3)* Port-Nummer dieser TCP Verbindung auf lokaler Seite.
- *tcpConnectionRemAddressType(4)* Adresstyp für die angegebene Adresse auf der entfernten Seite dieser TCP Verbindung. Bislang definiert sind die

drei Typen *ipv4* (IP Adresse), *ipv6* (IPv6 Adresse) und *dns* (Domainname).
- *tcpConnectionRemAddress(5)* Adresse dieser TCP Verbindung auf der entfernten Seite.
- *tcpConnectionRemPort(6)* Port-Nummer dieser TCP Verbindung auf der entfernten Seite.
- *tcpConnectionState(7)* Status dieser TCP Verbindung. Es sind dieselben zwölf Statuszustände für Verbindung wie im *tcpConnState* Eintrag möglich: *closed(1)*, *listen(2)*, *sysSent(3)*, *sysReceived(4)*, *established(5)*, *finWait1(6)*, *finWait2(7)*, *closeWait(8)*, *lastAck(9)*, *closing(10)*, *timeWait(11)* und *deleteTCB(12)*.
- *tcpConnectionProcess(8)* Zusätzliche Angabe zu dieser TCP Verbindung, mit welcher der Prozess auf dem lokalen System näher beschrieben werden kann.

***tcpListenerTable* (*tcp.20*).** Tabelle mit Einträgen zu lokalen Prozessen, die neue IP oder IPv6 TCP Verbindungen entgegennehmen können. Der einzige Eintrag ist – wie aus dem Schema in Abbildung 4.3 bereits bekannt – das Objekt zur Spaltendefinition *tcpListenerEntry* (*tcpListenerTable.1*), unter dem sich die folgenden vier Spaltenobjekte befinden:

- *tcpListenerLocalAddressType(1)* Typ der angegebenen lokalen Adresse dieses auf eingehende TCP Verbindungsaufbauversuche wartenden Prozesses. Bislang definiert sind die drei Typen *ipv4* (IP Adresse), *ipv6* (IPv6 Adresse) und *dns* (Domainname).
- *tcpListenerLocalAddress(2)* Lokale Adresse dieses auf eingehende TCP Verbindungsaufbauversuche wartenden Prozesses.
- *tcpListenerLocalPort(3)* Port-Nummer dieses auf eingehende TCP Verbindungsaufbauversuche wartenden Prozesses.
- *tcpListenerProcess(4)* Nähere Beschreibung dieses auf eingehende TCP Verbindungsaufbauversuche wartenden Prozesses. Dies kann beispielsweise die Prozess Identifikationsnummer (PID) sein.

udp (*mib.7*)

Abbildung 4.22 veranschaulicht den *udp* Zweig der MIB-I/MIB-II, der wie *icmp* und *tcp* mit der Kategorie *ip* vergleichbar ist. Er enthält vier verschiedene Angaben zum System bezüglich des Protokolls UDP. Die MIB-II ergänzt diese vier Attribute um eine Tabelle, so dass sich insgesamt folgende fünf Einträge ergeben, die auch in der zur Kapselung des *udp* Zweigs verwendeten RFC 2013 [116] aufgelistet sind:

***udpInDatagrams* (*udp.1*).** Enthält die Summe aller vom System empfangenen und erfolgreich an den zugehörigen Systemdienst weitergeleiteten UDP Pakete.

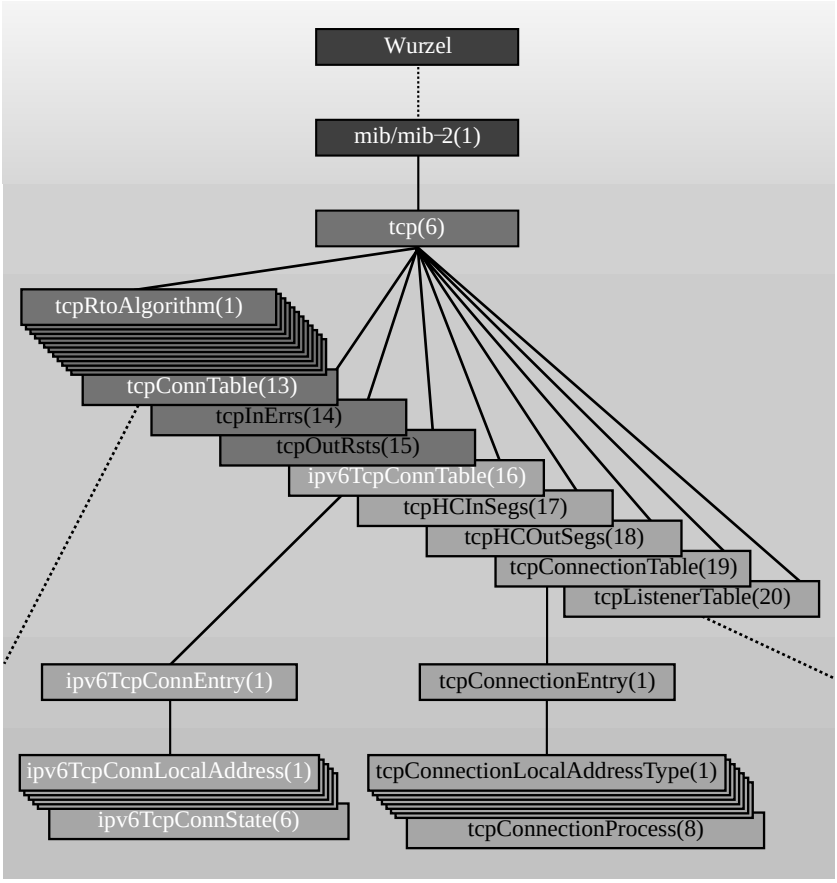

Abb. 4.21. *tcp*-Zweig des MIB-I/MIB-II Baumes. Dunkel beschriftete Objekte sind nur in der MIB-II definiert.

***udpNoPorts* (*udp.2*).** Liefert die Summe aller empfangenen UDP Pakete, für die es auf dem lokalen System keinen passenden Empfängerdienst gab.

***udpInErrors* (*udp.3*).** Entspricht der Summe aller UDP Pakete, die auf Grund von fehlerhaften Angaben nicht ausgeliefert werden konnten.

***udpOutDatagrams* (*udp.4*).** Zähler, welcher die Summe aller vom System gesendeten UDP Pakete beinhaltet.

***udpTable* (*udp.5*).** Enthält eine Tabelle mit Angaben über aktuelle Prozesse des Systems, die UDP Nachrichten entgegennehmen können. Diese Tabelle enthält gemäß dem Schema aus Abbildung 4.3 nur das Spaltendefinitionsobjekt *udpEntry* (*updTable.1*) mit zwei dort untergeordneten OIDs:

- *udpLocalAddress(1)* IP Adresse des lokalen Prozesses, der auf eingehende UDP Pakete wartet.
- *udpLocalPort(2)* Port-Nummer des lokalen Prozesses, der auf eingehende UDP Pakete wartet.

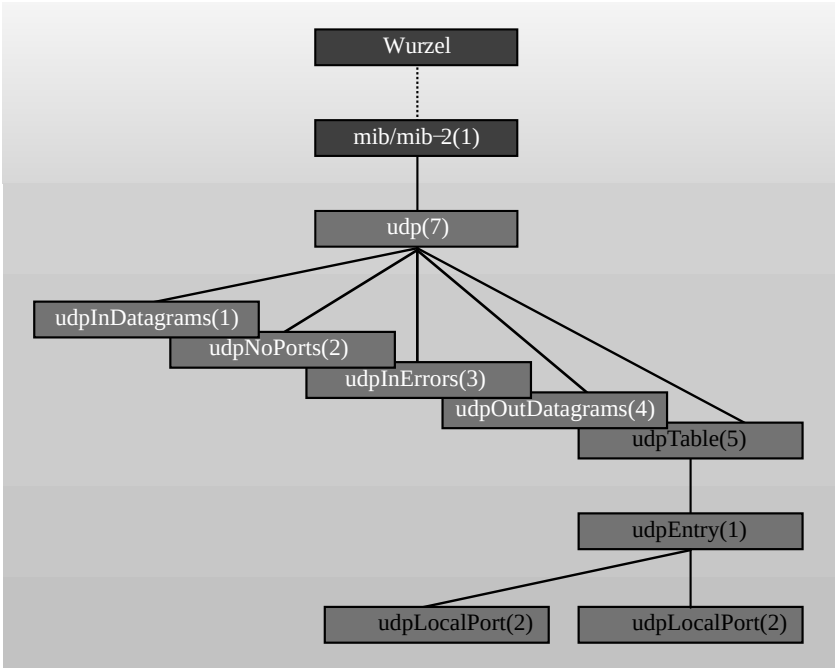

Abb. 4.22. *udp*-Zweig des MIB-I/MIB-II Baumes. Dunkel beschriftete Objekte sind nur in der MIB-II definiert.

egp (mib.8)

Abbildung 4.23 veranschaulicht den *egp* Zweig der MIB-I/MIB-II, der ebenfalls mit der Kategorie *ip* vergleichbar ist. Er enthält fünf verschiedene Angaben zum System, die sich auf das Exterior Gateway Protocol (EGP) [130] beziehen. Die MIB-II ergänzt ein weiteres Attribut, so dass sich insgesamt folgende sechs Einträge ergeben:

egpInMsgs (egp.1). Beinhaltet die Summe aller erfolgreich vom System empfangenen EGP Pakete.

egpInErrors (egp.2). Liefert die Summe aller fehlerhaft empfangenen EGP Pakete.

***egpOutMsgs* (*egp.3*).** Zähler, welcher die Summe aller gesendeten EGP Pakete enthält.

***egpOutErrors* (*egp.4*).** Entspricht der Summe aller auf Grund von zu hoher Systemauslastung nicht gesendeten EGP Pakete.

***egpNeighTable* (*egp.5*).** Enthält eine Tabelle mit Informationen zu EGP Kommunikationspartnern. Unterhalb der Tabelle befindet sich analog dem Schema aus Abbildung 4.3 nur das Spaltendefinitionsobjekt *epgNeighEntry* (*egpNeighTable.1*), unter welchem die einzelnen Spalten angeordnet sind. Jeder Zeileneintrag der Tabelle besteht aus zwei Angaben, die durch die MIB-II auf 15 Einträge erweitert wurden:

- *egpNeighState(1)* Enthält den Statuszustand des Systems in Bezug auf den angegebenen EGP Kommunikationspartner, wie er in RFC 904 beschrieben ist. Mögliche Werte sind *idle*, *acquisition*, *down*, *up* und *cease*.
- *egpNeighAddr(2)* IP Adresse des EGP Kommunikationspartners.
- *egpNeighAs(3)* Identifikationsnummer des Autonomen Systems (AS) dieses EGP Kommunikationspartners.
- *egpNeighInMsgs(4)* Zähler mit der Summe aller von diesem Kommunikationspartner erfolgreich empfangenen EGP Pakete.
- *egpNeighInErrs(5)* Zähler mit der Summe aller von diesem Kommunikationspartner fehlerhaft empfangenen EGP Pakete.
- *egpNeighOutMsgs(6)* Summe aller zu diesem Kommunikationspartner gesendeten EGP Pakete.
- *egpNeighOutErrs(7)* Summe aller EGP Pakete, die auf Grund zu hoher Systemauslastung nicht zu diesem Kommunikationspartner gesendet werden konnten.
- *egpNeighInErrMsgs(8)* Zähler mit der Summe aller von diesem Kommunikationspartner erhaltenen EGP Fehlermeldungen.
- *egpNeighOutErrMsgs(9)* Zähler mit der Summe aller zu diesem Kommunikationspartner gesendeten EGP Fehlermeldungen.
- *egpNeighStateUps(10)* Summe aller Zustandsänderungen in den Zustand *up* zu diesem EGP Kommunikationspartner.
- *egpNeighStateDowns(11)* Summe aller Zustandsänderungen vom Zustand *up* in einen anderen Zustand zu diesem EGP Kommunikationspartner.
- *egpNeighIntervalHello(12)* Anzahl der Hundertstelsekunden zwischen zwei zu sendenden EGP „Hello“ Paketen.
- *egpNeighIntervalPoll(13)* Anzahl der Hundertstelsekunden zwischen zwei zu sendenden EGP „Poll“ Paketen.
- *egpNeighMode(14)* Polling-Zustand dieses EGP Kommunikationspartners (*active* oder *passive*).
- *egpNeighEventTrigger(15)* Ermöglicht das Aktivieren oder Deaktivieren der Nachbarerkennung zu diesem EGP Kommunikationspartner.

egpAs (egp.6). Liefert die Identifikationsnummer des Autonomen Systems (AS), zu dem dieses System gehört.

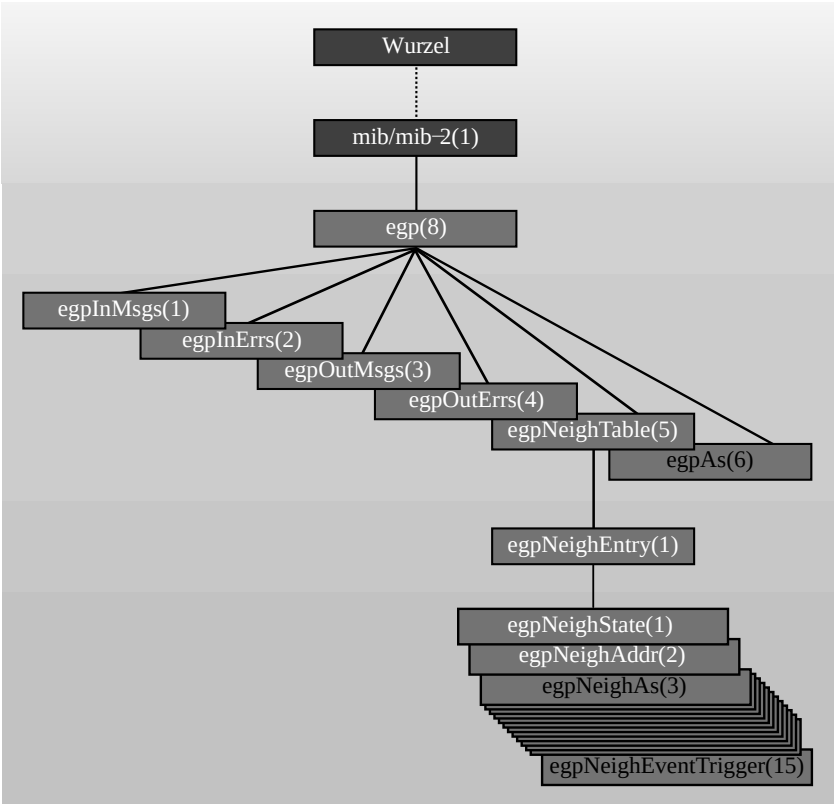

Abb. 4.23. *egp*-Zweig des MIB-I/MIB-II Baumes. Dunkel beschriftete Objekte sind nur in der MIB-II definiert.

cmot (mib-2.9)

cmot ist ein veralteter, nicht mehr verwendeter Eintrag der MIB-II.

transmission (mib-2.10)

transmission ist ein Platzhalter der MIB-II für zukünftige Einträge, die Auskunft über das Übertragungsmedium der einzelnen Schnittstellen des Systems geben sollen. Im Laufe der vergangenen Jahre sind viele verschiedene Objekte unterhalb dieser OID entstanden, die in zahlreichen RFCs definiert wurden. Dazu zählen auch die folgenden Objekte:

- *x25* (*transmission.5*): X.25
- *dot3* (*transmission.7*): Ethernet (IEEE 802.3)
- *dot5* (*transmission.9*): Token Ring (IEEE 802.5)
- *fddi* (*transmission.15*): FDDI
- *lapb* (*transmission.16*): X.25 LAPB
- *ds1* (*transmission.18*): DS1/E1
- *isdnMib* (*transmission.20*): ISDN
- *dialControlMib* (*transmission.21*): Dial Control
- *ppp* (*transmission.23*): PPP/LCP
- *ds3* (*transmission.30*): DS3/E3
- *sip* (*transmission.31*): SIP
- *frameRelayDTE* (*transmission.32*): Frame Relay on DTE
- *rs232* (*transmission.33*): RS-232
- *para* (*transmission.34*): Parallel Printers
- *miox* (*transmission.38*): Multiprotocol Interconnect over X.25
- *sonetMIB* (*transmission.39*): SONET/SDH
- *frnetservMIB* (*transmission.44*): Frame Relay Service
- *dot12MIB* (*transmission.45*): IEEE 802.12
- *ipoaMIB* (*transmission.46*): IP/ARP over ATM
- *mfrMib* (*transmission.47*): UNI/NNI Multilink Frame Relay
- *hdsl2ShdslMIB* (*transmission.48*): HSDL2/SHDSL
- *apsMIB* (*transmission.49*): SONET APS
- *ds0* (*transmission.81*): DS0
- *ds0Bundle* (*transmission.82*): DS0 Bundle
- *adslMIB* (*transmission.94*): ADSL
- *l2tp* (*transmission.95*): L2TP
- *vdslMIB* (*transmission.97*): VDSL
- *docsIfMib* (*transmission.127*): MCNS/DOCSIS
- *tunnelMIB* (*transmission.131*): IP Tunnel
- *coffee* (*transmission.132*): Drip-Type Heated Beverage Hardware Devices. Diese MIB zeigt deutlich, dass RFC Autoren durchaus Spaß verstehen, denn es handelt sich hierbei um eine MIB für Kaffeemaschinen.
- *optIfMibModule* (*transmission.133*): Optical Interface
- *etherWisMIB* (*transmission.134*): WAN Interface Sublayer
- *mplsStdMIB* (*transmission.166*): MPLS
- *vdslExtSCMMIB* (*transmission.228*): VDSL using SCM
- *vdslExtMCMMIB* (*transmission.229*): VDSL using MCM

snmp (*mib-2.11*)

Abbildung 4.24 veranschaulicht den *snmp* Zweig der MIB-II. Diese Kategorie enthält in ihrer aktuellen Version in RFC 3418 [166] nur noch 8 der ursprünglich 28 verschiedene Angaben aus RFC 1213, die sich auf das Protokoll

SNMP und dessen Verwendung beziehen. Die Attribute können dabei sowohl auf Agenten als auch auf Managementstationen zutreffen.

***snmpInPkts* (*snmp.1*).** Liefert die Summe aller vom System empfangenen SNMP Pakete.

***snmpOutPkts* (*snmp.2*).** Zähler, welcher die Summe aller gesendeten SNMP Pakete enthält. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInBadVersions* (*snmp.3*).** Beinhaltet einen Zähler, welcher die Summe aller SNMP Pakete enthält, die für eine nicht unterstützte SNMP Version ausgelegt sind.

***snmpInBadCommunityNames* (*snmp.4*).** Zähler, welcher die Summe aller SNMP Pakete enthält, die einen unbekannten und ungültigen Community Namen enthalten. Dies ist vergleichbar mit der Angabe eines falschen Passwortes.

***snmpInBadCommunitUses* (*snmp.5*).** Enthält die Summe aller mit einem korrekten Community Namen versehenen empfangenen Pakete, für welche die angeforderte Operation mit dem angegebenen Community Namen nicht erlaubt war. Beispielsweise fällt ein SNMP Paket mit einer gültigen „Read-Community“, welches eine Schreiboperation anfordert, unter diese Kategorie.

***snmpInASNParseErrs* (*snmp.6*).** Zähler, welcher die Summe aller empfangenen SNMP Pakete enthält, die einen Syntaxfehler aufwiesen.

***snmp.7*.** Nicht verwendet.

***snmpInTooBigs* (*snmp.8*).** Liefert die Summe aller empfangenen SNMP Pakete mit der Fehlermeldung „tooBig“. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInNoSuchNames* (*snmp.9*).** Liefert die Summe aller empfangenen SNMP Pakete mit der Fehlermeldung „noSuchName“. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInBadValue* (*snmp.10*).** Liefert die Summe aller empfangenen SNMP Pakete mit der Fehlermeldung „badValue“. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInReadOnlies* (*snmp.11*).** Liefert die Summe aller empfangenen SNMP Pakete mit der Fehlermeldung „readOnly“. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInGenErrs* (*snmp.12*).** Liefert die Summe aller empfangenen SNMP Pakete mit der Fehlermeldung „genErr“. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInTotalReqVars* (*snmp.13*).** Zähler, welcher die Summe aller durch gültige Leseanfragen erhaltenen MIB Objekte enthält. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInTotalSetVars* (*snmp.14*).** Zähler, der die Summe aller durch gültige Schreibanfragen erfolgreich geänderten MIB Objekte enthält. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInGetRequests* (*snmp.15*).** Zähler, der die Summe aller empfangenen gültigen SNMP Leseanfragen des Typs „get" enthält. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInGetNexts* (*snmp.16*).** Liefert die Summe aller empfangenen gültigen SNMP Leseanfragen des Typs „get-next". Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInSetRequests* (*snmp.17*).** Zähler, der die Summe aller empfangenen gültigen SNMP Schreibanfragen vom Typ „set" enthält. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInGetResponses* (*snmp.18*).** Enthält die Summe aller empfangenen gültigen SNMP Antworten auf Anfragen des Typs „get". Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpInTraps* (*snmp.19*).** Enthält die Summe aller empfangenen gültigen SNMP Nachrichten (Traps). Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutTooBigs* (*snmp.20*).** Liefert die Summe aller erzeugten SNMP Pakete, die eine Fehlermeldung vom Typ „tooBig" ausgelöst haben. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutNoSuchNames* (*snmp.21*).** Liefert die Summe aller erzeugten SNMP Pakete, die eine Fehlermeldung vom Typ „noSuchName" ausgelöst haben. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutBadValue* (*snmp.22*).** Liefert die Summe aller erzeugten SNMP Pakete, die eine Fehlermeldung vom Typ „badValue" ausgelöst haben. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutReadOnlies* (*snmp.23*).** Nicht verwendet.

***snmpOutGenErrs* (*snmp.24*).** Liefert die Summe aller erzeugten SNMP Pakete, die eine Fehlermeldung vom Typ „genErr" ausgelöst haben. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutGetRequests* (*snmp.25*).** Liefert die Summe aller erzeugten SNMP-Anfragen des Typs „get". Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutGetNexts* (*snmp.26*).** Liefert die Summe aller erzeugten SNMP-Anfragen des Typs „get-next". Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutSetRequests* (*snmp.27*).** Liefert die Summe aller erzeugten SNMP-Anfragen des Typs „set“. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutGetResponses* (*snmp.28*).** Enthält die Summe aller erzeugten SNMP Antworten auf eine Anfrage des Typs „get“. Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpOutTraps* (*snmp.29*).** Enthält die Summe aller erzeugten SNMP Nachrichten (Traps). Dieses Objekt gilt seit RFC 3418 als veraltet.

***snmpEnableAuthenTraps* (*snmp.30*).** Gibt an, ob das System Fehlermeldungen des Typs „authenticationFailure“ senden darf.

***snmpSilentDrops* (*snmp.31*).** Zähler über die Summe aller empfangenen SNMP Anfragen, die vom System verworfen werden mussten, da die zu erzeugende Antwort die maximal zulässige Paketgröße überschritten hätte. Dieses Objekt ist erst in RFC 1907 [32] zum *snmp* MIB Baum hinzugefügt worden.

***snmpProxyDrops* (*snmp.32*).** Im Gegensatz zu *snmpSilentDrops* werden bei *snmpProxyDrops* die verworfenen Pakete gezählt, zu denen die Antworten nicht erfolgreich an den ursprünglichen Proxy zurückgesendet werden konnten. Auch diese OID ist erst in RFC 1907 neu definiert worden.

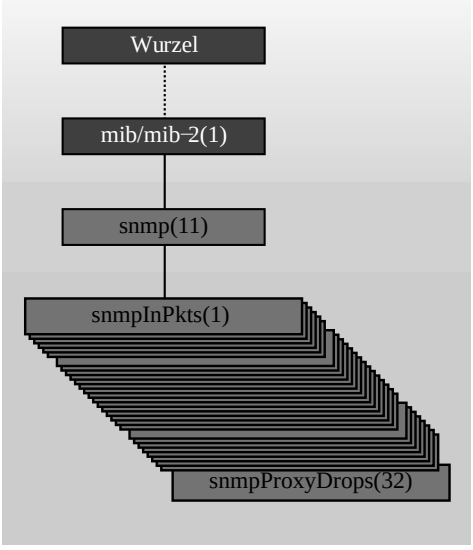

Abb. 4.24. *snmp*-Zweig des MIB-II Baumes.

4.5 SNMP Versionen

Im Laufe der Zeit gewannen immer mehr Sicherheitsaspekte an Bedeutung, so dass sich aufbauend auf der Version 1 des SNMP Protokolls weitere Varianten bildeten. Jede dieser Varianten betrachtet jeweils andere Sicherheitsaspekte, jedoch ist keine dieser Varianten als Standard in die Request for Comments (RFCs) eingegangen. Dies erklärt auch, warum nicht nur eine einzelne Version 2 des SNMP Protokolls existiert, sondern gleich vier verschiedene Varianten bekannt sind (SNMPv2p, SNMPv2c, SNMPv2u und SNMPv2*). Es bleibt allerdings festzuhalten, dass SNMPv2c sich gegenüber den anderen Varianten durchgesetzt hat und damit weitgehend als SNMPv2 anerkannt ist[10]. Version 3 des Simple Network Management Protocols (SNMPv3) gilt als der heutige Standard und ist in erster Linie für seine Sicherheitsfunktionalitäten bekannt.

4.5.1 SNMP Version 1

Version 1 des Simple Network Management Protocols in seiner letzten Version definiert in RFC 1157 [28] einen Rahmen, zu dem verschiedene Standards zählen. Wichtigste Bestandteile sind die Management Information Base aus der RFC 1156 [124] und deren aktualisierte Fassung in RFC 1212 [185], in welcher die verschiedenen zu verwaltenden Objekte definiert sind (siehe auch Abschnitt 4.4.1), sowie die Structure of Management Information aus der RFC 1155 [184], welche die Syntax der Definitionssprache beschreibt, in der alle MIBs anzugeben sind (siehe auch Abschnitt 4.3.1). Erst die Kombination aus diesen drei Standards ergibt ein voll funktionstüchtiges Netzwerkmanagement.

In der SNMPv1 RFC 1157 wird vor allem beschrieben, über welche Kommunikationswege die MIBs eines SNMP Agenten abgerufen oder verändert werden können. Zu diesem Zweck wird die Kommunikation zwischen SNMP Agenten und Managementstationen näher definiert. Zu Beginn dieses Kapitels wurde bereits beschrieben, dass die SNMP Kommunikation über das Protokoll UDP verläuft. SNMP Anfragen verwenden den UDP Port 161; SNMP Nachrichten den Port 162. Jedes SNMP Paket setzt sich ferner aus genau drei Hauptbestandteilen zusammen:

1. eine Ganzzahl, welche die SNMP Version angibt (in diesem Fall „1“)
2. eine Zeichenkette, welche einen Community Namen angibt
3. ein Datenblock, der beliebige Daten enthalten kann.

Der Community Name stammt aus einer Zeit, in der das Internet deutlich weniger ausgeprägt war, als es heute ist. Damals definierte man eine zusammengehörende Gruppe aus einem MIB-Unterbaum auf einem SNMP Agenten

[10] Wenn im Folgenden von SNMPv2 gesprochen wird, dann ist damit SNMPv2c gemeint.

und einer beliebigen Anzahl SNMP Managementstationen als eine Community[11]. Jede Netzwerkmanagementstation kann darüber hinaus zu mehreren Communities gehören und jeder SNMP Agent kann mehrere Communities sogar mit unterschiedlichen Berechtigungen unterstützen. Damit im Netzwerk die SNMP Pakete eindeutig den Gesprächspartnern dieser Community zugeordnet werden können, wird jedes der Pakete mit dem entsprechenden Community Namen versehen. Da dieser Community Name im Klartext in jedem Paket vorhanden ist, kann er nicht ernsthaft die Aufgabe einer Autorisierung, Authentifizierung oder Verschlüsselung übernehmen. Das ist der Grund, warum spätere SNMP Versionen entwickelt wurden, die vor allem weitere Sicherheitsmechanismen implementieren.

Die Kernkomponente eines SNMP Paketes liegt sicherlich im Datenblock, welcher die eigentlichen SNMP Nachrichten und Anfragen enthält. Diese werden auch Protocol Data Unit (PDU) genannt. Die Sicherheit ist in SNMPv1 nicht gänzlich unberücksichtigt geblieben. Das SNMPv1 Rahmenwerk hat zumindest vorgesehen, dass die einzelnen PDUs nicht zwangsweise im Klartext den einzigen Bestandteil des Datenblocks bilden müssen. In RFC 1157 wird eine optionaler Authentifizierungsmethode angesprochen, die beim Empfang eines SNMP Paketes nicht nur auf den richtigen Community Namen achtet, sondern auch beispielsweise mittels Verschlüsselungsverfahren das Paket als authentisch identifiziert. Die Entscheidung über die Berechtigungen einer Managementstation liegen dann nicht nur in einem Klartext „Passwort". Diese Authentifizierungsmethode wurde in SNMPv1 jedoch nicht ausformuliert, so dass in der Praxis nicht davon Gebrauch gemacht wurde. Folgerichtig werden auch die PDUs unverschlüsselt in den Datenblock eines SNMPv1 Paketes eingebettet und sind für jeden lesbar. SNMPv1 definiert genau fünf verschiedene Arten von PDUs, über welche die vollständige SNMP Kommunikation abgewickelt wird.

get-request PDU

Eine *get-request* Anfrage wird von einer Managementstation initiiert und ist eine Anfrage an einen SNMP Agenten mit der Bitte um Auskunft über den Inhalt einer angegebenen OID. Die gesendete PDU enthält vier Informationen: eine eindeutige Identifikationsnummer (`request-id`) zur Identifizierung der Anfrage, einen Fehlerzustand (`error-status`) und einen Fehlerindex (`error-index`), die immer Null sind, sowie eine Variablenbindung (`variable-bindings`), die aus Objekt-Wert-Paaren besteht. Bei den Paaren aus Objekt und Wert steht das Objekt für die abzufragende OID und der Wert ist für das Paket irrelevant, da er durch diese PDU vom SNMP Agenten abgefragt werden soll.

[11] Eine direkte Übersetzung des englischen Wortes „Community" ins Deutsche ist schwierig. Sowohl die Bedeutung „Kommune" als auch andere Bedeutungen wie „Gemeinde" oder „Gemeinschaft" laufen Gefahr, eher verwirrend zu wirken. Aus diesem Grund wird hier weiterhin von „Community" gesprochen.

get-next-request PDU

Die *get-next-request* Anfrage ist nahezu identisch zur *get-request* Anfrage. Sie enthält dieselben Informationen, nur dass als Antwort nicht der Inhalt der abgefragten OID, sondern der im MIB-Baum nachstehenden OID geliefert wird.

get-response PDU

get-response beschreibt nicht die Anfrage einer SNMP Managementstation an einen SNMP Agenten, sondern die Antwort des Agenten auf eine derartige Anfrage. Die Inhaltsfelder sind dieselben wie auch bei den *get-request* und *get-next-request* Anfragen. Lediglich Fehlerzustand und Fehlerindex sind mit entsprechenden Werten gefüllt. Der Fehlerindex weist gegebenenfalls auf diejenige Variable hin, welche den Fehler verursacht hat.

set-request PDU

Die *set-request* Anfrage wird von einer Managementstation zu einem SNMP Agenten gesendet und enthält eine Bitte, den Wert einer angegebenen OID auf den mitgelieferten Wert abzuändern. Die vier Inhaltsfelder der PDU sind identisch zu *get-request* und *get-next-request* Anfragen gefüllt.

trap PDU

Ein Trap ist eine SNMP Nachricht, die vom SNMP Agenten selbst initiiert wird und an eine Managementstation gerichtet ist, die mit dieser Nachricht über ein besonderes Ereignis unterrichtet werden soll. Der Inhalt einer SNMP Nachricht unterscheidet sich deutlich vom Inhalt der anderen PDUs. Enthalten sind in einer Trap PDU sechs verschiedene Felder. Das `enterprise` Feld enthält eine OID mit der Herkunft des Objektes, welches die Nachricht veranlasst hat. Das `agent-addr` Feld beinhaltet die Netzwerkadresse des SNMP Agenten, `generic-trap` und `specific-trap` geben den Typ der SNMP Nachricht an (siehe auch Abschnitt 4.3.1), `time-stamp` liefert die Zeitspanne, die seit der letzten Initialisierung des SNMP Agenten verstrichen ist, und `variable-bindings` enthält wiederum Objekt-Wert-Paare mit Informationen zur SNMP Nachricht.

4.5.2 SNMP Version 2

Wie schon SNMPv1 ist auch die zweite Version des Simple Network Management Protocol durch ein ganzes Rahmenwerk aus verschiedenen Definitionen zusammengesetzt. An dieser Stelle sei noch einmal wiederholt, dass kein SNMPv2 Standard in den RFCs definiert ist, dass jedoch SNMPv2c weitläufig

als SNMPv2 Standard angesehen wird. Aus diesem Grund widmet sich dieses Kapitel auch dem Community-Based SNMP.

In vielerlei Hinsicht ist SNMPv2 offener und flexibler gestaltet, als es die erste Version von SNMP war. Das in RFC 1901 [31] definierte Rahmenwerk setzt sich deshalb auch aus weiteren sechs RFCs zusammen, die jeweils einzelne Teilaspekte von SNMPv2 beleuchten.

SMI für SNMPv2

In RFC 1902 [34], die später durch die RFC 2578 [120] ersetzt wurde, ist die SMIv2 definiert worden, welche als die Definitionssprache für alle SNMPv2 MIBs vorgeschrieben ist (siehe auch Abschnitt 4.3.2).

Textuelle Konventionen für SNMPv2

Die RFC 1903 [35] gilt als Ergänzung zur SMIv2 und enthält textuelle Konventionen, die in der SMIv2 verwendet werden können. Zu diesen Makros zählen beispielsweise die `STATUS`, `DESCRIPTION` oder `SYNTAX` Attribute. RFC 1903 ist in einer aktualisierten Version in der RFC 2579 [121] vorhanden.

Konformitätsangaben für SNMPv2

Auch die RFC 1904 [30] ist eine Ergänzung zur SMIv2 und enthält vor allem die wichtigen Neuerungen der SMIv2 gegenüber der SMIv1. Diese sind die `OBJECT-GROUP` und `NOTIFICATION-GROUP` Blöcke, der `MODULE-COMPLIANCE` Block sowie der `AGENT-CAPABILITIES` Block. Wie schon in Abschnitt 4.3.2 besprochen, existiert eine aktualisierte Version zu den Konformitätsangaben in der RFC 2580 [119].

Protokolloperationen für SNMPv2

In der RFC 1905 [33], die in einer aktualisierten Version in der RFC 3416 [167] existiert, werden hauptsächlich Angaben zu den SNMP Paketen, deren Aufbau sowie zu einer neuen Proxy-Funktionalität gemacht. Unter der Proxy-Funktionalität ist die Eigenschaft eines SNMP Agenten zu verstehen, der SNMP Anfragen für einen anderen Agenten entgegennimmt und diese an den eigentlichen Adressaten weiterleitet. Gleiches gilt auch für die SNMP Antwortpakete. Im Wesentlichen sind in SNMPv2 die PDUs identisch zu denen in SNMPv1 definiert. Lediglich der neue PDU Typ Bulk-PDU hat eine etwas andere Form. Insgesamt sind bei SNMPv2 acht verschiedene Arten von PDUs erlaubt:

***get-request* PDU.** Die *get-request* Anfrage in SNMPv2 ist identisch zu SNMPv1 definiert.

***get-next-request* PDU.** Auch die *get-next-request* Anfrage in SNMPv2 ist identisch zu SNMPv1 definiert. Der Unterschied zur *get-request* Anfrage besteht lediglich im zurückgelieferten Objekt und dessen Wert.

***get-bulk-request* PDU.** Ein neuer PDU Typ ist die *get-bulk-request* Anfrage, mit der eine Managementstation mehr als nur ein Objekt mit dessen Wert beim SNMP Agenten abfragen kann. Zu diesem Zweck sind in der PDU die beiden Attribute `error-status` und `error-index`, die bei einer Leseoperation ohne Bedeutung sind, durch die beiden Attribute `non-repeaters` und `max-repetitions` ersetzt worden. Einfach gesprochen geben diese beiden Werte die minimale und die maximale Anzahl an übertragenen Objekten in der Bulk-Anfrage an.

***response* PDU.** Die Antwort eines SNMP Agenten auf die SNMP Anfrage eines SNMP Managers ist in der *response* PDU enthalten, die identisch zu SNMPv1 definiert ist. Zusätzlich sind in RFC 1905 die genauen Fehlertypen definiert, mit denen ein SNMP Agent im `error-status` Feld antworten kann. Tabelle 4.14 gibt weitere Auskunft.

***set-request* PDU.** Wie auch in SNMPv1 beschreibt die *set-request* PDU eine Anfrage einer Managementstation mit der Bitte um Änderung des Wertes der angegebenen OID.

***snmpV2-trap* PDU.** Im Gegensatz zur Version 1 von SNMP, bei welcher die Trap Nachrichten einen besonderen PDU Typ verwendet haben, wird in SNMPv2 derselbe PDU Typ wie für alle anderen SNMP Pakete auch verwendet. Lediglich die beiden ersten Variablenbindungen sind bereits mit den OIDs *sysUpTime.0* und *snmpTrapOID.0* vorgegeben.

***inform-request* PDU.** Der neue PDU Typ *inform-request* bezeichnet eine Nachricht, die von einer Managementstation zu einer anderen Managementstation gesendet wird, um diese mit Informationen zu versorgen. Hierbei handelt es sich im Regelfall um die Weiterleitung einer von einer Managementstation empfangenen SNMP Nachricht an einen anderen SNMP Manager.

***report* PDU.** Der *report* PDU Typ ist in SNMPv2 nicht näher spezifiziert und bietet zukünftigen Implementierungen von SNMP Managementstationen und Agenten Raum für Erweiterungen.

Transportwege für SNMPv2

In der RFC 1906 [36] werden Angaben zu Protokollen gemacht, über die SNMP Nachrichten versendet werden können. Zwar gilt noch immer unverändert die bereits in SNMPv1 festgelegte Verwendung des Protokolls UDP als wichtigster Kommunikationsweg, der von allen SNMP-fähigen Geräten unterstützt werden sollte, jedoch sind in SNMPv2 auch andere Kommunikationswege erlaubt. Zu diesen gehören:

Tabelle 4.14. Mögliche Fehlermeldungen einer *response* PDU eines SNMP Agenten auf die Anfrage einer SNMP Managementstation.

Art der Abweichung	Beschreibung
noError(0)	Es ist kein Fehler aufgetreten
tooBig(1)	Das zu generierende Antwortpaket würde die Beschränkung für die Maximallänge überschreiten.
noSuchName(2)	MIB Objekt existiert nicht (Proxy)
badValue(3)	Ungültiger Wert (Proxy)
readOnly(4)	Nur Lesezugriff erlaubt (Proxy)
genErr(5)	Nicht weiter spezifizierter Fehler
noAccess(6)	Zugriff zur OID zum Schreiben nicht erlaubt
wrongType(7)	Falscher Typ für einen Schreibzugriff
wrongLength(8)	Zu schreibender Inhalt außerhalb des gültigen Wertebereiches
wrongEncoding(9)	ASN.1 Kodierungsfehler beim Schreibzugriff
wrongValue(10)	Zu schreibender Wert ist nicht erlaubt
noCreation(11)	Zu schreibendes Objekt existiert nicht und kann auch nicht angelegt werden
inconsistentValue(12)	Zu schreibender Wert ist aktuell nicht erlaubt
resourceUnavailable(13)	Wert kann nicht geschrieben werden, da eine dazu benötigte Ressource nicht zur Verfügung steht
commitFailed(14)	Schreibvorgang wurde abgebrochen
undoFailed(15)	Schreibvorgang wurde abgebrochen und ein Teil der Objekt-Neuzuweisungen konnte nicht rückgängig gemacht werden
authorizationError(16)	Fehler bei der Autorisierung
notWritable(17)	Wert ist nicht schreibbar
inconsistentName(18)	Objekt existiert nicht

- der OSI Connectionless-Mode Transport Service (CLTS).
- das AppleTalk Datagram Delivery Protocol (DDP). Hier werden vor allem Probleme bei der Umsetzung der beiden völlig verschiedenen Namensräume der TCP/IP Protokollgruppe und der AppleTalk Protokollgruppe beschrieben. Außerdem wird beschrieben, wie das AppleTalk Name Binding Protocol (NBP) diese Problematik bearbeiten soll.
- das Novell Internetwork Packet Exchange Protokoll (IPX).

Insbesondere die verschiedenen Protokolle und Architekturen machen eine exakte Definition der genauen Zusammensetzung und Transformation der PDUs besonders wichtig. In diesem Zusammenhang gibt RFC 1906 ein Beispiel (siehe Abbildung 4.25), wie eine konkrete in ASN.1 angegebene SNMP PDU nach den Regeln der BER in eine Byte-Folge übersetzt wird, die dann als Nutzdaten in die Datagramme der jeweiligen Protokolle eingeht (siehe Abbildung 4.26).

```
[5] IMPLICIT SEQUENCE {
        request-id      1414684022,
        non-repeaters   1,
        max-repetitions 2,
        variable-bindings {
            { name sysUpTime,
              value { unSpecified NULL } },
            { name ipNetToMediaPhysAddress,
              value { unSpecified NULL } },
            { name ipNetToMediaType,
              value { unSpecified NULL } }
        }
    }
```

Abb. 4.25. Beispiel für eine mittels ASN.1 kodierte PDU aus RFC 1906.

```
[5] IMPLICIT SEQUENCE          a5 82 00 39
    INTEGER                    02 04 54 52 5d 76
    INTEGER                    02 01 01
    INTEGER                    02 01 02
    SEQUENCE (OF)              30 2b
        SEQUENCE               30 0b
            OBJECT IDENTIFIER  06 07 2b 06 01 02 01 01 03
            NULL               05 00
        SEQUENCE               30 0d
            OBJECT IDENTIFIER  06 09 2b 06 01 02 01 04 16 01 02
            NULL               05 00
        SEQUENCE               30 0d
            OBJECT IDENTIFIER  06 09 2b 06 01 02 01 04 16 01 04
            NULL               05 00
```

Abb. 4.26. Mittels BER kodierte ASN.1 Beispiel PDU aus RFC 1906.

Eine geringfügig veränderte und aktualisierte Version des Standards steht mit RFC 3417 [168] zur Verfügung.

MIB für SNMPv2

Die letzte zum SNMPv2 Rahmenwerk gehörende RFC 1907 [32] definiert die Management Information Base (MIB-II), die mit SNMPv2 zusammen verwendet wird. Eine genauere Beschreibung der MIB-II findet sich in Abschnitt 4.4.2. In RFC 3418 [166] liegt eine geringfügig aktualisierte und fehlerkorrigierte Version der MIB-II vor.

4.5.3 SNMP Version 3

Das Ziel von SNMPv3 war es, die bereits in SNMPv1 gelegten Grundsteine bezüglich Sicherheit, die in SNMPv2 von unterschiedlichen Gruppen auf unterschiedliche Weise behandelt und ausgeführt worden waren, in ein einheitliches Konzept zusammenzuführen. Die vielen Varianten von SNMPv2, von denen letztlich keine als Standard übernommen wurde, werden in der SNMPv3 erneut aufgegriffen, entsprechend erweitert und als neuer Standard veröffentlicht. SNMPv3 ist damit das ideale Protokoll für ein sicheres Netzwerkmanagement.

Im Zuge der Aufarbeitung der älteren SNMP Versionen wurde in RFC 3411 [81] ein völlig neues Rahmenwerk aufgestellt, welches die Bestandteile von SNMP teilweise ganz anders benennt. Die aus SNMPv1 und SNMPv2 bekannten Managementstationen und SNMP Agenten haben in SNMPv3 einen allgemeineren Namen erhalten, nämlich SNMP Entitäten („SNMP Entity"). Zur besseren Verständlichkeit sollen hier weiterhin die Begriffe Managementstation und SNMP Agent verwendet werden. Im Folgenden sollen als eine vereinfachende Darstellung die Managementstation und der SNMP Agent in einem Schichtenmodell ähnlich dem ISO OSI Referenzmodell [51] abgebildet werden[12].

In der obersten Schicht des OSI Referenzmodells (Anwendungsschicht) befinden sich die Applikationen eines Systems. Diese Schicht lässt sich direkt in das hier eingeführte SNMP Schichtenmodell übertragen. In SNMPv3 werden die Entitäten nach den Anwendungen unterschieden, die sie unterstützen. Eine Managementstation beinhaltet in der SNMP Applikationsschicht typischerweise die Applikationen *Anfrage-Ersteller* und *Nachrichten-Empfänger*, wohingegen ein SNMP Agent üblicherweise die Anwendungen *Antwort-Ersteller* und *Nachrichten-Ersteller* unterstützt. Ein SNMP Agent kann zusätzlich die Applikation *Weiterleiter* besitzen, wenn er auch als Proxy fungiert. Außerdem befinden sich in der Applikationsschicht des SNMP Agenten sämtliche unterstützten MIBs. Eine Managementstation kann zusätzlich die Anwendung *Nachrichten-Ersteller* implementieren, wenn sie SNMP Nachrichten an andere Managementstationen übermittelt.

Unmittelbar unterhalb der Applikationsschicht befindet sich die Zugriffsschicht. Das darin befindliche Zugriffskontrollsystem („Access Control Subsystem") kontrolliert in einem SNMP Agenten den Zugriff auf die einzelnen Objekte der implementierten MIBs. Die Zugriffsschicht ist demnach nur bei einem SNMP Agenten wichtig und mit dem entsprechenden Subsystem belegt.

Die unmittelbar unterhalb der Zugriffsschicht liegenden Schichten des SNMPv3 Modells werden in RFC 3411 zu einem SNMP Prozessor („SNMP Engine") zusammengefasst, welcher die SNMP Pakete vom Netzwerk zu den

[12] Tatsächlich kann man bei SNMPv3 nicht exakt von einem Schichtenmodell sprechen. Korrektere Darstellungen für eine typische Netzwerkmanagementstation und einen typischen SNMP Agenten findet sich in den Kapiteln 3.1.3.1 und 3.1.3.2 der RFC 3411.

einzelnen Applikationen und zurück leitet. Dieser Prozessor übernimmt verschiedene Aufgaben, die man sich jeweils in einer einzelnen SNMP-Schicht vorstellen kann:

Die oberste Schicht im SNMP Prozessor, die direkt unterhalb der Applikationsschicht und Zugriffsschicht liegt, beinhaltet die Sicherheitsschicht mit dem Sicherheitssystem („Security Subsystem"). In dieser Schicht werden Authentifizierung und Verschlüsselung behandelt. Für eine sichere Kommunikation zwischen Managementstation und SNMP Agent muss diese Schicht bei allen Entitäten implementiert werden.

Die nächstniedrigere Meldungsschicht kümmert sich weniger um Sicherheit als um Kompatibilität. SNMPv3 unterstützt explizit die Pakettypen der alten Versionen SNMPv1 und SNMPv2. In dieser Schicht befinden sich die Meldungstransformierer („Message Processing Subsystem"), welche die SNMP Pakete von ihrem speziellen versionsabhängigen Format in ein einheitliches Format übersetzen können. Beim Versenden von Meldungen geschieht dies im jeweiligen Meldungstransformierer auch umgekehrt.

Die unterste Schicht des SNMP Prozessors schließlich beinhaltet die Versandeinheit („Dispatcher"), welche sich um die Weiterleitung empfangener SNMP Pakete an den korrekten Meldungstransformierer sowie um die Weiterleitung der zu sendenden Pakete an das korrekte Netzwerkprotokoll kümmert. Die Versandschicht ist eng mit der Meldungsschicht verknüpft und genauso wie diese in jeder SNMP Entität notwendig.

Ganz unten im SNMP Schichtenmodell – unterhalb des SNMP Prozessors – liegt die Transportschicht. SNMPv3 unterstützt wie auch SNMPv2 nicht nur das Protokoll UDP der TCP/IP Protokollgruppe, sondern auch verschiedene andere Protokolle. Die Transportschicht übernimmt die Kodierung in und Dekodierung vom jeweils verwendeten Protokoll.

Abbildung 4.27 veranschaulicht das komplette SNMP Schichtenmodell, das an dieser Stelle eingeführt wurde, um das SNMPv3 Rahmenwerk besser veranschaulichen zu können. Es sei noch einmal betont, dass eine Abbildung des SNMPv3 Rahmenwerks in ein Schichtenmodell nicht vollständig korrekt ist. Die verschachtelte Struktur des SNMP Prozessors liefert einen klaren Hinweis darauf.

Im Folgenden soll das SNMPv3 Rahmenwerk beispielhaft an einem eingehenden SNMP Lesezugriff eines SNMP Agenten erläutert werden. Das zugehörige SNMP Paket läuft also über das Netzwerk in die erste SNMP Schicht (Transportschicht) ein und wird bis in die sechste Schicht (Applikationsschicht) weitergeleitet. Dort wird dann die zugehörige Antwort generiert und wieder zurück in die Transportschicht geleitet, bis sie über das Netzwerk zur ursprünglichen SNMP Entität zurückgesendet wird.

Transportschicht

Bei SNMPv3 kann ein Paket über verschiedene Protokolle einen SNMP Agenten erreichen. RFC 3417 [168] gibt Auskunft über die vier bislang definierten

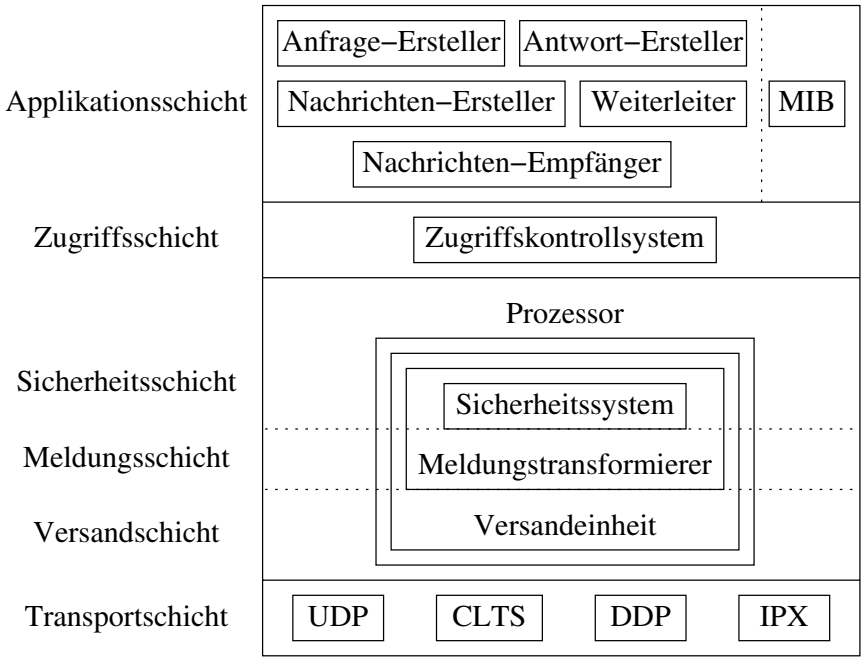

Abb. 4.27. Schematische Darstellung des SNMP Schichtenmodells zur Veranschaulichung des SNMPv3 Rahmenwerkes.

Protokolle UDP, OSI CLTS, AppleTalk DDP und Novell IPX. Andere Protokolle können in Zukunft ergänzt werden. RFC 3430 [190] beschreibt beispielsweise die Verwendung des Protokolls TCP. In der Transportschicht werden vor allem die BER kodierten Nachrichten in eine ASN.1 Kodierung übersetzt. Von der untersten SNMP Schicht erreicht das Paket dann die Versandeinheit im SNMP Prozessor.

Versandeinheit

Damit ein SNMP-fähiges Gerät die Forderung von SNMPv3 erfüllen kann, alle vorherigen SNMP Versionen zu unterstützen, muss das Gerät eine Versandeinheit („Message Dispatcher“) besitzen. An dieser Stelle zeigt sich die Unzulänglichkeit des hier eingeführten Schichtenmodells. Die Versandeinheit besitzt nämlich zwei Aufgaben, von denen eine an den jeweils benötigten Meldungstransformierer delegiert wird. Zunächst identifiziert die Versandeinheit die verwendete SNMP Version der von der Transportschicht eingehenden Meldung. Diese ASN.1 kodierte Meldung muss als nächstes in eine gültige PDU übersetzt werden. Zu diesem Zweck liefert die Versandeinheit das Paket an den korrekten Meldungstransformierer in der dritten Schicht weiter. Danach erhält die Versandeinheit die korrekte PDU zurück, um sie der Zugriffsschicht

zuzuführen. Die dabei eingesetzten Protokolloperationen folgen bei SNMPv3 uneingeschränkt den Vorgaben von SNMPv2, die in der RFC 3416 [167] definiert sind. Abbildung 4.27 verdeutlicht noch einmal die verschachtelte Aufgabenteilung der Versandschicht und der Meldungsschicht.

Eine korrekte Beschreibung der Versandeinheit mit „Message Dispatcher“ und „PDU Dispatcher“ findet sich in RFC 3412 [38].

Meldungstransformierer

Für jede unterstützte SNMP Version muss ein SNMPv3-fähiges Gerät einen eigenen Meldungstransformierer („Message Processing Subsystem“) besitzen. Dieser kennt die in der jeweiligen SNMP Version gültigen Anfragen und Nachrichtenformate und kann diese in ein einheitliches Format hin und zurück transformieren. Konkret wird die ASN.1 kodierte Meldung in eine der SNMP Version entsprechende PDU übersetzt. Die einzige aktuell unterstützte SNMP Version ist SNMPv3, aber andere in der Zukunft unterstützte Formate wie SNMPv1, SNMPv2c, SNMPv2u und SNMPv2* sind denkbar. Eine nähere Definition findet sich in RFC 3412 [38].

Gegenüber SNMPv1 und SNMPv2 hat sich das Format der SNMPv3 Pakete geändert. Ein SNMPv3 Paket besitzt folgende Attribute:

- eine Ganzzahl, welche die SNMP Version angibt (in diesem Falle „3“)
- einen globalen Paketkopf, der folgende vier Informationen enthält:
  - eine Ganzzahl mit der eindeutigen Identifikationsnummer des Paketes
  - eine Ganzzahl mit der maximalen vom Sender unterstützten Paketgröße
  - ein Oktett mit verschiedenen Flags wie *authFlag* und *privFlag* mit Informationen zum Sicherheitsniveau
  - eine Ganzzahl mit der Angabe des verwendeten Sicherheitsmodells
- eine Zeichenkette mit zusätzlichen Angaben zum Sicherheitsmodell
- einen Datenblock, der beliebige Daten enthalten kann.

Für eine korrekte Transformation der eingegangenen Meldung in eine PDU muss je nach eingesetztem Sicherheitslevel der empfangene Datenblock zuvor entschlüsselt werden. Gegebenenfalls delegiert der Meldungstransformierer diese Aufgabe an das Sicherheitssystem in der vierten Schicht des SNMP Schichtenmodells. Nach erfolgreicher Dekodierung besitzt der Meldungstransformierer eine entschlüsselte PDU, die er zur weiteren Verarbeitung zurück an die Versandeinheit sendet. Die dreifache Verschachtelung innerhalb des SNMP Prozessors wird in Abbildung 4.27 verdeutlicht.

Sicherheitssystem

Zur Implementierung von Authentifizierungsmethoden und Verschlüsselungsmechanismen besitzt jedes SNMPv3-fähige Gerät ein Sicherheitssystem. Dieses kann mit einem oder mehreren Sicherheitsmodellen versehen sein, die jeweils einen Schutz gegen verschiedene Bedrohungen realisieren. Zur Zeit ist

nur das benutzerorientierte User-Based Security Model (USM) in RFC 3414 [18] definiert. Es können jedoch auch die Sicherheitsmodelle von SNMPv1 oder SNMPv2c verwendet werden, deren Sicherheitsgewinn jedoch fraglich ist.

Das Sicherheitssystem in SNMPv3 wurde vorrangig eingeführt, um die für die Vorgängerversionen SNMPv1 und SNMPv2 identifizierten Sicherheitsbedrohungen zu bekämpfen. Es wurden insgesamt sechs Bedrohungen ausgemacht, die jeweils verschieden priorisiert und mit unterschiedlich starken Mitteln bekämpft werden. Tabelle 4.15 beschreibt die verschiedenen Bedrohungen und ordnet sie in das etwas allgemeinere Schema aus Bedrohungen und Angriffen in Teil III dieses Buches ein.

Tabelle 4.15. Identifizierte Bedrohungen aus SNMPv1 und SNMPv2, die zum Sicherheitsmodell in SNMPv3 geführt haben.

Bedrohung	Beschreibung	Verweis
Ein Verfälschen von Informationen	Eine SNMP Entität ohne die notwendige Autorisierung verändert den Inhalt einer SNMP Anfrage oder Nachricht beispielsweise mit dem Ziel, einen Objektwert unerlaubt zu verändern	Abschnitt 9.1.3
Eine Identität vortäuschen	Die Zugriffsbeschränkungen einer SNMP Managementaufgabe für einen konkreten Benutzer werden durch Vortäuschen einer anderen autorisierten Identität umgangen	Abschnitt 9.1.4
Bekanntwerden von Informationen	Durch Abfangen von SNMP Paketen werden Managementinformationen bekannt	Abschnitt 9.1.2
Verfälschen von einem Datenstrom	SNMP Pakete werden außerhalb des Normalbereiches umsortiert, verzögert oder wiederholt	Abschnitt 9.1.3
Denial of Service	Verschiedene Szenarien, die dafür sorgen, dass der SNMP Dienst für autorisierte Benutzer nicht zur Verfügung steht	Abschnitt 9.1.1
Datenstromanalyse	Die Informationen in SNMP Paketen werden durch hinreichende Analyse des Datenstroms berechnet	Abschnitt 9.2.2

Für die ersten vier Bedrohungen liefert das USM aus SNMPv3 einen Schutzmechanismus; die beiden letzten Punkte stellen eher Angriffe dar, für die kein gesonderter Schutz eingerichtet wurde. Implementierte Sicherheitsmodelle sind das Aktualitätsmodul („Timeliness Module“), das Authentifizierungsmodul („Authentication Module“) und das Vertraulichkeitsmodul („Privacy Module“). Das Aktualitätsmodul ist unmittelbar mit dem Authentifizierungsmodul verknüpft, so dass sich schließlich zwei Hauptmodule ergeben, welche die Sicherheitsfunktionen in SNMPv3 abbilden. Beiden Modellen ist

gemein, dass sie unterschiedliche Benutzer mittels deren Benutzernamen unterstützen.

Authentifizierungsmodul. Das Authentifizierungsmodul in SNMPv3 arbeitet mit einem Authentifizierungsprotokoll, von dem zur Zeit genau zwei definiert sind: das HMAC-MD5-96 Protokoll und das HMAC-SHA-96 Protokoll. Im SNMPv3 Sicherheitsmodell wird der Hash-Based Message Authentication Code (HMAC) [106] zusammen mit der Einweg-Hashfunktion Message Digest 5 (MD5) [175] verwendet, wobei die Ausgabe auf 96 Bit abgeschnitten wird. HMAC arbeitet mit einem geheimen Schlüssel, den zwei Kommunikationspartner teilen, und einer Einweg-Hashfunktion. Anstelle der MD5 Einweg-Hashfunktion kann auch der etwas sicherere Secure Hash Algorithm 1 (SHA-1) [136] verwendet werden. Mit den in SNMPv3 verwendeten Authentifizierungsprotokollen kann sichergestellt werden, dass nur SNMP Entitäten eine gültige Meldung erstellen können, die über den geheimen Schlüssel verfügen. Auf diese Art kann eine korrekte Authentifizierung der Benutzer und der sendenden SNMP Entität durchgeführt werden. Ein klarer Nachteil, der nicht verschwiegen werden soll, findet sich in der Verwendung von geheimen Schlüsseln. Sind diese einmal kompromittiert – und eine Verteilung an alle beteiligten Kommunikationspartner erhöht auch das Risiko einer Kompromittierung – so können auch andere Entitäten gültige SNMP Meldungen erstellen.

Aktualitätsmodul. Wie bereits erwähnt, hängt das Aktualitätsmodul direkt mit dem Authentifizierungsmodul zusammen. Um die Aktualität einer SNMP Meldung bestimmen zu können, werden zwei Werte mit jedem authentifizierten Paket mitgesendet. Es handelt sich um die beiden Werte *snmpEngineBoots* und *snmpEngineTime*, aus deren Kombination die Erstellungszeit des Paketes unabhängig von der lokalen Uhrzeit eindeutig ermittelbar ist. *snmpEngineBoots* enthält die Anzahl der Neuinitialisierungen der SNMP Entität und *snmpEngineTime* enthält die Anzahl der Sekunden seit der letzten Neuinitialisierung, bei der auch der Zähler *snmpEngineBoots* inkrementiert worden ist.

Vertraulichkeitsmodul. Im Vertraulichkeitsmodul von SNMPv3 ist aktuell nur ein einziger Verschlüsselungsalgorithmus definiert, jedoch können zukünftig je nach Bedarf weitere neue Algorithmen hinzugefügt werden. Momentan steht der Cipher Block Chaining Data Encryption Standard (CBC-DES) zur Verfügung. Der Blockchiffrierungsalgorithmus DES gilt weithin als besonders sicher, und er lässt sich nur durch Ausprobieren des gesamten Schlüsselraumes knacken. Die Verwendung des Rückkopplungsmodus CBC sorgt weiterhin, dass zwei identische Klartextblöcke nicht denselben kodierten Datenblock ergeben.

Sicherheitsniveau. Zur Identifizierung der in einem SNMP Paket verwendeten Sicherheitsmodelle enthält dieses noch zwei Flags. Es sind allerdings nur drei der vier möglichen Kombinationen erlaubt und im Sicherheitsniveau zusammengefasst:

1. Das Sicherheitsniveau *noAuthNoPriv* beinhaltet weder Authentifizierung noch Verschlüsselung.
2. Das Sicherheitsniveau *authNoPriv* beinhaltet Authentifizierung aber nicht Verschlüsselung.
3. Das Sicherheitsniveau *authPriv* beinhaltet sowohl Authentifizierung als auch Verschlüsselung.

Zugriffskontrollsystem

Nachdem ein SNMP Paket erfolgreich die Sicherheitsschicht des SNMP Schichtenmodells durchlaufen hat, steht als Endergebnis des SNMP Prozessors eine entschlüsselte PDU, deren Authentizität und Aktualität gegebenenfalls bereits geprüft worden ist. Der SNMP Prozessor übergibt diese PDU nicht an die jeweilige Applikation ohne zuvor eine Zugangskontrolle zu initiieren. Dazu erhält das Zugriffskontrollsystem in der fünften Schicht die PDU. Hier werden die dekodierten PDUs verarbeitet und über die Berechtigungen der in ihr enthaltenen Anfragen entschieden. Da eine Zugriffskontrolle nur in unmittelbarem Zusammenhang mit einer MIB stehen kann, ist die Zugriffsschicht auch nur bei SNMP Agenten zu implementieren. Nur dieser kann über die Berechtigung für einen Zugriff auf Teile der MIB entscheiden. Aktuell unterstützt und in RFC 3415 [227] definiert ist das View-Based Access Control Model (VACM), das den Zugriff über so genannte Sichten reguliert. Die in der VACM gespeicherten Zugriffsdaten sind außerdem teilweise via SNMP administrierbar.

Um das VACM möglichst verständlich beschreiben zu können, hilft ein Blick auf die interne Abarbeitung einer *isAccessAllowed* Anfrage an das Zugriffskontrollsystem.

Kontext. Zunächst wird überprüft, ob der angegebene Kontext in der internen Kontext-Tabelle *vacmContextTable* enthalten ist. Unter einem Kontext versteht SNMPv3 einen individuellen Satz an MIB Objekten, die von einem SNMP Agenten unterstützt werden. Ein SNMP Agent darf dabei mehrere Kontexte unterstützen, die sich auch überlappen dürfen. Dies bedeutet, dass sich OIDs durchaus in mehreren Kontexten befinden dürfen. Das System aus Kontexten liefert beispielsweise eine elegante Lösung für einen SNMP Proxy. Ein SNMP (Proxy)-Agent kann auf diese Weise mehrere identische MIBs unterstützen, jedoch jeweils in einem anderen Kontext. Eine Managementstation, welche die Proxy-Funktionalität des SNMP Agenten anspricht, um Informationen von einem anderen SNMP Agenten zu erhalten, müsste in diesem Fall lediglich den Kontext entsprechend dem gewünschten Zielsystem setzen.

Sicherheitsgruppe. Findet sich im SNMP Paket ein gültiger Kontext, über welchen der SNMP Agent Informationen besitzt, so wird als nächstes die Sicherheitsgruppe bestimmt, in der sich der gegebenenfalls authentifizierte Benutzer befindet. Die Sicherheitsgruppe definiert sich aus der Kombination des verwendeten Sicherheitsmodells und des Sicherheitsnamens. Wird das

SNMPv3 Sicherheitsmodell USM verwendet, so ist der Sicherheitsname identisch zum Benutzernamen. Wird das Community-basierte Sicherheitsmodell der SNMP Vorgängerversionen verwendet, so ist der Sicherheitsname identisch mit dem Community Namen. Der SNMP Agent hält eine Liste aller gültigen Sicherheitsgruppen in der internen Tabelle *vacmSecurityToGroupTable*.

Berechtigung. Ist auch die ermittelte Sicherheitsgruppe gültig, so setzt das Zugriffskontrollsystem seine Arbeit fort mit der Überprüfung der zugewiesenen Berechtigung. Zu diesem Zweck wird die Tabelle *vacmAccessTable* mit der Wertekombination Kontext, Sicherheitsgruppe, Sicherheitsmodell und Sicherheitsniveau befragt. Wie bereits beim Sicherheitssystem beschrieben, existieren nur die drei gültigen Werte *noAuthNoPriv*, *authNoPriv* und *authPriv* für das Sicherheitsniveau. Existiert ein gültiger Eintrag in der *vacmAccessTable*, so besitzt das Zugriffskontrollsystem nun die korrekte Zugriffserlaubnis, die entweder *read*, *write* oder *notify* lauten kann.

Sicht. Als letzter Schritt wird die zum gewünschten Zugriff zugehörige Sicht ermittelt. Eine Sicht enthält dabei immer Kombinationen aus einer Zugriffserlaubnis und einer Gruppe von MIB Objekten. Nur wenn eine entsprechende Sicht zum angegebenen Objekt mit der gewünschten Zugriffserlaubnis existiert, liefert das Zugriffskontrollsystem als Antwort ein *accessAllowed* zurück.

MIB für SNMPv3

In der sechsten und obersten SNMP Schicht finden sich bei jeder SNMP Entität die jeweils implementierten Applikationen und zusätzlich bei jedem SNMP Agenten ein oder mehrere implementierte MIBs. Abbildung 4.28 zeigt noch einmal zusammenfassend den schematischen Aufbau einer SNMP Managementstation, und in Abbildung 4.29 ist der schematische Aufbau eines SNMP Agenten dargestellt.
SNMPv3 unterstützt jede mit SMIv1 oder SMIv2 definierte MIB. Dies können neben den Standard MIBs aus den RFCs (siehe Anhang A.2) auch herstellereigene MIBs für die verschiedensten Netzwerkgeräte sein. Die Anzahl aller zur Zeit unterstützten MIBs ist immens groß und praktisch nicht ermittelbar.

SMI für SNMPv3

Wie bereits erwähnt, sind in der SNMPv3 viele Dinge von den vorhergehenden SNMP Versionen übernommen worden. Zu diesen zählt auch die Structure of Management Information in der Version 2 (SMIv2) aus den RFCs 2578-2580, die bereits für SNMPv2 in unveränderter Form Gültigkeit hatte (siehe Abschnitt 4.3.2). Es wird empfohlen, neue MIBs in der SMIv2 zu spezifizieren; aus Gründen der Kompatibilität werden von SNMPv3 aber auch MIBs unterstützt, die in der älteren SMIv1 erstellt wurden.

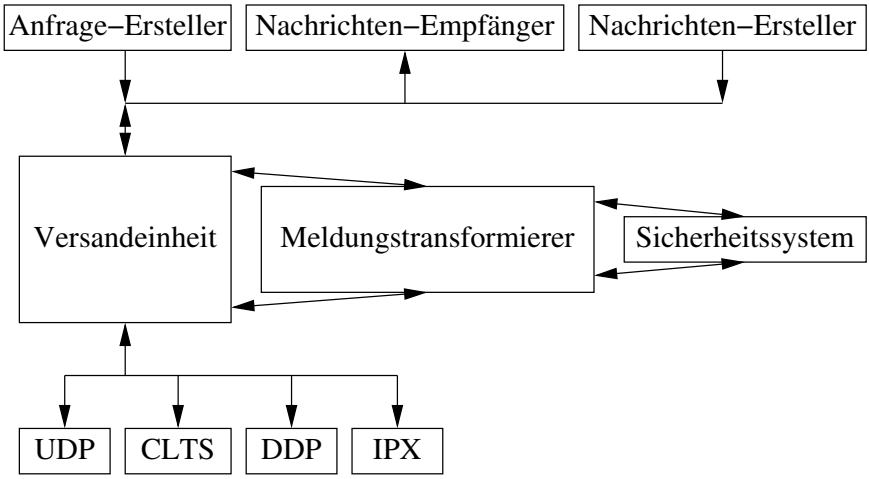

Abb. 4.28. Schematische Darstellung einer SNMP Managementstation.

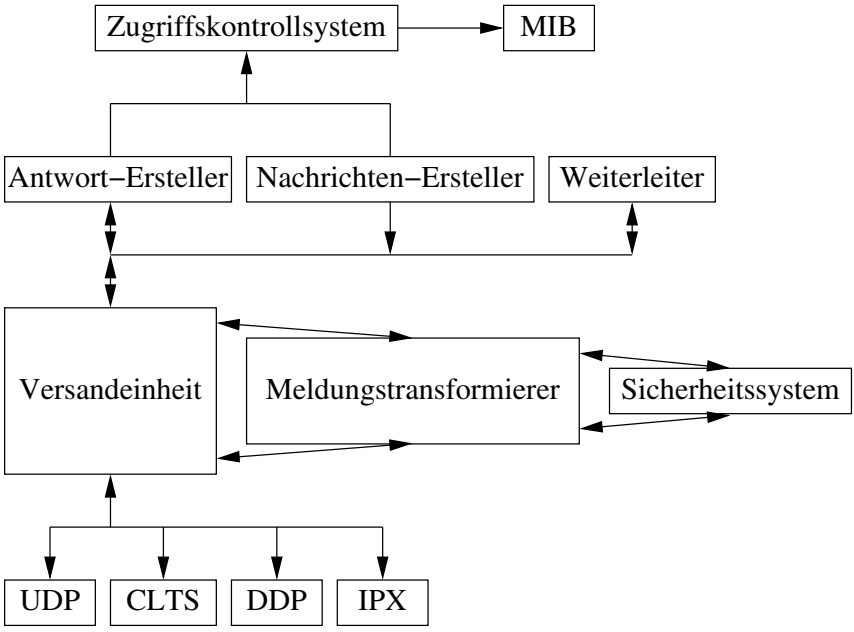

Abb. 4.29. Schematische Darstellung eines SNMP Agenten.

5

Logging

Einer der ersten und bis heute wichtigsten Mechanismen zur Überwachung von Systemen und Anwendungen ist das Protokollieren („Logging“). Bereits bei der Entwicklung von Software ist ein ausführliches Logging unverzichtbar. Einerseits lässt sich durch lückenloses Protokollieren aller internen Vorgänge ein einwandfreies und korrektes Verhalten der Anwendung nachweisen. Andererseits kann ein entdeckter Fehler durch vollständiges Logging eingekreist und identifiziert werden. Eine derartige Fehleranalyse und Fehlerbeseitigung über Anwendungsprotokolle nennt man auch „Debugging“[1]. Je nach erwartetem Aufwand für das Debugging werden die Daten in einer Datei zur weiteren Verarbeitung gespeichert, oder aber es genügt eine flüchtige Speicherung der Log-Meldungen beispielsweise auf dem Bildschirm.

Bei Computer-Betriebssystemen findet eine Protokollierung oftmals lokal auf jedem System statt. Dort landen alle Nachrichten vom Betriebssystemkern über Mitteilungen der Systemdienste bis hin zu Meldungen der einzelnen Anwendungsprogramme. Dieser Mechanismus nennt sich Ereignisprotokollierung oder bei Unix-artigen Betriebssystemen SYSLOG oder in einer neueren Version auch SYSLOG-NG.

Auch Netzwerke lassen sich über den SYSLOG Mechanismus überwachen, da eine Protokollierung auch über das Netzwerk möglich ist. In diesem Fall senden die Netzwerkkomponenten Nachrichten an einen oder mehrere SYSLOG Server. In allen Fällen werden die Informationen aber nur in eine Richtung vom Sender der Nachrichten zum Empfänger transportiert. Die Reaktion auf die gesendeten Daten muss über einen anderen Mechanismus erfolgen. Das Logging kann demnach nur zur Netzwerküberwachung eingesetzt werden.

[1] Der Begriff „Debugging“, „De-Bugging“ lässt sich wörtlich mit „Entwanzen“ übersetzen. Gemeint ist das Entfernen von „Bugs“ (Wanzen), also den Fehlern in einer Software.

5.1 syslog

Der SYSLOG Dienst wird bereits seit mehreren Jahrzehnten erfolgreich eingesetzt. Seinen Ursprung hat er im Berkley Software Distribution (BSD) [24] Betriebssystem, das an der University of California entwickelt wurde. Viele der später entworfenen Systemarchitekturen verwenden bis in die heutige Zeit noch immer den SYSLOG Dienst zur Protokollierung von Systemmeldungen. Selbst Komponenten, die nicht über einen lokalen Nachrichtenspeicher verfügen, sind oftmals dennoch für das Senden von SYSLOG Meldungen über das Netzwerk ausgelegt. Es wurde jedoch in der Vergangenheit versäumt, ein eindeutiges und verbindliches Nachrichtenformat für SYSLOG zu definieren, so dass sich verschiedene Dialekte entwickelt haben, die teilweise nicht mit einander kompatibel sind. Auch die im Jahr 2001 entstandene RFC 3164 [109] gilt nicht als Internet-Standard, sondern hat lediglich informellen Charakter. Zumindest werden dort aber einige Empfehlungen zum SYSLOG Mechanismus und den gesendeten Meldungen gegeben. Eine Erweiterung des SYSLOG Mechanismus ist zudem in RFC 3195 [140] beschrieben.

5.1.1 Transportmechanismus

Wie auch schon bei SNMP ist beim SYSLOG Konstrukt die Performanz höher als die Zuverlässigkeit der Nachrichtenauslieferung priorisiert. Aus diesem Grund verwendet SYSLOG ebenfalls das verbindungslose Protokoll UDP für die Übermittlung der Systemmeldungen. Ein SYSLOG Server sollte den dafür reservierten Port 514 verwenden. Die Sender von SYSLOG Meldungen sollten als Absende-Port ebenfalls den Port 514 verwenden.

In RFC 3195 wird der SYSLOG Mechanismus durch zwei Transportmechanismen auf der Basis von TCP erweitert. Auf Grund des verbindungsorientierten Charakters von TCP lassen sich somit SYSLOG Nachrichten verlässlich ausliefern. Der verwendete TCP-Port trägt die Nummer 601.

5.1.2 Architektur

In der SYSLOG Architektur sind zwei verschiedene Komponenten vorgesehen. Eine Komponente, die Nachrichten versendet, wird lediglich Gerät („Device“) genannt. Ein Nachrichten-Empfänger erhält den Namen Nachrichtensammler („Collector“). Zusätzlich kann es noch Geräte geben, die SYSLOG Nachrichten entgegennehmen und an andere Komponenten weiterleiten. Diese Komponenten werden als Relaisstationen („Relay“) bezeichnet. In der SYSLOG Architektur können Nachrichten von einem Gerät an beliebig viele Nachrichtensammler und über beliebig viele Relaisstationen gesendet werden. Abbildung 5.1 zeigt ein Beispiel für eine komplexere SYSLOG Architektur.

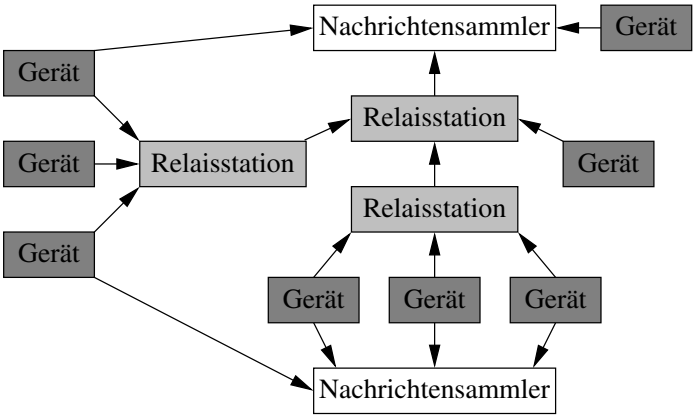

Abb. 5.1. Verzweigte SYSLOG Architektur mit mehreren Geräten, Relaisstationen und Nachrichtensammlern.

5.1.3 Kritikalität

Jede SYSLOG Meldung sollte mit einer der vordefinierten Kritikalitäten („Severity“) versehen sein. Anhand dieser Kritikalität kann ein Nachrichtensammler Entscheidungen über die Bearbeitungspriorität der Nachricht treffen. Insgesamt stehen die acht in Tabelle 5.1 aufgelisteten Kritikalitäten zur Verfügung.

Tabelle 5.1. Beschreibung der acht möglichen Kritikalitäten für SYSLOG Nachrichten.

Kritikalität	Beschreibung
emergency(0)	Das angegebene System ist zur Zeit unbrauchbar
alert(1)	Es besteht dringender Handlungsbedarf für das System
critical(2)	Das System befindet sich in einem kritischen Zustand
error(3)	Das angegebene System weist einen Fehler auf
warning(4)	Es wurde eine Warnung für das System herausgegeben
notice(5)	Das System arbeitet in der Nähe der definierten Grenzwerte
informational(6)	Das angegebene System macht eine unkritische Meldung
debug(7)	Es werden Debub-Informationen vom System übermittelt

5.1.4 Nachrichtenherkunft

Aus Performanz-Gründen erfolgt die Angabe der Kritikalität in einem SYSLOG Paket kombiniert mit der zweiten wichtigen Angabe in einer SYSLOG Nachricht. Um den Ersteller einer SYSLOG Meldung genauer identifizieren zu können, wird in jedes Paket eine Nachrichtenherkunft („Facility“) integriert.

Auch für diese Angabe existieren bereits vorgegebene Werte, die in Tabelle 5.2 aufgelistet sind. Hier zeigt sich besonders deutlich das Fehlen eines einheitlichen Standards, da Nachrichten derselben Kategorie teilweise mit verschiedenen Facility-Werten gekennzeichnet sind. Außerdem sind die Zuordnungen der Facitily-Werte nicht auf jedem System identisch. Ein Nachteil der Unterteilung aller SYSLOG Meldungen in die wenigen vorgegebenen Facility-Werte ist zwar eine nicht ausreichend feingranulare Abstufung der Pakete, jedoch liefert SYSLOG auch die acht Facility-Werte *local0* bis *local7* die vom System an beliebige Dienste frei vergeben werden können.

Tabelle 5.2. Beschreibung der möglichen Facility-Werte für die Angabe einer Nachrichtenherkunft in SYSLOG Nachrichten.

Facility	Beschreibung
kern(0)	Meldungen des Betriebssystemkerns
user(1)	SYSLOG Nachrichten von Applikationen des Benutzers
mail(2)	Pakete des Mail-Systems
daemon(3)	Meldungen von Systemdiensten
auth(4)	Nachrichten aus dem Bereich Sicherheit und Authentifizierung
syslog(5)	SYSLOG-eigene Pakete
lpr(6)	Meldungen des Druckersystems
news(7)	Pakete des Nachrichtendienstes
uucp(8)	Nachrichten des Unix-to-Unix Communications Package (UUCP)
cron(9)	Meldungen des CRON Dienstes
authpriv(10)	Nachrichten aus dem Bereich Sicherheit und Authentifizierung
ftp(11)	SYSLOG Pakete von File Transfer Protocol (FTP) Servern
ntp(12)	SYSLOG Meldungen von Network Time Protocol (NTP) Servern
audit(13)	Meldungen vom Typ „Log Audit“
alert(14)	Meldungen vom Typ „Log Alert“
at(15)	Nachrichten des Automatisierungsdienstes AT
local0(16)	Vom Benutzer frei definierbarer Meldungstyp
local1(17)	Vom Benutzer frei definierbarer Meldungstyp
local2(18)	Vom Benutzer frei definierbarer Meldungstyp
local3(19)	Vom Benutzer frei definierbarer Meldungstyp
local4(20)	Vom Benutzer frei definierbarer Meldungstyp
local5(21)	Vom Benutzer frei definierbarer Meldungstyp
local6(22)	Vom Benutzer frei definierbarer Meldungstyp
local7(23)	Vom Benutzer frei definierbarer Meldungstyp

5.1.5 Paketformate

Das genaue Format von SYSLOG Paketen ist nirgendwo exakt definiert. Da verwundert es auch nicht, dass eine Vielzahl an verschiedenen, teilweise inkompatiblen Nachrichtenformaten zu finden sind. Um diese Problematik zu-

mindest teilweise zu entschärfen, können Relaisstationen die erhaltenen SYSLOG Meldungen geringfügig abgeändert an die nächste Relaisstation oder den nächsten Nachrichtensammler weiterleiten. Die Relaisstationen übernehmen damit auch die Aufgabe eines Nachrichtentransformierers, der für kompatible Nachrichten sorgt. Die Änderungsmöglichkeit für Relaisstationen beschränken sich allerdings auf das Hinzufügen von Angaben im SYSLOG Paketkopf. Der Meldungsinhalt darf auf keinen Fall verändert werden.

Die Vielfalt an verschiedenen verwendeten SYSLOG Nachrichtenformaten wirkt sich verkomplizierend auf die Behandlung von SYSLOG Meldungen in einem Nachrichtensammler oder einer Relaisstation aus. Als allgemein anerkannt hat sich aber das in RFC 3164 beschriebene Format aus den drei Informationsblöcken PRI, HEADER und MSG herauskristallisiert[2]:

1. Die ersten Oktette eines SYSLOG Pakets sollen eine Prioritätenangabe enthalten. Die Priorität eines einzelnen Paketes errechnet sich dabei aus der Formel:

$$\text{Priorität} = 8 \times \text{Facility} + \text{Kritikalität} \tag{5.1}$$

   Für eine korrekte Prioritätenangabe PRI wird die berechnete Priorität als eine Dezimalzahl aus ein bis drei Ziffern (ohne führende Nullen) in spitzen Klammern (‚<' und ‚>') eingefasst. Daraus ergibt sich eine Prioritätenangabe mit einer variablen Länge von drei bis fünf Oktetten.
2. Unmittelbar nach der Prioritätenangabe sollte der Meldungskopf HEADER folgen, der aus zwei durch ein Leerzeichen voneinander getrennten Angaben besteht. Bei der ersten Angabe handelt es sich um eine Zeitangabe in dem in Tabelle 5.3 beschriebenen Format. Die zweite Angabe beinhaltet den Hostnamen des Senders oder dessen IP Adresse, wenn kein Hostname vorhanden ist.
3. Der letzte Part eines SYSLOG Paketes enthält die eigentliche Nachricht MSG, die durch ein Leerzeichen vom HEADER separiert ist. Für die Nachricht bestehen mit Ausnahme einer Beschränkung auf druckbare Zeichen keine besonderen Restriktionen. Es ist lediglich darauf zu achten, dass SYSLOG Meldungen eine Maximallänge von 1024 Byte nicht überschreiten. Außerdem wird der erste Teil der Meldung als Etikett TAG betrachtet, welches den genauen Dienst spezifiziert, der die Nachricht gesendet hat. Als Trennzeichen zwischen TAG und dem Inhalt CONTENT der Meldung kann jedes nicht alphanumerische Zeichen dienen, jedoch wird in der Praxis häufig der Doppelpunkt ‚:' oder die linke eckige Klammer ‚[' verwendet[3].

[2] Unabhängig vom hier angegebenen Format müssen sowohl Nachrichtensammler als auch Relaisstationen mit Nachrichten umgehen können, die keinen korrekten PRI oder HEADER Block enthalten.

[3] Oftmals wird dem Namen des Dienstes die zugehörige Prozess-ID in eckigen Klammern angehängt, beispielsweise *syslog[749]*. In einem solchen Fall gehört die Prozess-ID bereits zum Inhalt der SYSLOG Nachricht.

Tabelle 5.3. Format für Zeitangaben im HEADER einer SYSLOG Nachricht.

Oktette	Inhalt
1-3	Dreibuchstabige, englische Abkürzung des Monatsnamen. Gültige Werte sind *Jan*, *Feb*, *Mar*, *Apr*, *May*, *Jun*, *Jul*, *Aug*, *Sep*, *Oct*, *Nov* oder *Dec*.
4	Leerzeichen (, ')
5-6	Tag des Monats, gegebenenfalls mit einem Leerzeichen an der Stelle der Zehner aufgefüllt. Gültige Werte sind , 1' bis ,31'.
7	Leerzeichen (, ')
8-9	Stunde im 24-Stunden-Format, gegebenenfalls mit einer führenden Null aufgefüllt. Gültige Werte sind ,00' bis ,23'.
10	Doppelpunkt (,:')
11-12	Die gegebenenfalls mit einer führenden Null aufgefüllte Minute. Gültige Werte sind ,00' bis ,59'.
13	Doppelpunkt (,:')
14-15	Die gegebenenfalls mit einer führenden Null aufgefüllte Sekunde. Gültige Werte sind ,00' bis ,59'.

Wie bereits erwähnt, können Relaisstationen SYSLOG Pakete umschreiben. Immer dann, wenn die Syntax des eingehenden Paketes nicht der hier beschrieben Form entspricht, kann die Relaisstation die notwendigen und eventuell fehlenden Daten dem Paket hinzufügen. Dabei taucht potentiell das Problem auf, dass ein Paket nach der Bearbeitung die zulässige Gesamtlänge von 1024 Byte überschreitet. In diesem Fall sollte die Relaisstation den Inhalt der Nachricht entsprechend kürzen. Generell gilt:

- SYSLOG Pakete ohne oder mit ungültiger Prioritätenangabe PRI müssen von einer Relaisstation bearbeitet werden. Es muss die Prioritätenangabe ,<13>' (benutzerdefinierte Meldung mit der Kritikalität *notice*) und die aktuelle Zeitangabe im Meldungskopf HEADER hinzugefügt werden. Außerdem sollte das Paket um einen Hostname oder eine IP Adresse ergänzt werden. Der Inhalt des Originalpakets wird anschließend als Inhalt an den HEADER angehängt und gegebenenfalls auf 1024 Byte gekürzt.
- Pakete, die zwar eine gültige Prioritätenangabe besitzen, die jedoch keine korrekte Zeitangabe aufweisen, müssen um die aktuelle Zeitangabe ergänzt werden. Zusätzlich kann auch ein Hostname oder alternativ einer IP Adresse zum HEADER hinzugefügt werden. Der Rest des Paketes wird als Inhalt der Nachricht interpretiert. Auch hier werden die bearbeiteten Pakete bei Bedarf auf 1024 Byte gekürzt.
- SYSLOG Pakete, die sowohl eine gültige Prioritätenangabe als auch eine korrekte Zeitangabe besitzen, dürfen von einer Relaisstation nur in unveränderter Form weitergeleitet werden.

5.1.6 Sicherheitsaspekte

Die in RFC 3195 beschriebene Ergänzung zum SYSLOG Mechanismus in Form von zwei Transportmechanismen auf der Basis von TCP erweitern das ursprüngliche Konzept um Sicherheitsfunktionalitäten. Grundlage ist das in RFC 3080 [183] definierte Blocks Extensible Exchange Protocol (BEEP). Verschlüsselung wird in BEEP durch das Transport Layer Security (TLS) [60] Protokoll realisiert, während eine Authentifizierung über die Simple Authentication and Security Layer (SASL) [135] Methode abgehandelt wird.

Verschlüsselung

Das TLS Protokoll arbeitet mit symmetrischem Datenaustausch und unterstützt mehrere Verschlüsselungsverfahren. Neben der Null-Verschlüsselung (also keiner Verschlüsselung) sind in RFC 2246 noch die Algorithmen Rivest Cipher Version 2 und 4 (RC2 [178] und RC4 [189]), Data Encryption Standard (DES) [48] und Triple Data Encryption Standard (DES3) [78] sowie eine schwächere Version von DES, bei welcher der Schlüssel von 56 Bit auf 40 Bit verkleinert wird, definiert. Zum Schlüsselaustausch verwendet TLS die jeweils nach ihren Entwicklern benannten asymmetrischen Verschlüsselungsverfahren RSA (Rivest, Shamir, Adleman [179]) und DH (Diffie-Hellman [61]).

Authentifizierung

Die SASL Methode bietet zwar keine Verschlüsselung, jedoch können mit ihr Authentifizierungen durchgeführt werden. Dabei können verschiedene Sicherheitsmechanismen verwendet werden. RFC 2222 führt explizit die drei Mechanismen Kerberos Version 4 (Kerberos4 [203]), Generic Security Service Application Program Interface Version 2 (GSSAPI2 [107]) und (S/KEY [79]).

5.2 syslog-ng

SYSLOG ist zwar in den letzten Jahren als weit verbreiteter Logging Mechanismus anerkannt worden, jedoch besitzt er einige Nachteile, die eine flexible Arbeit schwierig oder teilweise unmöglich machen. Die gravierendste Einschränkung findet sich in der geringen Anzahl an möglichen Facility-Werten. Auf diese Weise ist die Herkunft einer Nachricht oftmals nur sehr ungenau bestimmbar. Erschwert wird diese Problematik durch die Tatsache, dass die für private Zwecke vorgesehenen Facility-Werte *local0* bis *local7* nicht bei allen Diensten konfiguriert werden können. Viele Serverdienste sind auf die einzige explizit vorgesehene Facility *daemon* beschränkt, so dass sich ihre Meldungen nicht klar unterscheiden lassen. Eine Applikation, die SYSLOG Nachrichten

einwandfrei den korrekten Absendern zuordnen will, muss demnach zusätzlich zur Prioritätenangabe auch den Inhalt der SYSLOG Pakete auslesen und auswerten.

Eine zweite Unzulänglichkeit von SYSLOG ist die Beschränkung auf reine Hostnamen im HEADER der Pakete. SYSLOG unterstützt nicht die Angabe von Full Qualified Domain Names (FQDN), die zur eindeutigen Identifizierung des Senders gerade in größeren Netzwerken mit mehreren Domänen äußerst hilfreich sind.

Um all diese Probleme umgehen zu können, wurde SYSLOG-NG entwickelt. Die Grundarchitektur von SYSLOG-NG ist vergleichsweise einfach und dennoch sehr flexibel. Das SYSLOG-NG Rahmenwerk arbeitet mit den drei einfachen Objekttypen *Quelle*, *Ziel* und *Filter*, die durch *Protokollpfade* miteinander verbunden werden. Eine Meldung, die von einer der angegebenen Quellen eingeht und allen Filterregeln des Protokollpfades genügt, wird an alle benannten Ziele weitergeleitet. Zusätzliche Flags können dabei das Verhalten der einzelnen Protokollpfade untereinander näher bestimmen. Abbildung 5.2 zeigt den schematischen Aufbau von Protokollpfaden im SYSLOG-NG Rahmenwerk.

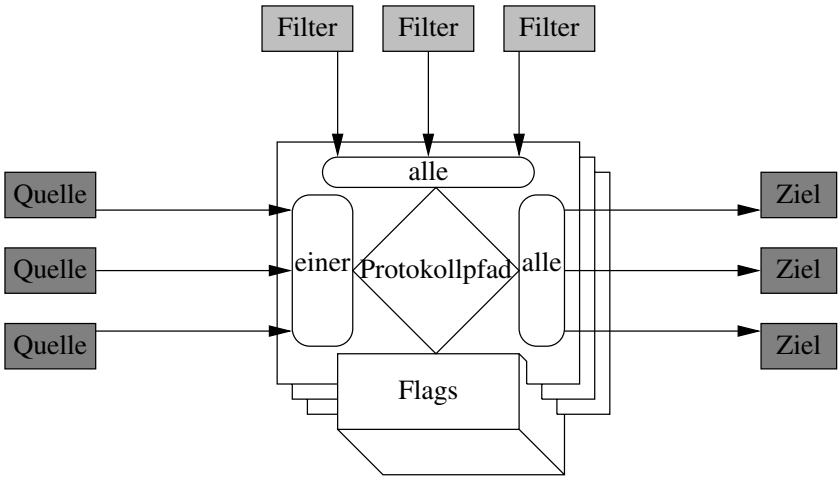

Abb. 5.2. Schematischer Aufbau der SYSLOG-NG Architektur. Zu jedem Protokollpfad sind ein oder mehrere Quellen definiert, von denen Nachrichten eingehen können. Stimmt eine Nachricht mit allen Filtern überein, so wird sie an alle Ziele weitergeleitet. Zusätzlich beeinflussen noch verschiedene Flags das Verhalten der einzelnen Protokollpfade untereinander.

5.2.1 Quelle

Damit SYSLOG-NG zu den verschiedenen Varianten von SYSLOG und vor allem zu den unterschiedlichen Kommunikationswegen innerhalb der meisten

Unix-artigen Betriebssysteme kompatibel ist, sind insgesamt acht mögliche Quellen definiert, von denen SYSLOG-NG Nachrichten entgegennehmen kann. Zur Nachrichtenbehandlung werden in SYSLOG-NG jeweils ein oder mehrere Quellen zu einer benannten Quellgruppe zusammengefasst, die anschließend durch die Protokollpfade angesprochen werden. Jede der Quellen kann zusätzlich mit verschiedenen Parametern versehen werden, um sie näher zu spezifizieren. Tabelle 5.4 listet die globalen Parameter auf, die zu allen acht verschiedenen Quellen angegeben werden können.

Tabelle 5.4. Optionale allgemeine Parameter, die zu jeder der acht möglichen Quellen in SYSLOG-NG angegeben werden können.

Parameter	Beschreibung
log_msg_size	Maximal erlaubte Größe für eingehende Nachrichten
log_iw_size	Maximale Größe des Puffers für eine Nachrichtenflusskontrolle
pad_size	Manche Unix-artigen Betriebssysteme füllen die erzeugten Nachrichten auf feste Werte auf. Um dies zu berücksichtigen, kann der entsprechenden Wert über diesen Parameter angegeben werden.
log_prefix	Mit diesem Parameter können die Quellen noch näher spezifiziert werden. Es lassen sich Meldungs-Prefixe angeben, die eine Nachricht aufweisen muss.
time_zone	Falls eine Quelle keine Zeitzone angibt, wird die mit diesem Parameter definierte Zeitzone verwendet.
log_fetch_limit	Maximale Anzahl an Nachrichten, die bei einer einzigen Anfrage gleichzeitig geholt werden.
follow_freq	Während Datenströme wie Sockets oder Pipes nur Daten liefern, wenn neue Nachrichten ankommen, sind Dateien grundsätzlich auch ohne Änderungen lesbar. Dass eine Quelle nur bei einer tatsächlichen Änderung abgefragt wird, kann über diesen Paraameter spezifiziert werden.
flags	Parameter für zukünftige weitere Meldungsanalysen.

internal

Die SYSLOG-NG Quelle *internal* ist ausschließlich Nachrichten vorbehalten, die von SYSLOG-NG selbst generiert werden. Zu dieser Quelle sind keine besonderen zusätzlichen (außer den in Tabelle 5.4 bereits definierten) Parameter vorgesehen.

unix-stream

Die *unix-stream* Quelle bezeichnet verbindungsorientierte Unix Socket Streams, wie sie vorrangig bei Linux Systemen zur SYSLOG Kommunikation verwendet werden. Typischerweise sind die unter Linux verwendeten Sockets mit

der Datei /dev/log im Verzeichnisbaum verbunden. SYSLOG-NG unterstützt aber auch andere Dateinamen, die über den ersten notwendigen Parameter definiert werden. Zusätzlich zum Dateinamen und den in Tabelle 5.4 angeführten allgemeinen Parametern können noch weitere optionale Parameter definiert werden, die in Tabelle 5.5 aufgelistet sind.

Tabelle 5.5. Optionale Parameter für die Angabe einer *unix-stream* Quelle in SYSLOG-NG.

Parameter	Beschreibung
max-connections	Gibt die maximale Anzahl an Sockets an, die von dieser Quelle parallel geöffnet werden dürfen.
keep-alive	Definiert, ob offene Socket-Verbindungen bei einem Neustart von SYSLOG-NG offen gehalten werden sollen.
owner	Legt die Benutzer ID (uid) des Sockets fest.
group	Setzt die Gruppen ID (gid) des Sockets fest.
perm	Gibt die Berechtigungen des Sockets im üblichen Unix-Format an.

sun-streams

Der SYSLOG Mechanismus hat bei früheren Betriebssystemen der Firma Sun Microsystems [207] eine besondere Art von Streams zur Kommunikation verwendet, die nicht auf Sockets basiert. Repräsentiert werden diese Streams ebenfalls durch eine Datei im Verzeichnisbaum des SYSLOG-NG Gerätes. Um auch diese Nachrichten empfangen zu können, unterstützt SYSLOG-NG die *sun-streams* Quelle. Mit der Angabe des optionalen Parameters door kann auch der aktuelle Mechanismus von Sun genutzt werden, der auf Interprocess Communications (IPC) basiert. Neben diesem optionalen Parameter und dem Pflichtparameter für den Dateinamen können zusätzlich noch die allgemeinen Parameter aus Tabelle 5.4 angegeben werden.

unix-dgram

Ähnlich zur *unix-stream* Quelle ist auch die *unix-dgram* Quelle mit Unix Sockets verbunden, allerdings nicht mit Streams, sondern mit Datagrammen. Analog zum Logging auf verteilte Rechner werden hier die verbindungslosen Datagramme zur Nachrichtenübermittlung verwendet. Ein typischer Name für die als Verknüpfungspunkt des Sockets dienende Datei wäre /var/run/log, es können aber auch andere Namen im vorgeschriebenen ersten Parameter angegeben werden. Zusätzlich können noch die allgemeinen Parameter aus Tabelle 5.4 und die drei in Tabelle 5.6 aufgelisteten Parameter angegeben werden.

Tabelle 5.6. Optionale Parameter für die Angabe einer *unix-dgram* Quelle in SYSLOG-NG.

Parameter	Beschreibung
owner	Legt die Benutzer ID (uid) des Sockets fest.
group	Setzt die Gruppen ID (gid) des Sockets fest.
perm	Gibt die Berechtigungen des Sockets im üblichen Unix-Format an.

file

Nachrichten des Betriebssystemkerns werden häufig in einer speziellen Datei an das System übermittelt, beispielsweise `/dev/kmsg` oder `/proc/kmsg`. Um diese regulären Dateien auslesen zu können, wird die Quelle *file* benötigt, die den Dateinamen als Pflichtparameter erwartet. Ansonsten können nur noch die in Tabelle 5.4 aufgelisteten allgemeinen Parameter angegeben werden.

pipe

Eine spezielle Eigenart der Unix-Variante HP-UX [88] ist der Kommunikationsmechanimus über so genannte Pipes. Ähnlich wie Sockets werden Pipes durch Dateinamen im Verzeichnisbaum repräsentiert. Neben dem einzigen Pflichtparameter zur Angabe des Dateinamen sind nur die allgemeinen Parameter aus Tabelle 5.4 erlaubt.

udp

Um auch Log-Meldungen empfangen zu können, die über das Netzwerk versendet wurden, sind in SYSLOG-NG noch zwei weitere Quellen definiert. Die *udp* Quelle entspricht Nachrichten, die von traditionellen SYSLOG Geräten versendet werden. Ohne weitere Angabe empfängt SYSLOG-NG UDP Nachrichten von allen Geräten, die auf Port 514 senden. Über die beiden optionalen Parameter aus Tabelle 5.7 lässt sich das Verhalten dieser Quelle jedoch noch weiter steuern. Die Angabe von allgemeinen Parametern aus Tabelle 5.4 ist ebenfalls möglich.

Tabelle 5.7. Optionale Parameter für die Angabe einer *udp* Quelle in SYSLOG-NG.

Parameter	Beschreibung
localip	Schnittstellenadresse, an welche die *udp* Quelle gebunden werden soll.
localport	Port-Nummer, an welche die *udp* Quelle gebunden werden soll.

tcp

Die *tcp* Quelle ist direkt vergleichbar mit der *udp* Quelle. Es wird allerdings nicht das verbindungslose Protokoll UDP zur Nachrichtenübermittlung verwendet, sondern das verbindungsorientierte Protokoll TCP. Aus diesem Grund können auch noch drei weitere Parameter für *tcp* Quellen spezifiziert werden, die jeweils Angaben zu den Verbindungen machen. Tabelle 5.8 listet alle möglichen optionalen Parameter auf, die neben den allgemeinen Parametern aus Tabelle 5.4 angegeben werden können.

Tabelle 5.8. Optionale Parameter für die Angabe einer *tcp* Quelle in SYSLOG-NG.

Parameter	Beschreibung
localip	Schnittstellenadresse, an welche die *udp* Quelle gebunden werden soll.
localport	Port-Nummer, an welche die *udp* Quelle gebunden werden soll.
max-connections	Gibt die maximale Anzahl an TCP Verbindungen an, die von dieser Quelle parallel geöffnet werden dürfen.
keep-alive	Definiert, ob offene TCP Verbindungen bei einem Neustart von SYSLOG-NG offen gehalten werden sollen.
tcp-keep-alive	Über diesen Parameter kann definiert werden, ob SYSLOG-NG aktiv `tcp-keep-alive` [22] Pakete zur Aufrechterhaltung der TCP Verbindungen senden soll.

5.2.2 Ziel

Über die Angabe eines oder mehrerer Ziele kann bei SYSLOG-NG bestimmt werden, wohin ausgehende Nachrichten gesendet werden sollen. Prinzipiell sind wieder ähnliche Angaben möglich, wie bei den Quellen auch. Eine Angabe von *internal* ist verständlicherweise nicht möglich. Auch *sun-streams* können nicht als Ziel definiert werden. Es können allerdings noch die beiden Ziele *usertty* und *program* angegeben werden, die nicht als Quelle fungieren können. Für alle Ziele existieren die beiden in Tabelle 5.9 aufgelisteten allgemeinen Parameter, die unabhängig von allen anderen Parametern zusätzlich angegeben werden können.

unix-stream

Das *unix-stream* Ziel hat die gleiche Bedeutung wie auch die *unix-stream* Quelle und bezeichnet verbindungsorientierte Unix Socket Streams, wie sie vorrangig bei Linux Systemen zur SYSLOG Kommunikation verwendet werden. Der zugehörige Dateiname wird über einen notwendigen Parameter definiert. Zusätzlich können nur noch die allgemeinen Parameter aus Tabelle 5.9 angegeben werden.

Tabelle 5.9. Optionale globale Parameter, die zu jeder der acht möglichen Ziele in SYSLOG-NG angegeben werden können.

Parameter	Beschreibung
template	Über diesen Parameter können Formatvorlagen für die erstellten Nachrichten angegeben werden. Auf diese Weise lassen sich alle Meldungen zu einem Ziel auf dieselbe Art formatieren.
template_escape	Mit diesem Flag kann dafür gesorgt werden, dass alle einfachen und doppelten Anführungszeichen in den ausgehenden Nachrichten maskiert werden. Auf diese Weise können die Meldungen gegebenenfalls vom Empfänger besser interpretiert werden.

unix-dgram

Auch das *unix-dgram* Ziel beschreibt wie schon die *unix-dgram* Quelle Unix Sockets, die nicht mit Streams, sondern mit den verbindungslosen Datagrammen verknüpft sind. Der erforderliche Dateiname für diesen Socket wird über den ersten Parameter angegeben. Zusätzlich können noch die beiden allgemeinen Parameter aus Tabelle 5.9 angegeben werden.

file

Als Ziel für eine Log-Meldung wird häufig eine Datei im lokalen Dateisystem verwendet. Zu diesem Zweck existiert das *file* Ziel, das mit einer ganzen Reihen von optionalen Parametern versehen werden kann. Tabelle 5.10 liefert einen Überblick und eine kurze Beschreibung der möglichen Parameter. Die außergewöhnliche Flexibilität von SYSLOG-NG zeigt sich insbesondere auch am Ziel *file*, denn die Dateinamen können auch verschiedene Variablen enthalten wie beispielsweise Zeitangaben sowie Angaben zur Kritikalität oder Nachrichtenherkunft.

pipe

Über das *pipe* Ziel kann eine Pipe im Dateisystem angelegt werden, an welche die Log-Meldungen gesendet werden können. Die Pipe wird nicht von SYSLOG-NG angelegt, sondern muss bereits im Verzeichnisbaum existieren. Es lassen sich allerdings von SYSLOG-NG die Attribute der Datei über die in Tabelle 5.11 aufgelisteten Parameter justieren, die zusätzlich zu den allgemeinen Parametern spezifiziert werden können. Auf jeden Fall muss aber über den ersten Pflichtparameter der Dateiname definiert werden, an welchen die Pipe gebunden ist.

Tabelle 5.10. Optionale Parameter für die Angabe eines *file* Zieles in SYSLOG-NG.

Parameter	Beschreibung
log_fifo_size	Maximale Größe des Ausgangspuffers
fsync	Gibt an, ob die Datei nach jeder Schreiboperation synchronisiert werden soll.
sync_freq	Gibt die Anzahl der Nachrichten an, die in die Datei geschrieben werden können, bevor eine erneute Synchronisierung stattfinden muss.
remove_if_older	Definiert eine Zeitspanne, nach der bereits in die Datei geschriebene Nachrichten gelöscht werden.
owner	Benutzer ID (uid) der Datei.
group	Gruppen ID (gid) der Datei.
perm	Unix Dateiberechtigungen für die Logdatei
create_dirs	Regelt, ob nicht existierende Verzeichnisse bei Bedarf angelegt werden sollen.
dir_owner	Benutzer ID (uid) aller von SYSLOG-NG angelegten Verzeichisse.
dir_group	Gruppen ID (gid) aller von SYSLOG-NG angelegten Verzeichisse.
dir_perm	Unix Dateiberechtigungen aller von SYSLOG-NG angelegten Verzeichisse.
compress	Gibt an, ob der Inhalt der Datei mittels `zlib` [59] komprimiert werden soll.
encrypt	Aktiviert die Verschlüsselung der Logdatei.

Tabelle 5.11. Optionale Parameter für die Angabe eines *pipe* Zieles in SYSLOG-NG.

Parameter	Beschreibung
owner	Benutzer ID (uid) der Pipe
group	Gruppen ID (gid) der Pipe
perm	Unix Dateiberechtigungen für die Pipe

udp

Mit dem *udp* Ziel können Log-Meldungen an beliebige andere Geräte im Netzwerk weitergeleitet werden. Der einzige vorgeschriebene Parameter gibt die IP Adresse des Ziels an. Optional können aber neben den allgemeinen Parametern aus Tabelle 5.9 noch mehrere Angaben gemacht werden, die in Tabelle 5.12 aufgelistet sind.

tcp

Ebenso wie beim *udp* Ziel können beim *tcp* Ziel Log-Meldungen beliebig mittels des Protokolls TCP an entfernte Geräte geleitet werden. Mit Hilfe des Ziels *tcp* sind vor allem die Voraussetzungen für einen Sicherheitsmechanismus geschaffen, der allerdings in SYSLOG-NG nicht explizit vorhanden ist. In

Tabelle 5.12. Optionale Parameter für die Angabe eines *udp* Zieles in SYSLOG-NG.

Parameter	Beschreibung
destport	Alternativ kann über den `destport` Parameter ein anderer UDP Port als 514 definiert werden.
localip	Schnittstellenadresse, an welche das *udp* Ziel gebunden werden soll.
localport	Port-Nummer, an welche das *udp* Ziel gebunden werden soll.
spoof_source	Bestimmt, ob SYSLOG-NG bei Nachrichten, die über das Netzwerk eingegangen sind, die Quelladresse der von SYSLOG-NG generierten Pakete auf die Quelladresse des Originalpaketes gesetzt werden soll. In diesem Fall täuscht SYSLOG-NG einen falschen Absender (den ursprünglichen Sender) vor.

Tabelle 5.13 sind die zum *udp* Ziel ähnlichen optionalen Parameter aufgelistet, die sich durch die allgemeinen Parameter aus Tabelle 5.9 ergänzen lassen. Die Angabe einer Zieladresse ist jedoch verpflichtend.

Tabelle 5.13. Optionale Parameter für die Angabe eines *tcp* Zieles in SYSLOG-NG.

Parameter	Beschreibung
destport	Zu verwendender Zielport.
localip	Schnittstellenadresse, an welche das *udp* Ziel gebunden werden soll.
localport	Port-Nummer, an welche das *udp* Ziel gebunden werden soll.
tcp-keep-alive	Über diesen Parameter kann definiert werden, ob SYSLOG-NG aktiv `tcp-keep-alive` [22] Pakete zur Aufrechterhaltung der TCP Verbindungen senden soll.

usertty

Über das *usertty* Ziel können Log-Meldungen an interaktive Konsolenfenster im System weitergeleitet werden. Der einzige vorgeschriebene Parameter gibt den Benutzer an, der diese Nachrichten in seinen Konsolenfenstern erhalten soll. Die allgemeinen Parameter aus Tabelle 5.9 können ebenfalls spezifiziert werden.

program

Ebenfalls nicht als Quelle steht das *program* Ziel zur Verfügung. Eine eingehende Nachricht, die an ein *program* Ziel geleitet werden soll, initiiert den Aufruf

des über den einzigen vorgeschriebenen Parameter angegebenen Programms auf dem SYSLOG-NG Gerät. Das Programm erhält die Nachricht über seinen Standard-Eingabekanal `STDIN`. Damit nicht eine überproportional große Anzahl an Programmen gestartet werden muss, bleibt jedes Programm nach einmaliger Ausführung bis zur nächsten Neuinitialisierung von SYSLOG-NG aktiv.

5.2.3 Filter

Jede eingehende Nachricht kann durch einen oder mehrere Filter geleitet werden, um die Nachrichten noch genauer auswählen zu können. Bei jedem spezifizierten Filter kann auf mehrere Filterfunktionen zurückgegriffen werden, die mit den üblichen Operatoren `and`, `or` und `not` verknüpft werden können. Tabelle 5.14 listet die sieben vordefinierten Filterfunktion auf.

Tabelle 5.14. Vordefinierte Filterfunktionen von SYSLOG-NG.

Funktion	Beschreibung
level	Sucht nach Nachrichten, welche die angegebene Kritikalität besitzen.
facility	Sucht nach Nachrichten mit dem angegebenen Facility-Wert.
host	Sucht nach Nachrichten, bei denen der Hostname dem angegebenen Regulären Ausdruck [75] genügt.
netmask	Sucht Nachrichten, der Absender im angegebenen Subnetz liegt.
program	Sucht nach Nachrichten, die vom angegebenen Programm erzeugt wurden.
match	Sucht allgemein nach Nachrichten, welche dem angegebenen Regulären Ausdruck genügen.
filter	Wendet einen anderen bereits definierten Filter an.

5.2.4 Protokollpfad

Die Hauptangaben für das Verhalten von SYSLOG-NG finden sich in den einzelnen Protokollpfaden. In diesen werden die verschiedenen definierten Quellen, Ziele und Filter miteinander verknüpft. Jede eingehende Nachricht wird von SYSLOG-NG an alle Protokollpfade zur weiteren Bearbeitung übergeben. Für eine schematische Übersicht der internen Zusammenhänge von SYSLOG-NG soll noch einmal auf Abbildung 5.2 verwiesen werden. Daraus lässt sich ablesen, dass ein einzelner Protokollpfad aus drei einzelnen Elementen besteht:

1. Das erste Element beschreibt die Quellen, von denen die Nachricht empfangen werden kann. Es kann mehr als eine Quelle definiert werden, deren Nachrichten alle gleichermaßen an die Filter weitergeleitet werden.

2. Nachdem eine Nachricht von einer der angegebenen Quellen empfangen worden ist, wird sie durch die angegebenen Filter geleitet. Nur Nachrichten, die allen definierten Filtern genügen, können in die Nachrichtenausgabe gelangen.
3. Das letzte Element besteht aus einer Auflistung von einem oder mehreren Zielen, an welche die Nachrichten schließlich geleitet werden sollen.

Die Bearbeitung von Log-Meldungen durch die verschiedenen Protokollpfade von SYSLOG-NG kann durch unterschiedliche Flags beeinflusst werden, die sich sowohl auf die Quellen, die Filter oder die Ziele auswirken können.

catchall

Mit der Angabe des *catchall* Flags werden einem Protokollpfad sämtliche eingehenden Nachrichten unabhängig von ihrer Quelle zugeführt. Gegebenenfalls spezifizierte Quellen werden demnach nicht berücksichtigt.

final

Ein Protokollpfad mit gesetztem *final* Flag sorgt dafür, dass alle Nachrichten, welche die angegebenen Filter erfolgreich passieren, an keinen nachfolgenden Protokollpfad mehr weitergereicht werden. Die Bearbeitung der entsprechenden Nachrichten endet im jeweiligen Protokollpfad.

fallback

Ist bei einem Protokollpfad das *fallback* Flag gesetzt, so erhält dieser Protokollpfad eine eingehende Nachricht nur dann zur Bearbeitung, wenn sich kein anderer Protokollpfad ohne gesetzten *fallback* Flag findet.

flow-control

Mit der Angabe des *flow-control* Flags kann verhindert werden, dass SYSLOG-NG die spezifizierten Ziele mit Log-Meldungen überflutet. Es wird dann darauf geachtet, dass erst wieder Nachrichten eingelesen werden, wenn der Ausgangspuffer Platz für weitere Meldungen besitzt.

6

Intelligent Platform Management Interface

Die Intelligent Platform Management Interface (IPMI) [95] ist eine von mehreren Herstellern in gemeinschaftlicher Zusammenarbeit erstellte Definition für ein Out-of-Band Managementsystem, das auf einer separaten Hardware basiert, die auch beim Ausfall des zu verwaltenden Gerätes noch zur Verfügung steht. Der IPMI Mechanismus ist kein aktueller Standard, wie SNMP oder die Internetprotokolle es sind; vielmehr haben sich einige Hersteller zusammengeschlossen, um eine vereinheitlichte Definition dieser Out-of-Band Management Schnittstelle zu erarbeiten. Zwar ist IPMI nicht als Netzwerkmanagementsystem im klassischen Sinne zu verstehen, da es sich in großen Teilen auf einem sehr niedrigen und Hardware-nahen Niveau befindet. Dennoch sind die Funktionen und Möglichkeiten des IPMI eine wertvolle Bereicherung herkömmlicher Netzwerkmanagement-Software. Aus diesem Grund soll an dieser Stelle auch eine nähere Betrachtung des IPMI Rahmenwerks erfolgen.

Durch die Beteiligung mehrerer Hersteller an diesem Rahmenwerk ist es nicht verwunderlich, dass IPMI äußerst flexibel ist und die verschiedensten Kommunikationswege, Hardwarekomponenten und Nachrichtenformate unterstützt. Neben den hauptverantwortlichen Intel [91], Hewlett-Packard [83], NEC [137] und Dell [58] ist noch eine ganze Reihe weiterer Hersteller von Hardware und Software beteiligt – sei es durch die Mitwirkung bei der Entwicklung der Spezifikation oder durch die Unterstützung und Implementierung in ihren Produkten. Im Anhang B findet sich eine Liste der entsprechenden Hersteller vom Stand August 2005. Die exakten Spezifikationen aller Details sind vermutlich nur für Hersteller und Implementierer des Mechanismus von Interesse und können in der offiziellen Spezifikation [95] nachgelesen werden. Zum grundlegenden Verständnis soll daher in diesem Kapitel lediglich eine Übersicht über den Mechanismus geliefert werden. Dazu zählen in erster Linie der zentrale Prozessor sowie die möglichen Kommunikationswege, Nachrichtenformate und Hardware-Spezifikationen.

6.1 Hardware

Zur Hardware in einem IPMI Gerät zählen verschiedene Systemkomponenten, die auch mehrere logische Aufgaben gleichzeitig erfüllen können. Kernkomponente ist der Baseboard Management Controller (BMC), der durch den Intelligent Platform Management Bus (IPMB) mit anderen IPM Geräten verbunden sein kann.

6.1.1 BMC

Die zentrale Rolle in der IPMI Architektur übernimmt der Baseboard Management Controller (BMC), der als Vermittler zwischen der Hardware, den verschiedenen Bus-Systemen und der Management-Software steht. Der BMC übt ferner die Überwachung der verschiedenen angeschlossenen Hardware-Komponenten aus, protokolliert die unterschiedlichsten Ereignisse innerhalb des Systems und kann die Wiederherstellung der Arbeitsfähigkeit eines nicht mehr reagierenden Systems einleiten. Insbesondere aus letztem Grund ist es auch wichtig, dass der BMC in einer separaten Hardware realisiert ist, die vom eigentlichen System unabhängig arbeiten kann. Gleichzeitig ist aber eine enge Verzahnung des BMC mit den übrigen Komponenten des System zwingend erforderlich, damit er seine vielfältigen Aufgaben erfüllen kann. Damit eine reibungslose Kommunikation zwischen dem BMC und den diversen Teilkomponenten des Systems möglich ist, wird in der IPMI Architektur schließlich noch ein gemeinsames Nachrichtenformat spezifiziert, das gekapselt über die verschiedenen Bus-Systeme und Protokolle übermittelt werden kann.

Damit ein IPMI-fähiges Gerät Managementfunktionalitäten selbst im ausgeschalteten Zustand anbieten kann, muss der BMC über eine separate Spannungsversorgung und einen separaten Kommunikationskanal verfügen. Um die technischen Anforderungen an heutige Systeme nicht über Gebühr zu beanspruchen, können zur Implementierung dieser Anforderungen bereits bestehende Mechanismen und Architekturen verwendet werden.

Sensoren und Sensorgeräte

Die verschiedenen Sensoren in einem IPMI System sind im Normalfall in externer Hardware untergebracht, obwohl immer mehr Geräte über eingebaute Messwertgeber verfügen. Die Anbindung der Sensoren an IPMI Geräte erfolgt über ein Sensorgerät, welches auch für die Erstellung von Sensor Data Record (SDR) Nachrichten sowie System Event Log (SEL) Nachrichten beim Eintreffen bestimmter Ereignisse zuständig ist. Die SDR Nachrichten werden zur weiteren Speicherung an eine SDR Datenbank gesendet, während die SEL Nachrichten an den zentralen Ereignis-Logger geschickt werden.

SDR Datenbank

Die SDR Datenbank enthält eine vollständige Liste aller am IPMB Bus vorhandenen Sensoren und speichert diese vorzugsweise in einem nichtflüchtigen Speicher. Die Hauptaufgabe der SDR Datenbank besteht darin, eine vollständige Liste aller angeschlossenen Sensoren zu liefern, damit die Management-Software den Ausfall eines Sensors vom Fehlen des Sensors an einem freien Anschluss unterscheiden kann. Anhand der Sensorenliste kann die Management-Software gezielt die einzelnen Komponenten abfragen und das Ausbleiben einer Antwort unmittelbar als einen Fehlerzustand interpretieren.

SEL Datenbank

Die SEL Datenbank empfängt die SEL Nachrichten von allen Sensoren und Geräten des IPMB Busses und speichert sie im System-Log ab. Der BMC ist außerdem in der Lage, das System-Log abzufragen oder zu löschen.

SEL Filter

Ein Platform Event Filtering (PEF) übernimmt im Wesentlichen die Aufgabe eines Filters, der eingehende SEL Nachrichten nach Mustern durchsucht und bei Übereinstimmung entsprechende Aktionen auslöst. Mit Hilfe eines PEF kann vergleichsweise einfach auf vordefinierte Ereignisse reagiert werden. Als Ergebnis auf ein erkanntes Ereignis wird ein Alarm an die vorab spezifizierten Ziele gesendet.

Chassis-Gerät

Das Chassis-Gerät bietet vorrangig eine Schnittstelle zu systemübergreifenden Komponenten wie der Spannungsversorgung. Über das Chassis-Gerät kann beispielsweise das System gestartet, heruntergefahren oder auch neu initialisiert werden.

Informationsverzeichnis der Hardwareeinheiten

Jede als Field Replacable Unit (FRU) bezeichnete Hardwareeinheit verfügt über ein spezielles logisches Gerät, welches das Verzeichnis über alle Informationen zu dieser Hardwareeinheit verwaltet. In diesem Verzeichnis sind beispielsweise Informationen wie die Seriennummer oder die Typennummer der Hardware enthalten.

6.1.2 IPM Gerät

Neben dem BMC können noch andere IPM Geräte zur selben IPMI Komponente hinzugezählt werden. Der Unterschied zum BMC besteht in einer eingeschränkten Funktionalität. Beispielsweise können einfache IPM Geräte nur Nachrichten über Ereignisse versenden, aber nur der BMC kann auch als Empfänger für diese Nachrichten fungieren.

6.1.3 Spannungsversorgung

In einem modernen Rechnersystem wird die Spannungsversorgung der einzelnen Teilsysteme vom Advanced Configuration and Power Interface (ACPI) [45] übernommen. ACPI ist in der Lage, jede Komponente zustandsabhängig mit der notwendigen Betriebsspannung zu versorgen, um so einzelne Komponenten nach Bedarf einschalten und ausschalten zu können. Eng damit gekoppelt sind die Netzteile, die zu diesem Zweck Betriebsspannungen dynamisch liefern können. Mechanismen wie beispielsweise „Wake-on-LAN“ nutzen die Besonderheiten von ACPI aus, in dem ein Teil der Netzwerkschnittstelle selbst bei ausgeschaltetem Gerät aktiv ist und den Netzwerkverkehr nach Mustern durchsucht. Durch das Empfangen einer „Magischen Zeichenfolge“ initiiert die Netzwerkschnittstelle einen Start des Systems. Dazu wird über den ACPI Mechanismus ein Impuls an das Netzteil gesendet, welches dann die Stromzufuhr für das Hauptsystem herstellt.

IPMI arbeitet auf eine ähnliche Art und Weise wie der „Wake-on-LAN“ Mechanismus. Auch der BMC erhält seine Spannungsversorgung über ACPI selbst dann noch, wenn sich das zu überwachende System im ausgeschalteten Zustand befindet. Über IPMI kann daher eine zu „Wake-on-LAN“ ähnliche Funktionalität abgebildet werden, bei der das Hauptsystem durch einen Befehl an den BMC gestartet wird. Darüber hinaus kann der BMC das System auch herunterfahren, was beispielsweise beim Ausfall eines unverzichtbaren Kühlelementes vor einem kapitalen Hardware-Schaden schützen kann. Zusätzlich sind häufig mehrere Zwischenzustände zwischen ausgeschaltet und vollständig aktiviert möglich. Die ACPI Spezifikation und auch IPMI erlauben im Bedarfsfall eine sukzessive Drosselung der Leistung und der Funktionalität eines verwalteten Gerätes. Anstatt das System vollständig herunterzufahren kann der BMC bei einem Teilausfall der Kühlung beispielsweise den Prozessortakt etwas herunterfahren, wodurch die Wärmeentwicklung signifikant gedrosselt werden kann.

6.2 Kommunikationskanäle

Damit ein BMC selbst bei deaktiviertem Hauptsystem noch funktionieren kann, reicht eine separate Spannungsversorgung allein nicht aus. Entscheidend ist die zusätzliche Errichtung eines Kommunikationskanals, der unter diesen

Bedingungen noch funktionstüchtig ist. Ein Beispiel stellt wieder der bereits weiter oben erwähnte „Wake-on-LAN“ Mechanismus dar. Trotz ausgeschaltetem System ist ein Teil der Netzwerkschnittstelle in der Lage, eingehende Pakete nach einem vorher definierten Muster zu untersuchen und bei Identifizierung der „Magischen Zeichenfolge“ eine Aktion auszulösen.

Das Beispiel „Wake-on-LAN“ zeigt eindrucksvoll, dass der separate Kommunikationskanal nicht zwingend eine IPMI eigene Schnittstelle sein muss; es genügt eine Zugriffsmöglichkeit auf vorhandene Strukturen, die auch im ausgeschalteten Zustand noch gegeben ist. IPMI unterstützt zur Erfüllung seiner Aufgaben die verschiedensten Kommunikationskanäle mit den unterschiedlichsten Komplexitätsstufen. Der direkt daraus folgende Vorteil liegt in dem flexiblen Out-of-Band Management der verwalteten Systeme. Durch die Möglichkeit einer unabhängigen Anbindung kann die IPMI Architektur nicht nur den Ausfall der Hauptkomponente kompensieren, sondern auch die Unterbrechung des Datenkommunikationsweges zum Gerät.

Im Folgenden sollen die bei IPMI möglichen Kommunikationskanäle vorgestellt und näher beschrieben werden.

6.2.1 IPMB

Der Intelligent Platform Management Bus (IPMB) [93] erweitert den von Philips [156] entwickelten und eingeführten Inter Integrated Circuit (I^2C) Bus [192] um eine höhere Kommunikationsebene, die einen Datenaustausch zwischen beliebigen I^2C Geräten und dem IPMI ermöglicht. Der bewährte I^2C Mechanismus erlaubt eine einfache Kopplung von mehreren intelligenten oder nicht-intelligenten Systemkomponenten. Die Spezifikation ist vergleichsweise einfach und unterstützt auch einfache Sensortypen wie Temperatursensoren oder Spannungsmesser. Über den I^2C Bus können physikalische Messwerte dieser Art an ein intelligenteres System übergeben werden, welches diese Größen überwachen kann und im Bedarfsfall auch geeignete Aktionen auslösen kann. Die oben beschriebenen Sensoren werden in die Kategorie der „nicht-intelligenten“ I^2C Geräte einsortiert. Sie besitzen keinerlei Intelligenz und liefern lediglich ihre Messwerte an intelligentere Systeme. Daneben können aber auch andere intelligentere Komponente wie Mikroprozessoren an den I^2C Bus angeschlossen werden, die untereinander einfache Daten austauschen können. IPMB setzt auf den bestehenden I^2C Bus auf und erlaubt den Anschluss weiterer intelligenter Geräte. Ein solches intelligentes Gerät kann beispielsweise ein Mikroprozessor sein, der nicht nur einzelne Messgrößen überwacht, sondern bei Überschreiten oder Unterschreiten eines Grenzwertes auch automatisch einen Alarm triggern kann. Über den IPMB Kommunikationskanal kann sogar der Austausch oder eine Aktualisierung der Firmware des intelligenten I^2C Gerätes durchgeführt werden. Andere Typen von intelligenten I^2C Geräten bilden Konnektoren zwischen IPMB und anderen Bus-Systemen. Diese Brücken können beispielsweise IPMB mit dem Intelligent Chassis Management Bus (ICMB) verbinden. Über diesen Weg wird auch das Intelligent

Platform Management Interface an den IPMB Kommunikationskanal angebunden.

6.2.2 ICMB

Der Intelligent Chassis Management Bus (ICMB) [94] erweitert den ausschließlich innerhalb eines einzelnen Gerätes operierenden IPMB um die Möglichkeit zur Kommunikation mit anderen Geräten. Abbildung 6.1 veranschaulicht den Zusammenhang zwischen dem Intelligent Chassis Management Bus und dem Intelligent Platform Management Bus.

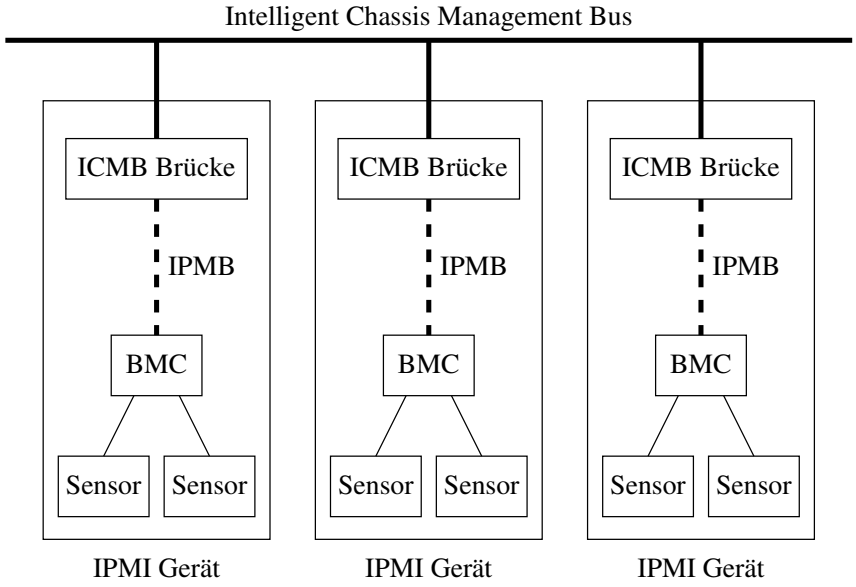

Abb. 6.1. Schematische Darstellung des Zusammenhangs zwischen IPMB und ICMB. Die IPMI-fähigen Geräte sind jeweils mit ICMB Brücken an den ICMB angeschlossen, während der IPMB nur innerhalb der einzelnen Geräte verläuft.

Verschiedene Geräte, die über einen IPMB Bus verfügen, werden über ICMB Brücken miteinander verbunden. Jede der ICMB Brücken besitzt jeweils eine Schnittstelle im ICMB Bus und im IPMB Bus, zu dem sie gehört. Daraus folgt, dass die Kommunikation über den Bus ICMB immer von einer ICMB Brücke zu einer anderen verläuft. Aus diesem Grund kommt den Brückengeräten auch eine besondere Bedeutung zu. Die Hauptaufgabe der ICMB Brücken besteht im Weiterleiten und Transformieren von Nachrichten zwischen den beiden Bus-Systemen. Erhält eine ICMB Brücke auf der IPMB Seite eine Nachricht für eine Komponente in einem anderen Gerät, so leitet die Brücke diese Nachricht an den ICMB Bus weiter. Empfängt die Brücke über den ICMB Bus eine

Nachricht für eine lokale Komponente, so leitet sie die Nachricht über ihren IPMB Bus an diese Komponente weiter.

Man unterscheidet zwei verschiedene Arten von ICMB Brücken. Die Management ICMB Brücken befinden sich im Normalfall in einem verwalteten Gerät und verfügen daher auch über Funktionalitäten zur Verwaltung und Überwachung anderer Brücken. Dazu zählt auch das aktive Erkennen aller am ICMB Bus angeschlossenen Brücken sowie das Versenden oder Verarbeiten von Ereignismeldungen. Zu den Geräten mit einer aktiven Management Brücke zählen beispielsweise die Brücken auf der Hauptplatine eines Systems.

Die zweite Art von ICMB Brücken übt lediglich eine passive Rolle als Paketvermittler aus. Diese Peripherie Brücken befinden sich häufig in Systemkomponenten, die nicht über eine aktive Überwachung und Verwaltung verfügen. Beispielsweise wird die ICMB Brücke in einem Netzteil häufig in Form einer Peripherie Brücke implementiert, die wiederum von der Management Brücke überwacht und verwaltet wird.

Adressverwaltung

Damit eine geräteübergreifende Kommunikation von IPMB Komponenten zuverlässig funktionieren kann, müssen die Komponenten und Bus-Systeme eine eindeutige Adresse erhalten, über die sie identifiziert werden können. Vor allem für die ICMB Brücken ist eine Adressverwaltung wichtig, da sie an zwei verschiedene Bus-Systeme angeschlossen sind, die jeweils ihre eigenen Adressräume besitzen.

IPMB Adressen. Die Spezifikation des IPMB Busses ist eng mit dem I^2C Mechanismus verknüpft. Da sich im IPMB Bus sowohl intelligente als auch nicht-intelligente I^2C Geräte befinden können, verwendet der IPMB Bus das Adressierungsschema von I^2C in unveränderter Form. Dadurch wird sichergestellt, dass auch die nicht-intelligenten Geräte ohne IPMB Unterstützung ohne Beeinträchtigung denselben Bus benutzen können. IPMB Geräte können folgerichtig nur die im I^2C Mechanismus spezifizierten 7-Bit oder 10-Bit Adressen verwenden.

I^2C Geräte beginnen eine Nachrichtensendung auf dem Bus immer mit einem Adress-Byte, dessen erste sieben Bit der Adresse der angesprochenen Komponente entsprechen. Das achte Bit dient der Angabe der Kommunikationsrichtung. Dieses achte Bit ist bei der IPMB Spezifikation immer auf Null gesetzt, da lediglich Schreibzugriffe, aber keine Lesezugriffe auf den Bus definiert sind. Die zur 7-Bit Adressierung voll kompatible 10-Bit Adressierung verwendet eine reservierte Adressangabe in diesen ersten sieben Bit, deren erste fünf Bit auf den Wert 11110 gesetzt sind und die folgenden beiden Bit zusammen mit dem nächsten Oktett eine Adresse mit zehn Bit Länge ergeben. Mit diesem Adressierungsschema ist die Forderung nach eindeutigen Adressen innerhalb eines IPMB Busses leicht zu erfüllen. Gleichzeitig ist allerdings auch klar, dass IPMB Adressen nur innerhalb eines einzelnen Bus-Systems

eindeutig sein müssen und können; zur Adressierung von Komponenten in einem anderen IPMB Bus müssen daher zusätzliche Adressangaben gemacht werden.

ICMB Adressen. ICMB verwendet für die Adressierung verschiedener Geräte und Brücken ein von IPMB unabhängiges Schema. ICMB Adressen bestehen aus 16-Bit Werten auf und müssen genauso wie die IPMB Adressen innerhalb desselben Busses eindeutig sein. Analog zur IPMB Spezifikation können unabhängige Bus-Systeme dieselben Adressen verwenden, solange sie innerhalb eines Busses eindeutig bleiben. ICMB ist allerdings im Gegensatz zu IPMB ein dynamisches System, bei dem sich die am Bus angeschlossenen Geräte ständig ändern können. Dies kann sowohl durch das manuelle Hinzufügen oder Entfernen von ICMB Komponenten begründet sein, sowie auch durch das Einschalten oder Ausschalten von Systemen, was dem Hinzufügen oder Entfernen der Komponenten entspricht. Die Forderung nach eindeutigen Adressen in einem dynamischen Bus-System lässt sich aber nur mit einer dynamischen Adressvergabe widerspruchsfrei umsetzen. Als Folge daraus sind in der ICMB Spezifikation auch die Mechanismen festgelegt, mit denen die Adressen im Bus vergeben werden, und wie die einzelnen Geräte sich über die anderen angeschlossenen Systeme desselben Bus-Systems informieren können. Wird ein Gerät an einen ICMB Bus angeschlossen, so hat dies keinerlei Auswirkungen auf die Operationalität der ICMB Brücke in Bezug auf den internen IPMB Bus. Damit sich das Gerät aber erfolgreich in den ICMB Bus integrieren kann, muss es zunächst eine gültige und eindeutige ICMB Adresse erhalten. Aus diesem Grund befindet sich ein neues Gerät am ICMB Bus anfänglich im Statuszustand *Deaktiviert*. Abbildung 6.2 veranschaulicht die vier Statuszustände und deren mögliche Übergangsbedingungen.

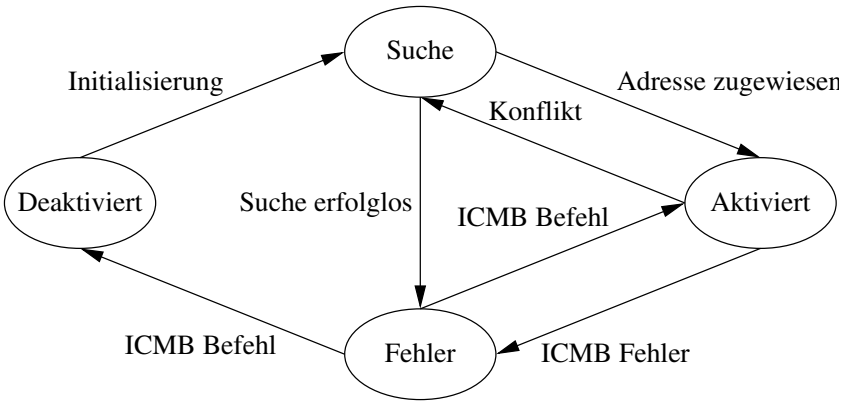

Abb. 6.2. Die vier möglichen Werte des Zustandsautomaten eines ICMB Gerätes in Bezug auf die Adresszuweisung. Neue Geräte starten im Zustand *Deaktiviert* und suchen den Zustand *Aktiviert* zu erreichen.

Nach der Initialisierung eines ICMB Gerätes verfügt es zunächst nicht über eine gültige Adresse. Im nichtflüchtigen Speicher der Brücke kann jedoch die Adresse des vorherigen Betriebs gespeichert sein. Die ICMB Brücke versucht nun, eine eindeutige Adresse im Bus-System zu ermitteln und für sich zu reservieren. Dazu beginnt sie mit der letzten gültigen Adresse aus dem nichtflüchtigen Speicher und sendet eine *GetICMBAddress* Nachricht an diese Adresse. Ein Adresskonflikt besteht nur dann, wenn ein anderes Gerät auf diese Nachricht antwortet, da dann die gewünschte Adresse bereits vergeben ist. Erhält die Brücke keine Antwort, so übernimmt sie die zugehörige ICMB Adresse und geht in den Zustand *Aktiv* über. In diesem Zustand ist die Brücke voll funktionsfähig und bearbeitet eingehende Anfragen sowohl aus dem ICMB als auch aus dem IPMB Bus. Erhält die Brücke jedoch eine Antwort, so liegt ein Adresskonflikt vor und der Vorgang wird bis zum Ausbleiben einer Antwort mit anderen Adressen wiederholt. So wird sichergestellt, dass die ICMB Adressen im Bus-System eindeutig bleiben.

Unter ungünstigen Bedingungen können trotz der Vorkehrungen des Protokolls dennoch Adresskonflikte auftreten. Um auch in diesem Fall die Operationalität des ICMB Busses aufrechterhalten zu können, überprüfen die ICMB Brücken alle eingehenden Nachrichten auf mögliche Adresskonflikte. Insbesondere eingehende Nachrichten, bei denen die Absender-Adresse mit der eigenen Adresse übereinstimmt, deuten auf einen solchen Adresskonflikt hin. In diesem Fall geht die Brücke direkt in den Zustand *Suche* über und beginnt mit dem Versenden von *GetICMBAddress* Nachrichten. Einzige Ausnahme bilden Nachrichten, bei denen nicht nur der Absender sondern auch der Empfänger die eigene Adresse besitzt. Diese Nachrichten werden von neu initialisierten anderen Geräten am ICMB Bus gesendet, um freie Adressen ausfindig zu machen. Empfängt eine ICMB Brücke eine derartige Nachricht, so kann sie im Zustand *Aktiv* verbleiben, muss jedoch eine Antwort an den Absender schicken – also an die eigene Adresse.

Auf Grund der dynamischen Adressen und der Tatsache, dass ICMB Geräte während des laufenden Betriebs ihre Adresse wechseln können, müssen die Managementsysteme ständig den ICMB Bus nach allen Geräten abfragen. Auf diese Weise können Managementsysteme eine eigene Repräsentation des Busses aufbauen. Die Managementstationen senden zu diesem Zweck Nachrichten des Typs *PrepareForDiscovery* an alle Geräte, was zur Folge hat, dass die nachfolgende *GetICMBAddress* Nachricht von allen Geräten mit Angabe der eigenen Adresse beantwortet wird. Anschließend beginnen alle Geräte nach einem zufälligen Zeitmuster ihre Antworten zu senden. Bei einer kleinen Anzahl von Geräten sind Nachrichtenkollisionen auf dem ICMB Bus eher selten und der Mechanismus ist vergleichsweise effektiv. Sicherheit über den Erhalt aller Nachrichten verschafft sich das Managementsystem jedoch dadurch, dass es die Broadcast Nachrichten mindestens viermal wiederholt.

ICMB Brücken Befehle

Da sich die ICMB Brücken zwischen dem ICMB Bus-System und einem IPMB Bus-System befinden, existieren verschiedene Arten von Befehlen, die entweder von den ICMB oder den IPMB Geräten stammen. Entsprechend unterschiedlich sind auch die möglichen Befehlssätze.

Managementbefehle. Die Managementbefehle kontrollieren ganz allgemein das Verhalten der ICMB Brücken. Die Befehle können sowohl über den IPMB als auch über den ICMB Bus empfangen werden. Aufgabe der Befehle sind Informationenabfragen zum Zustandsautomaten oder der ICMB Adresse der Brücke sowie Statistiken oder Detailangaben zu den unterstützten Fähigkeiten der Brücke und der eindeutigen Identifikationsnummer des Hardware Anschlusses der Brücke.

Gerätebefehle. Zur Verwaltung einer Liste aller an den Bus angeschlossenen ICMB Geräte und deren Fähigkeiten kann sich ein Managementsystem der Gerätebefehle bedienen. Zu den Gerätebefehlen gehört auch der bereits weiter oben angesprochene *PrepareForDiscovery* Befehl.

Weiterleitungsbefehle. Mit Hilfe der Weiterleitungsbefehle können Nachrichten vom IPMB Bus zum ICMB Bus und umgekehrt transportiert werden. Die Weiterleitungsbefehle realisieren somit die Hauptaufgabe der ICMB Brücken.

Ereignisbefehle. Um das Auftreten eines Fehlers oder besonderen Ereignisses an das Managementsystem weiterzuleiten, kann eine IPMB Komponente mit einem Weiterleitungsbefehl direkt an den Empfänger senden. Die Weiterleitungsbefehle stellen eine Punkt-zu-Punkt Nachrichtenvermittlung dar, so dass ein Broadcast des Ereignisses an mehrere oder alle Komponenten in anderen Geräten außerhalb desselben IPMB Busses nicht möglich ist. Zur Umgehung dieses Nachteils kann die IPMB Komponente auch einen Ereignisbefehl an die ICMB Brücke senden, da diese auch einen Broadcast unterstützen. Außerdem bleiben Ereignisbefehle unbeantwortet, was bei einem Broadcast von großer Bedeutung sein kann.

6.2.3 System Schnittstelle

Die System Schnittstelle von IPMI stellt eine direkte Verbindung des BMC mit dem Kern-Bus des Hauptsystems dar. Damit eine Herstellerneutralität und Software-Unabhängigkeit erreicht wird, existiert die System Schnittstelle in drei verschiedenen Ausprägungen, die einander sehr ähnlich sind. Alle drei Varianten sind Input/Output (I/O) gesteuert und unterliegen verschiedenen Voraussetzungen für die Signalsynchronisierung.

Keyboard Controller Style

Die Keyboard Controller Style (KCS) Schnittstelle basiert auf der Spezifikation für den Intel 8742 Universal Peripheral Interface Microcontroller [92]. Durch die hohe Verbreitung des 8742 Mikroprozessors und die damit verbundene weitläufige Akzeptanz der KCS Spezifikation als Tastatur Schnittstelle findet sich eine entsprechende Unterstützung in vielen Systemen, und ein BMC kann sich dieser Schnittstelle leicht bedienen. Die Daten werden Oktett-weise über die KCS Schnittstelle übertragen.

System Management Interface Chip

Wird zur Implementierung des BMC ein Mikroprozessor ohne Unterstützung der KCS Spezifikation verwendet, so kann alternativ die Schnittstelle eines System Management Interface Chip (SMIC) verwendet werden. Zur Realisierung genügt ein einfacher Application-Specific Integrated Circuit (ASIC) oder ein Field-Programmable Gate Array (FPGA), welche die I/O gestützte Kommunikation regeln. Auch bei SMIC werden die Daten Oktett-weise über die Schnittstelle übertragen.

Block Transfer

Für eine performantere Übermittlung von Daten kann die Block Transfer (BT) Schnittstelle zum Einsatz kommen. Die Daten werden dabei nicht Oktettweise, sondern blockweise versendet. Wie beim SMIC genügt zur Realisierung der BT Schnittstelle ein einfacher ASIC oder FPGA.

6.2.4 Serielle Schnittstelle

Nahezu alle modernen Systeme verfügen über eine Serielle Schnittstelle, über die auf einfache Weise Daten übertragen werden können. Eine Anbindung an andere Systeme kann sowohl statisch über eine feste Verdrahtung als auch dynamisch über ein Modem realisiert werden. Der Multiplexer, der zwischen der Seriellen Schnittstelle des Systems, dem externen seriellen Anschluss und dem BMC liegt, vermittelt je nach gewünschter Kommunikationsbeziehung die Signale zwischen den beiden aktiven Gesprächspartnern. Im Normalzustand kann das System direkt auf die Serielle Schnittstelle zugreifen, und es besteht eine direkte Verbindung über den Multiplexer. Erbittet der BMC den Zugriff auf den externen seriellen Anschluss, um eine Kommunikation mit einem externen Managementsystem aufzunehmen, so trennt der Multiplexer das lokale System ab und vebindet den BMC mit dem externen Anschluss. Bei der weiter unten beschriebenen SOL Kommunikation kann der Multiplexer sogar den externen Anschluss trennen und den BMC direkt mit der Seriellen Schnittstelle des Systems verbinden.

Für eine maximale Flexibilität unterstützt IPMI drei verschiedene Kommunikationsprotokolle über die Serielle Schnittstelle. Neben dem IPMI proprietären Basis Modus existiert zusätzlich noch ein allgemeiner Modus, welcher den weit verbreiteten Standard Point-to-Point Protocol (PPP) [195] unterstützt sowie einen Terminal Modus, der einer Verbindung mit einer Textkonsole ähnelt.

Basis Modus

Der Basis Modus der Seriellen Schnittstelle verwendet ein IPMI eigenes Kommunikationsprotokoll, das sich besonders durch seine Performanz auszeichnet. Anstatt die IPMI Nachrichten aufwändig in mehrere höhere Protokolle einzubetten, wie es beispielsweise bei Verwendung des PPP Modus notwendig ist, werden die Nachrichtenpakete direkt Oktett-weise über die Serielle Schnittstelle versendet. Es werden lediglich vier Oktette in einer besonderen Weise behandelt, damit eine rudimentäre Kommunikationssteuerung möglich ist. Dazu gehört die Möglichkeit für den Sender, Beginn und Ende einer Datenübertragung ankündigen zu können sowie die Möglichkeit für den Empfänger, die Empfangsbereitschaft bestätigen zu können. Damit diese drei besonderen Oktette auch als normale Daten verwendet werden können, müssen sie bei der Übertragung „maskiert“ werden, wozu ein viertes spezielles Oktett zur Verfügung steht. Der Overhead einer im Basis Modus über die Serielle Schnittstelle übertragenen IPMI Nachricht ist demnach minimal und somit auch die Performanz optimal.

Zur Verdeutlichung des Basis Modus sind in Tabelle 6.1 die besonderen Zeichen mit ihrer Bedeutung in Ersatzzeichenfolge aufgelistet. Taucht eines der Zeichen im Datenstrom selbst auf, so wird es gemäß der Tabelle durch die Ersatzzeichenfolge ausgetauscht. Die Angaben in der Tabelle sind in hexadezimaler Form gemacht.

Tabelle 6.1. Sonderzeichen bei der Übertragung von IPMI Nachrichten über eine Serielle Schnittstelle im Basis Modus mit ihrer jeweiligen Ersatzzeichenfolge.

Sonderzeichen	Beschreibung	Ersatzzeichenfolge
A0h	Beginn der Übertragung	*AAh B0h*
A5h	Ende der Übertragung	*AAh B5h*
A6h	Empfangsbereitschaft	*AAh A6h*
AAh	Maskierungs-Zeichen	*AAh BAh*
1Bh	ASCII Escape-Zeichen	*AAh 3Bh*

Das Escape Zeichen *1Bh* des American Standard Code for Information Interchange (ASCII) [43] ist deshalb maskiert, weil es im IPMI Protokoll eine besondere Bedeutung übernimmt. Aus diesem Grund muss es wie die anderen Sonderzeichen ebenfalls maskiert werden.

Der einzige Nachteil des Basis Modus der Seriellen Schnittstelle liegt im IPMI proprietären Nachrichtenformat. Im Basis Modus kann ein IPMI Gerät nur andere IPMI-fähige Geräte über die Serielle Schnittstelle ansteuern.

PPP Modus

Über den Point-to-Point Protocol Modus der Seriellen Schnittstelle können auch Verbindungen mit anderen Geräten aufgenommen werden, die zwar keine Implementierung des Basis Modus besitzen, aber höhere Protokolle der TCP/IP Protokollfamilie unterstützen. In diesem Fall wird die Schnittstelle als asynchrone Serielle Schnittstelle mit High-Level Data Link Control (HDLC) [194] Frames verwendet. Die IPMI Nachrichten werden zu diesem Zweck in die PPP Pakete eingebettet, so dass ein leichter Overhead entsteht, der zwischen sieben und zehn Oktette aufweisen kann. Allerdings können die IPMI Nachrichten nicht direkt in PPP Paketen gekapselt werden, sondern es müssen mehrere andere Protokolle dazu verwendet werden. Konkret werden die IPMI Nachrichten in RMCP Paketen versendet, die auch bei der Übertragung über Local Area Network (LAN) Schnittstellen zum Einsatz kommen. Eine nähere Beschreibung findet sich weiter unten in Abschnitt 6.2.5.

In das Protokoll PPP sind verschiedene Sicherheitsfunktionalitäten implementiert wie beispielsweise der Rückrufmechanismus. Außerdem unterstützt PPP mehrere verschiedene Authentifizierungsprotokolle während des Verbindungsaufbaus. Dementsprechend kann auch bei der Errichtung eines Kommunikationsweges zu einem anderen System eine Authentifizierung möglich und notwendig sein. IPMI erlaubt die Verwendung eines von drei Authentifizierungsmechanismen. Es handelt sich dabei um drei verschiedene Versionen des Challenge Handshake Authentication Protocol (CHAP):

CHAP. Das Protokoll CHAP [196] verwendet einen einfachen Mechanismus aus Benutzernamen und Passwörtern, um die Authentizität einer eingehenden Verbindung zu überprüfen. Damit das geheime Passwort nicht unverschlüsselt über das Netzwerk gesendet werden muss, wird beim CHAP Verfahren vom Empfänger ein Challenge[1] Wert an den Verbindungsaufbauenden gesendet. Dieser bildet dann einen Hashwert über den Challenge Wert, das Passwort und die eindeutige Identifikationsnummer des ursprünglichen Paketes und sendet ihn an den Authentifizierungs-Server zurück. Dieselbe Berechnung kann auch bei der Gegenstelle durchgeführt werden, so dass schließlich beide Seiten über den selben Hashwert verfügen. Auf dieser Basis kann der Verbindungsaufbauende zuverlässig authentifiziert werden, ohne dabei das Passwort im Klartext senden zu müssen.

MS-CHAPv1. Das von Microsoft [128] aufgestellte Authentifizierungsprotokoll MS-CHAPv1 [233] ist in großen Teilen identisch mit dem ursprünglichen

[1] Die Übersetzung des Begriffs „Challenge“ lautet „Herausforderung“, was der Bedeutung aber nicht ganz gerecht wird.

Protokoll CHAP, das als Vorlage für dieses proprietäre Protokoll gedient hat. Die Entwicklung der besonderen Version MS-CHAPv1 des Protokolls CHAP ist deshalb notwendig gewesen, weil Microsoft Systeme zur ordnungsgemäßen Authentifizierung das Passwort im Klartext benötigen. Aus diesem Grund wird das Passwort nicht in der irreversible verschlüsselten Form des Protokolls CHAP an den Authentifizierungs-Server gesendet, sondern das Passwort wird auf Basis des Data Encryption Standard (DES) zusammen mit dem Challenge Wert reversibel verschlüsselt und versendet. Aus diesem Grund ist das Protokoll MS-CHAPv1 als hochgradig unsicher zu betrachten und erfüllt nicht die Ansprüche eines zeitgemäßen sicherheitskritischen Systems. Beim Einsatz von IPMI sollte also von den Authentifizierungs-Servern das Protokoll MS-CHAPv1 möglichst nicht akzeptiert oder für die Authentifizierung vorgeschlagen werden.

MS-CHAP-v2. Das neuere Authentifizierungsprotokoll MS-CHAPv2 [232] ist ein direkter Nachfolger des MS-CHAPv1 Protokolls. Zur Abschwächung der bekannten Sicherheitsproblematik wurde vor allem ein anderes Verschlüsselungsverfahren verwendet. Anstatt des unsicheren DES Algorithmus kommt bei MS-CHAPv2 eine Kombination aus dem weniger alten und etwas sichereren Secure Hash Algorithm 1 (SHA-1) sowie der Einweg-Hashfunktion Message Digest 4 (MD4) [174] zum Einsatz. Insbesondere durch den Einsatz einer Einweg-Hashfunktion wird endgültig das Versenden des Passworts im Klartext oder in einer reversibel verschlüsselten Form unterbunden.

Terminal Modus

Der Terminal Modus ist vorwiegend für die Unterstützung von lokalen „dummen“ Terminals in das IPMI Rahmenwerk integriert worden. Im Normalfall ist nur eine dedizierte serielle Verbindung zu genau einem einzigen Terminal möglich. Aus diesem Grund findet auch keine Verschlüsselung des verwendeten Passworts statt. Das Terminal gilt als lokal angeschlossen und mit einer separaten Leitung verbunden.

Der Terminal Modus unterstützt ausschließlich das Senden und Empfangen von druckbaren Zeichen. Aus diesem Grund werden die einzelnen Oktette in zweistellige, hexadezimale Zeichenketten kodiert. Diese werden jeweils durch ein Leerzeichen voneinander getrennt zur vollständigen IPMI Nachricht zusammengefügt. Im Terminal Modus werden die IPMI Nachrichten anschließend in eckige Klammern („[“ und „]“) eingefasst und mit einem abschließenden Zeichen für einen Zeilenumbruch gesendet. Das Ende einer Nachricht kann daher leicht an der Kombination aus einer schließenden eckigen Klammer („]“) und einem Zeichen für den Zeilenumbruch erkannt werden.

Zusätzlich zu den gekapselten IPMI Nachrichten können auch noch andere Meldungen im Terminal Modus verschickt werden, die jeweils mit dem Schlüsselwort `SYS` beginnen. Diese Meldungen erfüllen im Wesentlichen Aufgaben zur Verwaltung der Terminal Verbindungen. Neben den im Folgenden

genannten Befehlen sind noch weitere herstellerspezifische Befehle möglich, die aber ebenfalls mit dem Schlüsselwort `SYS` beginnen.

`[SYS GET BOOTOPT]`. Mit dem diesem Befehl kann ein IPMI Gerät den Zustand aller oder auch ausgewählter Boot Flags und Parameter erfragen.

`[SYS GET TCFG]`. Mit dem Befehl `[SYS GET TCFG]` kann ein IPMI Gerät die Konfigurationseinstellungen des flüchtigen und des nichtflüchtigen Speichers abrufen. Als Antwort werden die beiden Zeilen `V:x1 x2` und `N:y1 y2` geliefert, wobei die beiden Oktette `x1` und `x2` sowie `y1` und `y2` den Werten der Konfigurationsoptionen entsprechen.

`[SYS HEALTH QUERY]`. Dieser Befehl dient dem BMC dazu, den Statuszustand des Systems abfragen zu können. Die Antwort besteht aus einer kompakten Zeile mit den sieben in Tabelle 6.2 aufgelisteten Parametern und deren Werten. Die Parameter sind jeweils durch ein Leerzeichen voneinander getrennt, und die Werte sind jeweils durch einen Doppelpunkt „:“ von den Parameternamen separiert. Mit Ausnahme des Einschaltzustands kann jeder Parameter einen der sechs in Tabelle 6.3 angegebenen Werte annehmen. Bei der Kompaktdarstellung werden sowohl die Parameter als auch deren Werte in der abgekürzten Version dargestellt. Wird der Befehl `[SYS HEALTH QUERY]` zusätzlich mit dem optionalen Kommandozeilenparameter `-V` angegeben, so erfolgt die Ausgabe in der Langfassung, bei der nicht die Kürzel jeweils aus der zweiten Spalte der beiden Tabellen verwendet und die Parameter zeilenweise ausgegeben werden.

`[SYS IDENTIFY]`. Mit diesem Befehl kann der BMC das IPMI System dazu veranlassen, ein optisches oder akustisches Signal an die Außenwelt zu senden. Dieser Befehl macht nur dann einen Sinn, wenn ein Administrator sich in unmittelbarer Nähe des Gerätes befindet, um diese optischen oder akustischen Signale wahrzunehmen. Sinnvoll ist diese Identifizierung vor allem in Umgebungen, in denen viele verwaltete Geräte stehen. Durch das Aussenden des `[SYS IDENTIFY]` Befehls kann der Administrator schnell und unkompliziert das System physikalisch finden. Wird der Befehl ohne weitere Kommandozeilenparameter gegeben, so wird eine Signalisierung von 15 Sekunden initiiert. Soll eine andere Zeitspanne verwendet werden, so kann diese über den optionalen Kommandozeilenparameter `-ON` spezifiziert werden. Mit dem Kommandozeilenparameter `-OFF` kann die Signalisierung des Systems explizit beendet werden.

`[SYS POWER OFF]`. Mit diesem kann der BMC das IPMI Gerät herunterfahren. Dies bezieht sich nicht auf den BMC selbst, da dieser auch ohne Spannungsversorgung des Hauptsystems aktiv bleibt.

`[SYS POWER ON]`. Mit diesem kann ein BMC ein heruntergefahrenes IPMI Gerät wieder starten. Der BMC kann diese Funktion ausführen, da er auch ohne Spannungsversorgung des Hauptsystems aktiv bleibt.

Tabelle 6.2. Zustandsparameter bei der Ausgabe des `SYS HEALTH QUERY` Befehls für serielle Verbindungen im Terminal Modus. Bei Angabe des Kommandozeilenparameters `-V` wird die Langfassung verwendet, ansonsten setzt sich die Ausgabe aus den Abkürzungen zusammen.

Parameter	Abkürzung	Beschreibung
Power	PWR	Einschaltzustand des Systems. Mögliche Werte sind *ON* für „eingeschaltet“, *OFF* für „ausgeschaltet“, *SLEEP* oder in der Kurzfassung *SLP* für einen nicht näher spezifizierten Ruhezustand, *S1* bis *S4* für die jeweiligen Ruhezustände und *Unknown* oder in der Kurzfassung *??* für einen unbekannten Einschaltzustand.
Health	H	Allgemeiner Zustand des gesamten Systems.
Temperature	T	Temperatur des Systems.
Voltage	V	Spannung im System.
Power Supply	PS	Zustand der Spannungsversorgungseinheit des Systems.
Cooling	F	Zustand der Kühleinheiten wie Lüfter.
Drives	D	Zustand der Laufwerke im System.
Security	S	Zustand des Systems für den physikalischen Zugriffsschutz.
Other	O	Andere, herstellerspezifische Angaben.

Tabelle 6.3. Mögliche Werte für die meisten der Zustandsparameter des `SYS HEALTH QUERY` Befehls für serielle Verbindungen im Terminal Modus. Bei Angabe des Kommandozeilenparameters `-V` wird die Langfassung verwendet, ansonsten setzt sich die Ausgabe aus den Abkürzungen zusammen.

Wert	Abkürzung	Beschreibung
OK	ok	Der Wert für den Zustandsparameter liegt innerhalb normaler Grenzen.
Non-critical	nc	Warnung für ein Überschreiten oder ein Unterschreiten der normalen Grenzen dieses Zustandsparameters.
Critical	cr	Alarm für das Überschreiten oder Unterschreiten kritischer Grenzen dieses Zustandsparameters.
Non-recoverable	nr	Das System ist durch das Überschreiten oder Unterschreiten kritischer Grenzen in großer Gefahr, einen irreversiblen Schaden zu nehmen, oder das System ist bereits irreparabel geschädigt.
Unspecified fault	uf	Es wurde ein Fehler mit unspezifizierter Schwere diagnostiziert.
Unknown	??	Der Wert des Zustandsparameters kann nicht ermittelt werden, beispielsweise weil das System ausgeschaltet ist.

[SYS PWD]. Dieser Befehl dient dem Aufbau oder dem Trennen einer Seriellen Verbindung im Terminal Modus. Bevor eine Terminal Verbindung vom IPMI Gerät verwendet werden kann, muss zunächst die Verbindung errichtet werden. Ohne eine bestehende Verbindung nimmt eine Gegenstelle keine IPMI Befehle oder Terminal Kommandos an. Zum Aufbau kann optional auch eine korrekte Authentifizierung des IPMI Gerätes erforderlich sein. Die möglichen Kommandozeilenparameter werden in Tabelle 6.4 beschrieben.

Tabelle 6.4. Kommandozeilenparameter des SYS PWD Befehls für serielle Verbindungen im Terminal Modus. Genau einer der drei Parameter muss angegeben werden.

Parameter	Beschreibung
-U	Eine neue serielle Verbindung wird im Terminal Modus geöffnet. Nach dem Parameter -U wird der Benutzername gefolgt vom Passwort spezifiziert. Nach erfolgreicher Anmeldung nimmt die Gegenstelle IPMI und andere Terminal Befehle entgegen. Wird ein falscher Benutzername oder ein falsches Passwort angegeben, so wird die Verbindung beendet, als ob der Befehl mit dem Parameter -X aufgerufen worden wäre. Die Antwort auf eine Autorisierung mittels [SYS PWD] Befehl und Kommandozeilenparameter -U ist entweder bei Erfolg ein [OK] oder bei Misserfolg ein [ERR xx], wobei xx für die Fehlernummer in hexadezimaler Schreibweise repräsentiert.
-N	Eine neue serielle Verbindung wird im Terminal Modus geöffnet. Im Unterschied zum Parameter -U wird kein Benutzername angegeben, sondern lediglich ein Passwort.
-X	Eine eventuell bestehende serielle Verbindung im Terminal Modus wird augenblicklich getrennt. Um weitere IPMI oder andere Terminal Befehle versenden zu können, muss zunächst eine neue Verbindung mit dem Parameter -U oder -N aufgebaut werden.

[SYS RESET]. Durch diesen Befehl wird ein sofortiger Neustart durch den BMC veranlasst. Wenn weniger als eine Minute davor die Boot Flags ebenfalls verändert worden sind, so gelten diese noch als gültig und werden bei der eingeleiteten Neustartsequenz berücksichtigt.

[SYS SET BOOT]. Durch diesen Terminal Befehl kann ein IPMI Gerät die Flags für den Neustart konfigurieren. Bei der nächsten Initialisierung des Systems werden dann die spezifizierten Flags berücksichtigt. Mit den insgesamt fünf Oktette umfassenden Flags kann unter anderem spezifiziert werden, von welchem Medium das System beim nächsten Neustart bootet, wie geschwätzig die Bildschirmausgabe beim nächsten Neustart sein soll, welche Tasten des Systems unberücksichtigt bleiben sollen, aber auch, ob ein eventuell vorhandenes Passwort des Basic Input/Output System (BIOS) umgangen werden soll.

Einige der Flags – insbesondere das letztgenannte – können durchaus einen negativen Einfluss auf die Sicherheit des Systems haben. Aus diesem Grund werden die Flags vom System automatisch nach einer Minute invalidiert, so dass nicht aus Versehen die eingestellten Optionen erst Monate später ausgeführt werden.

`[SYS SET BOOTOPT]`. Mit diesem Befehl kann ein IPMI Gerät neben den Boot Flags auch noch alle anderen Boot-Parameter beeinflussen. Die spezifizierten Optionen werden am Ende des `[SYS SET BOOTOPT]` Befehls angegeben.

`[SYS SET TCFG]`. Dieser Befehl erlaubt es einem IPMI Gerät, Konfigurationsangaben zum Terminal selber zu machen. Bei den Konfigurationsparametern wird zwischen flüchtigem und nichtflüchtigem Speicher differenziert. Dazu muss genau einer der beiden Parameter aus Tabelle 6.5 gefolgt von zwei Oktetten für die zu setzenden Werte angegeben werden.

Tabelle 6.5. Kommandozeilenparameter des `SYS PWD` Befehls für serielle Verbindungen im Terminal Modus. Genau einer der drei Parameter muss angegeben werden.

Parameter	Beschreibung
`-V`	Mit der Angabe des Parameters `-V` werden Konfigurationsangaben für den flüchtigen („volatile“) Speicher gemacht. Es folgen zwei Oktette Daten.
`-N`	Mit der Angabe des Parameters `-N` werden Konfigurationsangaben für den nichtflüchtigen („non-volatile“) Speicher gemacht. Es folgen zwei Oktette Daten.

`[SYS TMODE]`. Mit diesem Befehl kann ein IPMI Gerät in Erfahrung bringen, ob die Serielle Schnittstelle im Terminal Modus verwendet wird. Handelt es sich tatsächlich um eine Terminal Verbindung, so erhält das IPMI Gerät die positive Bestätigungsmeldung `[OK TMODE]` zurück.

6.2.5 LAN

Eine sehr flexibler Kommunikationskanal eröffnet sich einem IPMI System durch die Verwendung einer LAN Schnittstelle. Die große Flexibilität, die unter anderem auch einen deutlichen Sicherheitsgewinn beinhaltet, zieht allerdings gleichzeitig einen signifikanten Overhead bei der Kommunikation mit sich. Im Gegensatz zur Seriellen Schnittstelle, die im Basis Modus gerade einmal ein Start-Oktett und ein Stopp-Oktett aufweist, sind bei der Implementierung eines LAN-Kommunikationskanals weitere niedrigere und höhere Protokolle beteiligt, in welche die jeweiligen Nachrichten eingebettet werden müssen.

VLAN

Der Virtual Local Area Network (VLAN) [212] Mechanismus arbeitet in derselben Schicht des OSI Referenzmodells, in der sich auch die MAC Adressen befinden. Durch den VLAN Standard ist es einem vermittelnden Gerät möglich, Netzwerkpakete noch unterhalb der Netzwerkschicht direkt in der Sicherungsschicht zu behandeln und entsprechend weiterzuleiten. Dazu werden die Pakete mit einer „Markierung“ (Tag) versehen, so dass sie von den vermittelnden Bridges erkannt und der korrekten VLAN Gruppe zugeordnet werden können. IPMI in der Version 2.0 ist in der Lage, die Kommunikation über das LAN auf ein frei konfigurierbares VLAN zu beschränken. Dadurch werden gleichzeitig sämtliche eingehenden Pakete ignoriert, die nicht demselben VLAN zugeordnet sind.

IP

Die Kapselung von IPMI Nachrichten in LAN Paketen ist ausschließlich über das Internet Protocol (IP) möglich. Existenzieller Bestandteil des Protokolls IP ist vor allem die IP Adresse, die jedem Knoten in einem IP Netzwerk zugeteilt wird. Vergleichbar zum ICMB Bus müssen auch die IP Adressen innerhalb eines zusammenhängenden Netzwerkes eindeutig sein. Daher ist die Verwendung einer dynamischen Zuweisung von IP Adressen über das Dynamic Host Control Protocol für IPMI empfohlen. Ein BMC muss zur Verwaltung der IP Adresse des Systems nicht zwangsweise über eine Implementierung von DHCP verfügen. Um in diesem Fall aber dennoch über das Netzwerk kommunizieren zu können, muss das System selbst oder ein anderer Teil der Software diese Aufgabe übernehmen. Eine Möglichkeit besteht darin, dass eine DHCP Implementierung im BIOS für die Verwaltung der IP Adresse sorgt. Nötigenfalls muss das BIOS auch die im IPMI konfigurierte IP Adresse aktualisieren. Probleme, die durch die Begrenzung des Zeitraums, für den eine IP Adresse vergeben wird, entstehen können, müssen dann ebenfalls vom BIOS durch Anforderung einer neuen IP Adresse gelöst werden.

Durch die Verwendung des Internet Protocol und weiterer höherer Protokolle der TCP/IP Protokollfamilie ergibt sich für IPMI Nachrichten, die über das Netzwerk versendet werden, ein nicht unerheblicher Overhead. Der Paket-Overhead allein durch das IP Protokoll beträgt 20 Byte pro IPMI Nachricht.

UDP

Zur Kapselung der verbindungslos übertragenen IPMI Nachrichten bietet sich innerhalb der TCP/IP Protokollfamilie insbesondere das verbindungslose Protokoll UDP an. Die IPMI Spezifikation sieht zwei UDP Ports vor (siehe nächster Abschnitt), über welche die Nachrichten versendet und empfangen werden können. Der Paket-Overhead, der nur durch die Verwendung des UDP Protokolls erzeugt wird, beläuft sich auf 8 Byte pro IPMI Nachricht.

RMCP

In einer weiteren Abstraktionsebene wird innerhalb des Protokolls UDP ein Managementprotokoll eingeführt, das vor allem vollständig unabhängig von einem Betriebssystem arbeitet. Dies ist für IPMI insbesondere deshalb wichtig, da das Betriebssystem nicht nur unabhängig von den Managementfunktionen für das System ausgelegt ist, sondern weil IPMI auch bei ausgeschaltetem System funktionieren soll – also zu einem Zeitpunkt, bei dem kein Betriebssystem zur Verfügung steht. Diese Anforderungen erfüllt das von der Distributed Management Task Force (DMTF) [63] entwickelte Remote Management Control Protocol (RMCP) [70] voll und ganz. Neben IPMI Nachrichten kann das Protokoll RMCP auch die in der Spezifikation beschriebenen Nachrichtentypen wie Alarme des Alerting Standard Formats (ASF) [70] kapseln.

Die RMCP Nachrichten sind in UDP Pakete eingebettet, die über zwei speziell dafür reservierten Ports versendet werden. Der primäre RMCP Port zum Versenden von RMCP Nachrichten trägt die Nummer 623; der sekundäre Port, der zur Übertragung von verschlüsselten Informationen eingerichtet worden ist, trägt die Nummer 664. Unabhängig davon, ob ein RMCP Paket eine IPMI Nachricht enthält, oder ob es sich um eine andere RMCP Nachricht zur Steuerung des Nachrichtenflusses handelt, beinhalten die RMCP Pakete einen Overhead von genau 4 Byte, von denen das vierte Oktett die genaue Klasse der RMCP Nachricht spezifiziert. Das genaue Format des RMCP Paketkopfes ist in Abbildung 6.3 veranschaulicht.

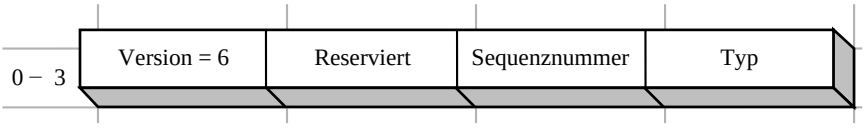

Abb. 6.3. Schematische Darstellung des Paketkopfes eines RMCP Paketes.

ASF *Ping* Nachrichten in RMCP. Mit der ASF *Ping* Nachricht kann ein System in Erfahrung bringen, ob die Gegenstelle RMCP unterstützt und welche RMCP Nachrichten die Gegenstelle akzeptiert und welche sie generieren kann. Abbildung 6.4 veranschaulicht den schematischen Aufbau einer ASF *Ping* Nachricht.

ASF *Pong* Nachrichten in RMCP. Auf eine eingegangene ASF *Ping* Nachricht antwortet ein RMCP Gerät mit der ASF *Pong* Nachricht, mit welcher das Gerät auch darüber Auskunft gibt, welche RMCP Nachrichtentypen es unterstützt. Das genaue Format einer ASF *Pong* Nachricht ist in Abbildung 6.5 schematisch dargestellt.

RMCP ACK Nachricht. Da es sich bei UDP um ein verbindungsloses Protokoll handelt, bei dem naturgemäß die Auslieferung der Nachrichten nicht

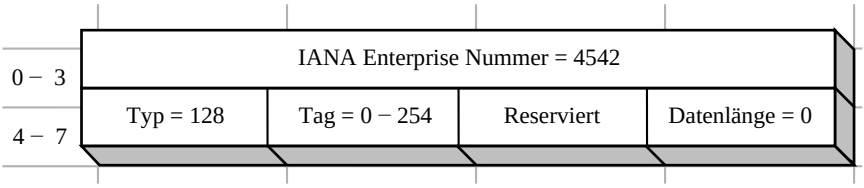

Abb. 6.4. Schematische Darstellung eines ASF *Ping* Nachrichtenpaketes.

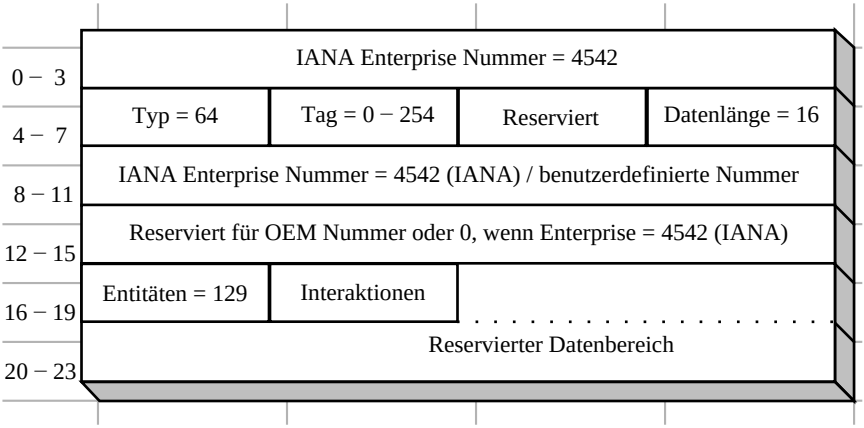

Abb. 6.5. Schematische Darstellung eines ASF *Pong* Nachrichtenpaketes.

gewährleistet wird, ist in RMCP ein eigener Mechanismus zur Überprüfung der korrekten Auslieferung der RMCP Nachrichten implementiert. Auf jedes eingehende RMCP Datenpaket, das einen Wert ungleich FFh für seine Sequenznummer besitzt, muss nach RMCP Spezifikation ein Bestätigungspaket vom Typ RMCP ACK zurückgesendet werden. Da aber in IPMI ebenfalls bereits ein ähnlicher Mechanismus integriert ist, kann auf den RMCP ACK Mechanismus zur Vermeidung von überflüssigem Datenaufkommen verzichtet werden. In RMCP gekapselte IPMI Nachrichten senden daher eine RMCP Sequenznummer von FFh, so dass keine RMCP Bestätigungspakete gesendet werden. Auch die ASF *Ping* und *Pong* Nachrichten erfordern keine separaten RMCP ACK Pakete.

Das Format der RMCP ACK Pakete ist identisch mit dem Format normaler RMCP Nachrichten. Mit Ausnahme des vierten Oktetts im Paketkopf, welches die Nachrichtenklasse enthält, werden alle Werte des RMCP ACK Paketkopfes von der eingehenden Nachricht kopiert. Bei der Nachrichtenklasse wird zusätzlich das höchstwertige Bit gesetzt.

Erweiterungen in RMCP+

Das RMCP+ Protokoll stellt eine Erweiterung zum RMCP Protokoll dar und unterstützt auch das Versenden von IPMI Paketen in der Version 2.0. Während RMCP nur IPMI Version 1.5 Pakete transportieren kann, werden durch IPMI Version 2.0 Nachrichten über RMCP+ Pakete vorrangig Sicherheitsfunktionalitäten implementiert. Dazu zählen eine bessere Authentifizierung und eine bessere Verschlüsselung.

6.2.6 Serial Over LAN

Eine Mischung aus einer LAN Verbindung und einer Kommunikation über die Serielle Schnittstelle findet sich in der Serial Over LAN (SOL) Verbindung. Bei dieser wird der BMC über die LAN Schnittstelle von der Managementstation angesprochen. Der Multiplexer trennt gleichzeitig den externen seriellen Anschluss des Systems von der Seriellen Schnittstelle des Systems und verbindet letztere direkt mit dem BMC. Die eingehenden LAN Pakete werden schließlich vom BMC transformiert und in übersetzter Form an die Serielle Schnittstelle des Systems weitergeleitet. Der BMC übernimmt bei dieser Kommunikation die Transformation zur Seriellen Schnittstelle bis hin zur Simulation der Hardware und der damit verbundenen Flusskontrolle. Erst durch die Transformation des BMC kann eine Verbindung zu einer System-Software aufgebaut werden, die lediglich eine Kommunikation über die Serielle Schnittstelle unterstützt, ohne dass gleichzeitig die Beschränkungen einer Punkt-zu-Punkt Verbindung über die Serielle Schnittstelle gelten. Die Kommunikation zwischen dem System und dem BMC verläuft über die von der System-Software vorgegebenen Verbindung über die Serielle Schnittstelle. Zwischen dem BMC und der entfernten Management-Software wiederum kommt die bereits weiter oben beschriebene Kommunikation über UDP und RMCP+ Pakete zum Einsatz.

6.2.7 PCI Management Bus

Die Schnittstelle Peripheral Component Interconnect (PCI) [151] erlaubt die Anbindung von Erweiterungskarten an komplexe Systeme. Damit auch das Intelligent Platform Management Interface Überwachungsfunktionalitäten der Erweiterungskarten ansprechen und erreichen kann, lässt sich IPMI auch über die PCI Schnittstelle kapseln.

6.3 Sicherheitsmechanismen

IPMI verwendet vor allem in der Version 2.0 mehrere Sicherheitsmechanismen, wie Authentifizierung und Verschlüsselung. Aber auch die Vorgängerversion 1.5 des Intelligent Platform Management Interface unterstützt einfache Sicherheitsmechanismen, zu denen beispielsweise eine Session-basierte Kommunikation zählt.

6.3.1 Sessions

IPMI unterstützt drei verschiedene Kommunikationsformen bezüglich Sessions. Die verschiedenen Typen werden abhängig von der Art der physikalischen Verbindung und dem eingesetzten Protokoll verwendet. Bei direkten Hardware-Verbindungen über ICMB, IPMB oder den PCI Bus ist eine Authentifizierung weniger wichtig und die Kommunikation verläuft verbindungslos. Bei einfachen Punkt-zu-Punkt Verbindungen wie dem Basis Modus oder dem Terminal Modus einer seriellen Verbindung wird zwischen den beiden Gesprächspartnern genau eine Verbindung aufgebaut. Bei mehrfach genutzten Leitungen wie einer LAN-Verbindung oder einer PPP-Verbindung können prinzipiell auch mehrere Sessions gleichzeitig initiiert werden.

Verbindungslose Kommunikation

Die verbindungslose IPMI Kommunikation verläuft direkt und ohne vorherige Authentifizierung der Kommunikationspartner. Diese Einbußen bei der Sicherheit wird durch eine dedizierte Hardware-Verbindung erkauft, die nur eine physikalische Kompromittierung der Verbindung erlaubt.

Kommunikation über eine Einzelverbindung

Insbesondere bei Verbindungen an der Seriellen Schnittstelle können nur einzelne direkte Verbindungen aufgebaut werden. Zu Beginn einer Einzelverbindung findet dennoch eine Benutzerauthentifizierung statt, die sich aber nicht auf den Rest der Kommunikation ausweitet.

Mehrfachverbindungen

Mehrfachverbindungen finden sich vor allem bei mehrfach genutzten Leitungen. Die LAN Schnittstelle ist ein gutes Beispiel für eine Mehrfachverbindung, denn über den Netzwerkanschluss können sehr viele verschiedene Verbindungen gleichzeitig abgewickelt werden. Das bezieht sich auch auf die IPMI Verbindung. Über eine LAN Schnittstelle können sowohl mehrere unterschiedliche Verbindungen zum selben IPMI Gerät aufgebaut werden, als auch mehrere parallele Verbindungen vom selben Host und Benutzer. Aus diesem Grund ist auch eine Authentifizierung der Verbindungen besonders wichtig. Prinzipiell verwendet IPMI entweder eine Benutzerauthentifizierung oder auch eine Authentifizierung auf Nachrichtenbasis.

6.3.2 Authentifizierung

Eine Authentifizierung kann nur bei gleichzeitiger Errichtung einer Verbindung zum Einsatz kommen. RMCP+ stellt für diesen Zweck das RMCP+

Authenticated Key-Exchange Protocol (RAKP) bereit, mit dem ein initialer Schlüsselaustausch vorgenommen werden kann. Zur Verfügung stehen die beiden Algorithmen HMAC-SHA1 und HMAC-MD5; optional kann auch auf eine Überprüfung der Integrität verzichtet werden.

RMCP+ Authenticated Key-Exchange Protocol

Wichtigster Bestandteil des Protokolls RAKP sind die vier RAKP Nachrichten eins bis vier, über welche der Schlüsselaustausch vollzogen wird. Erst nach erfolgreichem Nachrichtenaustausch besteht eine authentifizierte Verbindung.

RAKP Nachricht 1. Die RAKP Nachricht 1 wird von der Managementstation aus aufgerufen und beinhaltet neben der Session ID noch eine 16 Byte lange Zufallszahl sowie die Angabe der gewünschten Benutzerrolle, welche die Managementstation ausüben möchte. Optional kann auch noch ein Benutzername mitgesendet werden.

RAKP Nachricht 2. Erhält eine überwachte Komponente eine RAKP Nachricht 1, so überprüft diese die Korrektheit der mitgesendeten Session ID und optional noch den Benutzernamen. Sind alle Angaben konsistent, so sendet die überwachte Komponente eine RAKP Nachricht 2 als Antwort an die Managementstation. In dieser Nachricht sind eine eigene Session ID für den Rückweg, eine eigene Zufallszahl und die global eindeutige Identifikationsnummer der überwachten Komponente enthalten. Wurde einer der möglichen Verschlüsselungsalgorithmen spezifiziert, so wird der RAKP Nachricht 2 unter Berücksichtigung der empfangenen Zufallszahl noch ein Authentifizierungs-Code für den Schlüsselaustausch beigefügt.

RAKP Nachricht 3. Die Managementstation überprüft bei Erhalt einer RAKP Nachricht 2 zunächst wieder alle relevanten Daten auf Plausibilität. Dazu zählt in erster Linie die übergebene Session ID und die global eindeutige Identifikationsnummer der überwachten Komponente. Anschließend wird der vom überwachten Gerät berechnete Authentifizierungs-Code überprüft. Sind auch hier alle Angaben korrekt, so sendet die Managementstation eine RAKP Nachricht 3 als Rückantwort. Dazu werden erneut die Session ID und ein neuerlich berechneter Authentifizierungs-Code für den Schlüsselaustausch mit dem Paket verschickt.

RAKP Nachricht 4. Erreicht auch das zweite Paket der Managementstation die überwachte Komponente, so wird zunächst erneut die Korrektheit der erhaltenen Session ID und des Authentifizierungs-Codes überprüft. Treten keine Fehler auf, so sendet das überwachte System die vierte und letzte RAKP Nachricht. Inhalt sind wieder die Session ID für den Rückweg und eine Integritätsprüfsumme. Beide Werte werden beim Empfänger vor Einrichtung der authentifizierten Verbindung noch einmal gegengeprüft.

RAKP-HMAC-SHA1

Beim Verbindungsaufbau über die vier RAKP Nachrichten werden insgesamt drei verschiedene Prüfsummen generiert. Einigen sich die Kommunikationspartner auf den RAKP-HMAC-SHA1 Algorithmus, so werden die 20 Byte langen Authentifizierungs-Codes für den Schlüsselaustausch jeweils durch den HMAC-SHA1 Algorithmus berechnet. Für die Integritätsprüfsumme wird jedoch der HMAC-SHA1-96 Algorithmus zur Ermittlung der 12 Byte langen Prüfsumme verwendet.

RAKP-HMAC-MD5

Bei Verwendung des RAKP-HMAC-MD5 Algorithmus zum Aufbau einer authentifizierten Verbindung werden alle drei berechneten Prüfsummen der letzten drei RAKP Nachrichten mit dem HMAC-MD5 Algorithmus erzeugt. Aus diesem Grund sind sowohl der Authentifizierungs-Code für den Schlüsselaustausch als auch die Integritätsprüfsumme jeweils 16 Byte lang.

6.3.3 Integrität

Damit auch im weiteren Verlauf der Kommunikation die einzelnen Pakete auf Integrität überprüft werden können, kann eine Signierung der einzelnen Nachrichten erzwungen werden. Es kann aber auch auf die Signierung verzichtet werden, wie es sich beispielsweise bei den dedizierten Einzelverbindungen anbietet. Sind derartige Verbindungen erst einmal authentifiziert, so kann die Verbindung als glaubwürdig betrachtet werden. Eine weitere Signierung aller folgenden Pakete ist dann nicht mehr notwendig.

MD5-128

Kommt zur Überprüfung der Integrität einzelner Nachrichtenpakete der MD5-128 Algorithmus zum Einsatz, so wird für die Signatur ein Hashwert über den Paketinhalt gebildet. Um die Integrität wahren zu können, geht in die Bildung des Einweg-Hashwerts auch noch das Benutzerpasswort ein, dass beiden Kommunikationspartnern bereits bekannt ist.

HMAC-SHA1-96

Wird der HMAC-SHA1-96 Algorithmus für die Berechnung der Signatur verwendet, so geht der bereits zuvor ausgehandelte Session Integrity Key (SIK) in die Berechnung mit ein.

HMAC-MD5-128

Haben sich die Kommunikationspartner auf den HMAC-MD5-128 Algorithmus zur Berechnung der Signaturen geeinigt, so geht ähnlich dem HMAC-SHA1-96 Algorithmus der bereits zuvor ausgehandelte Session Integrity Key in die Berechnung mit ein.

6.3.4 Verschlüsselung

RMCP+ unterstützt auch den Aufbau von verschlüsselten Verbindungen, auch wenn dies nicht zwingend vorgeschrieben ist. Im Wesentlichen können die beiden Algorithmen xRC4 und AES-CBC zur Verschlüsselung verwendet werden, wobei ersterer in zwei verschiedenen Ausprägungen unterschiedlicher Schlüssellänge vorliegt. Die verwendeten Verschlüsselungsalgorithmen hängen dabei in großem Maße von der Güte der erstellten Initialisierungsvektoren (IV) ab. Dieser wiederum wird durch einen möglichst guten Zufallszahlengenerator sichergestellt.

Zufallszahlengenerator

Ein derart kleines und kompaktes System wie der BMC verfügt in vielen Fällen nicht über einen eigenen Hardware-Zufallsgenerator, so dass auf Pseudozufallszahlen zurückgegriffen werden muss. Zwar arbeitet RMCP+ mit einer 160-Bit langen Zufallszahl und zwei weiteren zufällig erstellten 32-Bit Werten, jedoch ist ein Pseudozufallszahlengenerator immer schlechter als ein „echter“ Zufallszahlengenerator, der in Hardware realisiert ist. Was den Schutzbedarf von IPMI Nachrichten betrifft, so sollte aber selbst der Pseudozufallszahlengenerator noch eine ausreichende Sicherheit bieten.

xRC4-40 und xRC4-128

Auch der auf dem RC4 [176] basierende Verschlüsselungsalgorithmus xRC4 hängt von der Güte der Initialisierungsvektoren IV ab, die eine Länge von 16 Byte aufweisen. Für eine vollständig verschlüsselte Kommunikation müssen beide Kommunikationspartner einen Initialisierungsvektor IV erzeugen. Der letztlich verwendete Schlüssel wird vergleichsweise einfach mit einer Einweg-Hashfunktion ermittelt, die bei xRC4 aus der MD5 Funktion besteht. Bei dieser Berechnung gehen nur der Session Integrity Key SIK und der zufällige Initialisierungsvektor IV ein.

xRC4-40 und xRC4-128 sind nahezu identische Verschlüsselungsmechanismen. Es existiert nur die einzige Ausnahme der Schlüssellänge. Beim xRC4-128 Algorithmus kommen die vollen 128 Bit des berechneten Schlüssels zum Einsatz; beim xRC4-40 Algorithmus sind es lediglich die ersten 40 Bit.

AES-CBC-128

Der AES-CBC-128 Algorithmus arbeitet wie der xRC4 Algorithmus ebenfalls über einen zufälligen Initialisierungsvektor IV und einen Schlüssel, der nur aus dem Session Integrity Key SIK abgeleitet wird.

Cipher Block Chaining (CBC) verknüpfen die zu einzelnen Blöcken zerteilten Daten dadurch, dass sie jeden Block vor der Verschlüsselung mit dem vorherigen Block in Verbindung setzen. Der Initialisierungsvektor kommt dabei nur im ersten Block zum Tragen, da dieser keinen Vorgängerblock besitzt. Im Unterschied dazu wird bei AES-CBC-128 keine Verbindung zwischen den einzelnen Blöcken hergestellt. Jeder Block wird mit einem neuen Initialisierungsvektor verschlüsselt. Aus diesem Grund ist auch die Unvorhersagbarkeit der IV von entscheidender Wichtigkeit.

6.4 IPMI Nachrichten

Innerhalb der IPMI Spezifikation sind eine ganze Reihe von verschiedenen Befehlssätzen definiert, welche die unterschiedlichsten Aufgaben innerhalb des IPMI Rahmenwerkes übernehmen. Neben den globalen IPMI Befehlen, die von jedem IPM-fähigen Gerät unterstützt werden, existieren noch weitere Kommandos, die teilweise von der Art des entsprechenden Gerätes abhängen. Beispielsweise finden sich Befehle, die speziell für Geräte der Gehäuseüberwachung oder für die einzelnen Sensoren konzipiert sind. Andererseits finden sich auch Befehle für Geräte mit besonderer Funktionalität wie beispielsweise Befehle für die SEL oder die SDR Datenbank. Viele dieser Kommandos sind aus diesem Grund optional von den Geräten zu unterstützen und sind daher auch nur in ausgewählten Komponenten implementiert.

6.4.1 Globale IPMI Befehle

Die globalen IPMI Nachrichten sind allgemeiner Natur und müssen von jedem IPM Gerät unterstützt werden. Die beiden von absolut jedem IPMI Gerät zu implementierenden Befehle dienen der Abfrage der eindeutigen Identifikationsnummer des Gerätes sowie der Ergebnisse des letzten Selbsttests. Ein Selbsttest wird typischerweise beim Einschalten oder bei einem Kaltstart des Gerätes durchgeführt. Manche Geräte können auch während des Betriebs oder bei einem Warmstart einen Selbsttest durchführen. Auch wenn ein Gerät keinen Selbsttest unterstützt, muss es dennoch den entsprechenden globalen Befehl unterstützen[2]

Zu den globalen Befehlen zählen noch weitere allgemeine Befehle, die aber nicht zwingend vom IPMI Gerät unterstützt werden müssen. Hierzu zählen

[2] In diesem Fall sendet das Gerät als Antwort einen Code zurück, der auf die fehlende Implementierung eines Selbsttests hinweist.

beispielsweise Befehle, die einen Kaltstart oder einen Warmstart auslösen, oder Befehle, welche den Status der ACPI Spannungsversorgung auslesen oder ändern können. Außerdem kann über einen weiteren Befehl optional die global eindeutige Geräte-Identifikationsnummer ausgelesen werden.

6.4.2 Befehle zur Erkennung des verfügbaren Befehlssatzes

Ein IPMI 2.0 Gerät unterstützt eine lokale Firewall, durch welche die von außen eingehenden Befehle eingeschränkt werden können. Zwar hat diese Firewall in einem einzelnen Gerät nur bedingt einen Nutzen, jedoch kann in einem komplexeren System aus mehreren Einzelkomponenten eine logische Trennung der verschiedenen Teilsysteme eingerichtet werden. So kann beispielsweise verhindert werden, dass ein Zugriff auf eines der Teilsysteme eine negative Auswirkung auf die anderen unabhängigen Teilsysteme hat. Gleichzeitig wurde ein neuer Satz an Befehlen in das IPMI System eingebracht, der vor allem dem Managementsystem eine Möglichkeit bietet, Informationen über eventuelle Einschränkungen in den Befehlen durch die lokale Firewall in Erfahrung bringen zu können. Außerdem sind eine Reihe von Befehlen spezifiziert worden, mit denen die Firewall konfiguriert werden kann. Typischerweise ist der Zugriff über die System Schnittstelle nicht eingeschränkt, während hingegen für andere Kommunikationswege wie der IPMB oder der ICMB Bus eine Einschränkung der verfügbaren Befehle vorgenommen werden kann.

6.4.3 IPMI LAN Befehle

Die IPMI LAN Befehle dienen vor allem der Konfiguration der LAN Schnittstelle und den zugehörigen notwendigen Parametern. Tabelle 6.6 liefert eine Auflistung über die konfigurierbaren und auslesbaren Parameter in Bezug auf die LAN Schnittstelle. Über die LAN Befehle können außerdem Statistiken zu den einzelnen Protokollen ausgelesen werden, die bei der Übermittlung der LAN Pakete zum Einsatz kommen – also IP, UDP und RMCP.

6.4.4 RMCP+ Befehle

Befehle aus der Kategorie RMCP+ ermöglichen die Steuerung einer RMCP+ Verbindung zwischen Managementstation und dem verwalteten IPMI Gerät. Die insgesamt zehn verschiedenen RMCP+ Befehle behandeln im Wesentlichen die Art und den Status der verwendeten IPMI Befehle. Die möglichen Arten von RMCP+ Paketen beinhalten beispielsweise die vier RAKP Nachrichten zum Aufbau einer authentifizierten Verbindung oder auch ganz normale IPMI Pakete. Mit Hilfe der RMCP+ Befehle kann nun der Typ der Nutzdaten einer RMCP+ Nachricht festgelegt oder abgefragt werden. Außerdem können verschiedene Informationen über den verwendeten Pakettyp und die verfügbaren Nutzdatentypen abgefragt werden.

Tabelle 6.6. Parameter, die über entsprechende LAN Befehle ausgelesen oder verändert werden können.

Parameter	Zugriff
Theoretisch unterstützte Authentifizierungsmethoden.	Lesen
Praktisch auswählbare Authentifizierungsmethoden.	Lesen/Schreiben
MAC Adresse des BMC.	Lesen
IP Adresse des Systems.	Lesen/Schreiben
Subnetzmaske.	Lesen/Schreiben
Quelle, über welche die IP Adresse und die Subnetzmaske zugewiesen wurde. Mögliche Quellen können eine manuelle statische Konfiguration, ein vom BMC betriebener DHCP Server sowie das BIOS oder das Betriebssystem sein.	Lesen
IP Adresse des Standard Gateways.	Lesen/Schreiben
IP Adresse des sekundären Standard Gateways.	Lesen/Schreiben
MAC Adresse des Standard Gateways.	Lesen/Schreiben
MAC Adresse des sekundären Standard Gateways.	Lesen/Schreiben
Optionen für das IP Protokoll. Konfiguriert werden können der Time-to-Live (TTL) Wert, der Type of Service (ToS) Wert oder die IP-Flags wie das „Don't Fragment" Flag.	Lesen/Schreiben
Nummer für den primären RMCP Port. Der Standardwert lautet 623.	Lesen/Schreiben
Nummer für den sekundären RMCP Port. Der Standardwert lautet 664.	Lesen/Schreiben
Community Name für gesendete SNMP Nachrichten.	Lesen/Schreiben
Anzahl der Ziele für zu sendende Alarme. Maximal 15 Ziele werden unterstützt.	Lesen
Art der Empfängersysteme für die Alarme.	Lesen/Schreiben
IP Adressen der Empfängersysteme für die Alarme.	Lesen/Schreiben
Identifikationsnummer des VLANs.	Lesen/Schreiben
Anzahl der zur Verfügung stehenden Verschlüsselungsalgorithmen.	Lesen
Art der zur Verfügung stehenden Verschlüsselungsalgorithmen.	Lesen
Prioritäts-Reihenfolge, in welcher die zur Verfügung stehenden Verschlüsselungsalgorithmen ausgewählt werden sollen.	Lesen/Schreiben

6.4.5 Befehle für die Serielle Schnittstelle

Da die Serielle Schnittstelle in drei verschiedenen Modi betrieben werden kann, sind auch die Konfigurationsmöglichkeiten für diese Schnittstelle entsprechend vielfältig. Neben den wenigen Befehlen, die einzelne Aspekte der Konfiguration für die Serielle Schnittstelle beeinflussen können, existieren vor allem zwei generische Befehle zum Auslesen und Konfigurieren der vielen Parameter aus Tabelle 6.7. Die Parameter beziehen sich teilweise auf die Serielle Schnittstelle im Allgemeinen oder aber auch auf einzelne Modi der Seriellen Schnittstelle.

Tabelle 6.7. Parameter, die über entsprechende Befehle für die Serielle Schnittstelle ausgelesen oder verändert werden können.

Parameter	Zugriff
Theoretisch unterstützte Authentifizierungsmethoden.	Allgemein
Praktisch auswählbare Authentifizierungsmethoden.	Allgemein
Modus der seriellen Verbindung. Mögliche Werte sind Basis, PPP oder Terminal.	Allgemein
Aktivierung der automatischen Terminierung der Session.	Allgemein
Zeitraum an Inaktivität, nach der eine Verbindung automatisch beendet wird.	Allgemein
Zeit, die zwischen zwei Wählversuchen nach einem „Besetzt"-Zeichen auf der Leitung gewartet wird.	Allgemein
Einstellungen für die Verbindungsparameter der Gegenstelle. Konfiguriert werden kann die Verbindungsgeschwindigkeit, die Flusskontrolle sowie das Vorhandensein eines Stopp- oder Paritäts-Bits.	Allgemein
Anzahl der Zielrufnummern.	Allgemein
Einstellungen für den Rückruf. Grundsätzlich unterstützt RMCP+ neben dem IPMI proprietären Rückrufverfahren auch das Callback Control Protocol (CBCP), welches neben mehreren vordefinierten Rückrufnummern auch die Möglichkeit zur Spezifizierung der Rückrufnummer durch den Anrufer ermöglicht.	Allgemein
Einstellungen für die Verbindungsparameter auf der seriellen Leitung. Konfiguriert werden kann die Verbindungsgeschwindigkeit und die Flusskontrolle.	Allgemein
Einstellungen für den Multiplexer der Seriellen Schnittstelle, welcher einen gleichzeitigen Zugriff von mehreren Systemkomponenten auf die Serielle Schnittstelle erlaubt. Neben der generellen Aktivierung und Deaktivierung der gemeinsamen Verwendung der Seriellen Schnittstelle kann auch genau spezifiziert werden, welche Komponenten Zugriff auf die Schnittstelle erhalten dürfen.	Allgemein
Zeitdauer, die eine Nachricht auf einem Pager mindestens angezeigt wird, bevor sie durch eine nachfolgende Nachricht überschrieben werden kann.	Allgemein
Community Name für gesendete SNMP Nachrichten.	Allgemein
Anzahl der Ziele für zu sendende Alarme. Maximal 15 Ziele werden unterstützt.	Allgemein
Art der Empfängersysteme für die Alarme.	Allgemein
Anzahl der Konten für das Telocator Alphanumeric Protocol (TAP) [44] zum Senden von Pager Nachrichten.	Allgemein
TAP Konto.	Allgemein

(Fortsetzung auf nächster Seite)

Tabelle 6.7. Parameter, die über entsprechende Befehle für die Serielle Schnittstelle ausgelesen oder verändert werden können. (Fortsetzung)

Parameter	Zugriff
TAP Passwort.	Allgemein
TAP Pager Identifikations-Zeichenkette.	Allgemein
Spezifizierung der TAP Verbindungsparameter.	Allgemein
Konfigurationsparameter für den Terminal Modus. Einstellungsmöglichkeiten sind die Aktivierung der Zeilenbearbeitungsmöglichkeit, die Behandlung von Eingaben zum Löschen von Zeichen, die Angabe der Zeichenkette für die Angabe einer neuen Zeile, das Unterdrücken der Ausgabe von empfangenen Eingabebefehlen oder die Art des Verbindungsaufbaus.	Terminal Modus
Einstellungen für die Klingeldauer, nach welcher der BMC einen eingehenden Anruf erkennt und darauf beispielsweise mit einem Wake-On-Ring reagiert.	PPP Modus
Befehlskette zur Initialisierung des Modems.	PPP Modus
Zeichenkette, die als Maskierungs-Zeichen für das Modem gilt.	PPP Modus
Befehlskette zum Trennen einer bestehenden Verbindung durch das Modem.	PPP Modus
Befehlskette für den Wählvorgang des Modems.	PPP Modus
Anzahl der IP Adressen für Ziele von Alarmen.	PPP Modus
IP Adressen der Ziele für Alarme.	PPP Modus
Einstellungen für das Aushandeln der Optionen beim Aufbau einer PPP Verbindung.	PPP Modus
Nummer für den primären RMCP Port. Der Standardwert lautet 623.	PPP Modus
Nummer für den sekundären RMCP Port. Der Standardwert lautet 664.	PPP Modus
Authentifizierungsverfahren für den PPP Verbindungsaufbau. Mögliche Mechanismen sind das Password Authentication Protocol (PAP) [108], CHAP sowie MS-CHAP in den beiden Versionen eins und zwei.	PPP Modus
Challenge Wert des Empfängers, der für den Aufbau einer authentifizierten Verbindung gesendet wird.	PPP Modus
Angaben zur Async-Control-Character-Maps (ACCM), über welche die Maskierungszeichen beim Verbindungsaufbau ausgehandelt werden.	PPP Modus
Anzahl der PPP Konten.	PPP Modus
Zeiger auf die zu verwendende Zielrufnummer.	PPP Modus
IP Adresse des anzuwählenden PPP Servers.	PPP Modus
Benutzername des PPP Kontos.	PPP Modus
Name der Domäne bei Verwendung der MS-CHAP Version eins oder zwei Authentifizierungsverfahren.	PPP Modus

(Fortsetzung auf nächster Seite)

Tabelle 6.7. Parameter, die über entsprechende Befehle für die Serielle Schnittstelle ausgelesen oder verändert werden können. (Fortsetzung)

Parameter	Zugriff
Passwort des PPP Kontos.	PPP Modus
Authentifizierungsverfahren.	PPP Modus
Anzahl der Sekunden, die eine bestehende Verbindung mindestens aktiv bleibt, bevor sie getrennt werden kann.	PPP Modus
Quell- und Zieladresse des UDP Proxy Paketes.	PPP Modus
Größe des Sendepuffers für den UDP Proxy.	PPP Modus
Größe des Empfangspuffers für den UDP Proxy.	PPP Modus
IP Adresse, die einer entfernten Konsole vom BMC zugewiesen werden kann, falls dies erwünscht wird.	PPP Modus
Eigene Einwahlnummer für das aktuelle Gerät, sofern vorhanden.	PPP Modus

6.4.6 Befehle für die SOL Kommunikation

Ganze drei Befehle für die SOL Kommunikation existieren in der IPMI Spezifikation. Mit dem ersten Befehl lässt sich lediglich die Nachricht übermitteln, dass eine SOL-Verbindung aufgebaut worden ist und dass daher alle bestehenden Verbindungen über die Serielle Schnittstelle zwangsweise getrennt worden sind. Mit den beiden anderen Befehlen können die in Tabelle 6.8 ausgelesen oder konfiguriert werden.

6.4.7 Gehäuse-Befehle

Die Gehäuse-Befehle sind größtenteils auf die Abfrage und Konfiguration der Parameter zu Spannungsversorgung und Initialisierungsvorgängen ausgerichtet. Über die Gehäuse-Befehle kann außerdem die Funktion des Hauptschalters und des Reset-Knopfes außer Kraft gesetzt werden, sofern das Gehäuse diese Funktion unterstützt. Zur Identifizierung der Möglichkeiten des Gehäuses existiert daher ein eigener Befehl. Zusätzlich kann der Neustart des Systems über einen der Befehle initiiert werden, sowie der Grund für den letzten Neustart über einen anderen Befehl in Erfahrung gebracht werden.

6.4.8 Ereignis-Befehle

Es existieren ganze drei Ereignis-Befehle, von denen zwei lediglich das Ziel für die zu sendenden Ereignismeldungen abfragen oder setzen können. Der dritte Befehl enthält schließlich eine Ereignismeldung, die anschließend an das konfigurierte Ziel gesendet wird. Jede Ereignismeldung enthält die folgenden sieben Informationen:

1. Eindeutige Identifikationsnummer des Erstellers dieser Ereignismeldung.

Tabelle 6.8. Parameter, die über entsprechende LAN Befehle ausgelesen oder verändert werden können.

Parameter	Zugriff
Angabe darüber, ob die zum Aufbau einer SOL Verbindung notwendigen Bedingungen erfüllt sind.	Lesen
Aktivierung der Authentifizierung sowie der Autorisierung.	Lesen/Schreiben
Angaben zur Übermittlung von Teilpaketen. Dies betrifft sowohl die Zeit in Millisekunden, die ein BMC nach dem Empfang von Teilen einer Nachricht wartet, bevor er den Teilinhalt weitersendet, als auch die Anzahl der Zeichen, die eine gesendete Teilnachricht mindestens enthalten muss.	Lesen/Schreiben
Angaben zur Nachrichtenwiederholung. Empfängt der BMC auf eine gesendete Nachricht keine Bestätigung, so unternimmt er eine einstellbare Anzahl von Wiederholungsversuchen jeweils nach einer ebenfalls einstellbaren Zeitspanne.	Lesen/Schreiben
Zu verwendende Verbindungsgeschwindigkeit über die Serielle Schnittstelle.	Lesen/Schreiben
Zu verwendende Verbindungsparameter wie MAC Adresse oder IP Adresse.	Lesen
Zu verwendende RMCP Port Nummer.	Lesen(/Schreiben)

2. Versionsnummer des verwendeten Typs von Ereignismeldung.
3. Typ des Sensors, welcher die Ereignismeldung generiert hat.
4. Nummer des Sensors, welcher die Ereignismeldung generiert hat.
5. Richtung der Meldungsübermittlung (senden oder empfangen).
6. Art des gemeldeten Ereignisses.
7. Weitere Daten zum gemeldeten Ereignis – abhängig von der genauen Ereignisart.

Im Wesentlichen existieren nur drei verschiedene Sensortypen, die jeweils verschiedene Ereignisse auslösen können. Unterschieden wird jeweils zwischen diskreten Sensoren, die eine maximale Anzahl von 15 verschiedenen Zuständen annehmen können, binären Sensoren, die genau zwei mögliche Zustände annehmen können und Grenzwert-Sensoren, die einen analogen Messwert[3] halten können, der beim Überschreiten oder Unterschreiten von verschiedenen Grenzwerten ein Ereignis auslösen kann. Tabelle 6.9 listet die verschiedenen Ereignisse auf, die von den Sensoren erzeugt werden können.

[3] Da es sich zwar um analoge Sensoren handeln kann, die Auswertung aber in einem digitalen System stattfindet, besitzen die Sensoren in letzter Konsequenz auch nur diskrete Zustände – wenn auch sehr viele.

Tabelle 6.9. Mögliche Ereignistypen für die drei verschiedenen Sensorklassen diskret, binär und Grenzwert.

Sensorklasse	Ereignistyp
Grenzwert	Unterschreiten eines unteren Grenzwertes. Man unterscheidet zwischen den drei verschiedenen Arten von unteren Grenzwerten nichtkritisch, kritisch und nichtwiederherstellbar. Es handelt sich um eine Verschlechterung des Zustands.
Grenzwert	Überschreiten eines unteren Grenzwertes. Man unterscheidet zwischen den drei verschiedenen Arten von unteren Grenzwerten nichtkritisch, kritisch und nichtwiederherstellbar. Es handelt sich um eine Verbesserung des Zustands.
Grenzwert	Überschreiten eines oberen Grenzwertes. Man unterscheidet zwischen den drei verschiedenen Arten von oberen Grenzwerten nichtkritisch, kritisch und nichtwiederherstellbar. Es handelt sich um eine Verschlechterung des Zustands.
Grenzwert	Unterschreiten eines oberen Grenzwertes. Man unterscheidet zwischen den drei verschiedenen Arten von oberen Grenzwerten nichtkritisch, kritisch und nichtwiederherstellbar. Es handelt sich um eine Verbesserung des Zustands.
diskret	Wechsel in einen spezifizierten Zustand. Mögliche Zustände sind Leerlauf, arbeitend und ausgelastet.
binär	Status erreicht oder verlassen.
binär	Zustand für einen voraussichtlichen Fehler erreicht oder verlassen.
binär	Innerhalb oder außerhalb der Grenzwerte.
binär	Performanz ausreichend oder ungenügend.
diskret	Änderung des Kritikalitätszustands: • Zustandsverbesserung zu *OK* • Zustandsverbesserung zu *nichtkritisch* • Zustandsverbesserung zu *kritisch* • Zustandsverschlechterung zu *nichtkritisch* • Zustandsverschlechterung zu *kritisch* • Zustandsverschlechterung zu *nichtwiederherstellbar*
binär	Gerät vorhanden oder entfernt.
binär	Gerät aktiviert oder deaktiviert.
diskret	Änderung der Verfügbarkeit. Mögliche Zustandswerte sind: System läuft, Testphase, System ausgeschaltet, System aktiv, System inaktiv, System außer Betrieb, Systemleistung heruntergestuft und System im Energiesparmodus.
diskret	Änderung des Zustands der Redundanzeinrichtung. Mögliche Werte sind volle Redundanz, eingeschränkte Redundanz und Verlust der Redundanz aus verschiedensten Gründen.
diskret	Änderung des ACPI-Spannungszufuhr-Zustands. Mögliche sind die vier Werte *D0*, *D1*, *D2* und *D3*.

6.4.9 Befehle für ereignisbasierte Alarme und Aktionen

Der ereignisbasierte Mechanismus von IPMI trägt den Namen Platform Event Filtering (PEF). Er basiert auf den verschiedenen Ereignissen, die von den Sensoren ausgelöst werden können, auf der Filterliste, die für eine Erkennung ausgewählter Ereignisse verantwortlich ist, auf den Aktionen, die durch Ereignisse angestoßen werden können, sowie auf den Alarm-Regeln, die eine Definition von verschiedenen Zielen für die Alarme der einzelnen Ereignisse erlauben.

Ereignisse

Die Ereignisse wurden im vorigen Abschnitt für die Ereignis-Befehle behandelt und sind in der Tabelle 6.9 aufgelistet. Eine Unterscheidung findet nach dem Typ des Sensors statt. Die Ereignismeldungen können je nach Typ ein bis drei Oktette an zusätzlichen Daten mitführen, welche das Ereignis näher spezifizieren.

Aktionen

Die möglichen Aktionen decken im Wesentlichen den Einschaltzustand des Systems ab. Als Aktion kann daher das Einschalten des Systems, das Ausschalten des Systems, eine Neuinitialisierung des Systems oder das Auslösen eines Diagnose-Interrupts sein. Parallel dazu kann für alle Aktionen gleichzeitig ein Alarm ausgelöst und versendet werden.

Filterliste

Mit Hilfe der Filterliste können Aktionen und Alarme mit einzelnen Ereignissen verknüpft werden. Jeder Eintrag in der Filterliste spezifiziert das Ereignis sowie die auszulösenden Alarme und Ereignisse:

- Filter ist aktiv oder inaktiv.
- Auszulösendes Ereignis aus Abschnitt 6.4.9.
- Nummer der anzuwendenden Alarm-Regel, falls ein Alarm ausgelöst werden soll.
- Mit dem Ereignis verbundene Kritikalität.
- Identifikationsnummer des Auslösers dieses Ereignisses.
- Sensortyp, welcher das Ereignis ausgelöst hat.
- Nummer des Sensors, welcher das Ereignis ausgelöst hat.
- Ereignistyp aus Tabelle 6.9.
- Angaben zum Bitmuster, mit denen die Daten eines Ereignisses übereinstimmen müssen, damit dieser Filter angewendet wird. Die Spezifikation der Bitmuster verläuft dreistufig: Zuerst wird eine Bitmaske definiert, mit

welcher die für dieses Ereignis irrelevanten Bits ausgeblendet werden können. Mit dem zweiten Bitmuster wird spezifiziert, welche Daten-Bits des übrigen Ereignisses verglichen werden sollen. Mit dem letzten Bitmuster kann anschließend über eine exakte oder eine `oder` Verknüpfung entschieden werden.

Alarm-Regeln

In der Alarmtabelle befinden sich ein oder mehrere Alarmeinträge, die sich aus genau drei Informationen zusammensetzen. Mit der Alarmnummer können mehrere Einträge aus der Alarmtabelle zu einem Alarm-Regelsatz zusammengefasst werden. Auf diese Weise können mehrere Ziele mit demselben Ereignis und Alarm verknüpft werden. Die zweite Information beschreibt den genauen Kanal und das genaue Ziel für die Auslieferung der zugehörigen Alarmmeldung. Mit der dritten Angabe für einen Eintrag in der Alarmtabelle kann die genaue Nachricht beeinflusst werden, welche über die spezifizierten Kanäle zu den angegebenen Zielen gesendet wird.

Durch diesen Mechanismus können Ereignisse sehr gut eingeteilt werden und zur Behandlung an unterschiedliche Gruppen weitergeleitet werden. Gerade in großen Netzwerken findet häufig eine Aufgabenteilung der Administratoren statt, die durch die Alarm-Regeln flexibel unterstützt wird.

Konfigurationsparameter

Die Befehle für ereignisbasierte Alarme und Aktionen dienen der Verwaltung der Ereignisse, Aktionen, Filter und Alarm-Regeln. In Tabelle 6.10 sind die verschiedenen Parameter aufgelistet, welche durch entsprechende Befehle für ereignisbasierte Alarme und Aktionen ausgelesen oder verändert werden können.

6.4.10 Logging-Befehle für die SEL Datenbank

Das IPMI-eigene Logging wird vom SEL Mechanismus übernommen. Die System Event Log Datenbank speichert Systemmeldungen verschiedener Komponenten des Systems. Verantwortlich für den Empfang und die Verwaltung der Systemnachrichten zeichnet der SEL Empfänger, der auch unabhängig von der SEL Datenbank betrieben werden kann. Damit nicht nur der SEL Empfänger neue Nachrichten in die Datenbank einpflegen kann, sondern damit auch andere Komponenten wie die Management-Software Nachrichten aus der Datenbank auslesen können, unterstützt die SEL Datenbank unterschiedliche Leseoperationen und Schreiboperationen. Die wichtigsten Befehle unterstützen das Eintragen neuer Systemnachrichten sowie das Entfernen von Systemnachrichten aus der Datenbank. Daneben existieren einige Befehle, mit deren Hilfe detailliertere Informationen über die einzelnen Einträge und die

Tabelle 6.10. Parameter, die über entsprechende Befehle für ereignisbasierte Alarme und Aktionen ausgelesen oder verändert werden können.

Parameter	Zugriff
PEF Konfiguration. Über diesen Parameter kann das Platform Event Filtering global eingeschaltet und ausgeschaltet werden. Außerdem kann eine Verzögerung der Aktivierung des PEF nach dem Einschalten oder einer Neuinitialisierung des Systems aktiviert werden.	Lesen/Schreiben
Verzögerungszeit für die Aktivierung des PEF nach dem Einschalten oder einer Neuinitialisierung des Systems.	Lesen/Schreiben
Verzögerungszeit für die Aktions- und Alarmgenerierung nach dem Einschalten oder einer Neuinitialisierung des Systems.	Lesen/Schreiben
Erlaubte Aktionen aus Abschnitt 6.4.9.	Lesen/Schreiben
Anzahl der Einträge in der Filterliste.	Lesen
Filterliste mit allen Einträgen.	Lesen/Schreiben
Anzahl der Regeln in der Alarm-Tabelle.	Lesen
Alarm-Tabelle mit allen Einträgen.	Lesen/Schreiben
Global eindeutige Identifikationsnummer des Systems für das Versenden von PET Nachrichten.	Lesen/Schreiben
Anzahl von benutzerdefinierten Alarm-Nachrichten.	Lesen
Liste der benutzerdefinierten Alarm-Nachrichten.	Lesen/Schreiben

gesamte SEL Datenbank in Erfahrung gebracht werden können. Zu den abrufbaren Informationen der Datenbank zählen beispielsweise die unterstützte SEL Versionsnummer, die aktuelle Uhrzeit des Datenbanksystems, die Anzahl der Einträge in der Datenbank oder der für weitere Einträge noch zur Verfügung stehende Speicherplatz. Außerdem können auch Aktionen auf die Datenbank angewendet werden, wie beispielsweise das Setzen der Uhrzeit des Datenbanksystems oder das Löschen der kompletten Datenbank.

Jeder Eintrag in der SEL Datenbank unterliegt einem vorgegebenen Schema. Herstellereigene Systemnachrichten dürfen vom vorgegebenen Raster abweichen und können mit weniger Informationen gespeichert werden. Alle anderen Meldungen werden mit den folgenden neun Angaben in der Datenbank abgelegt:

1. Eindeutige Identifikationsnummer des Eintrags.
2. Typ der Systemmeldung.
3. Zeitstempel für den Eintrag. Die Zeitangabe wird vom Datenbanksystem zu den einzelnen Nachrichten hinzugefügt.
4. Identifikationsnummer der Systemkomponente, welche die Systemnachricht erzeugt hat.
5. Versionsnummer, in welcher die Systemnachricht gespeichert worden ist. Dies ist zur Zeit immer die IPMI 1.5 Version.
6. Sensortyp, welcher die Nachricht erzeugt hat.

7. Nummer des Sensors, welcher die Nachricht erzeugt hat.
8. Ereignistyp nach Tabelle 6.9.
9. Ein bis drei Oktette zusätzliche Daten zum gespeicherten Ereignis.

6.4.11 Befehle für die SDR Datenbank

Die SDR Datenbank ist wesentlich kompakter als die SEL Datenbank, da sie deutlich weniger Einträge speichern muss. In der SDR Datenbank befindet sich lediglich genau ein Eintrag für jeden Sensor im System. Die Management-Software kann beim Überprüfen des Systemstatus die Liste aller Sensoren sukzessive abarbeiten und jeweils die gemessenen Werte aufnehmen. Antwortet ein Sensor nicht, so ist dies mit dem Ausfall des Sensors und daher mit einem Fehlerzustand gleichzusetzen.

Im Normalfall hat die SDR Datenbank einen eher statischen Charakter und wird einmal bei der Installation des Systems manuell initialisiert. Zu diesem Zweck verfügt IPMI über einen speziellen Befehl, mit dem Sensoren zum System hinzugefügt werden können. Werden weitere Sensoren an das System angeschlossen, so erfolgt die Aktualisierung häufig ebenfalls manuell und mit demselben Befehl. Das Löschen eines einzelnen Eintrags ist nicht möglich, so dass die Management-Software zunächst die vollständige SDR Datenbank auslesen sollte, die entsprechenden Einträge intern im Speicher löschen sollte und schließlich nach dem vollständigen Löschen der Datenbank die neuen Einträge wieder in die SDR Datenbank schreiben sollte.

Mit der Ausnahme der Löschfunktion stehen für die SDR Datenbank im Wesentlichen dieselben Befehle zur Verfügung wie für die SEL Datenbank. Dies sind beispielsweise die Befehle, mit deren Hilfe detailliertere Informationen über die einzelnen Einträge und die gesamte SDR Datenbank in Erfahrung gebracht werden können. Die abrufbaren Informationen für die Datenbank setzen sich zusammen aus Angaben zur unterstützten SDR Versionsnummer, zur aktuellen Uhrzeit des Datenbanksystems, zur Anzahl der Einträge in der Datenbank oder zu dem für weitere Einträge noch zur Verfügung stehenden Speicherplatz. Es können ebenfalls Aktionen auf die Datenbank angewendet werden wie beispielsweise das Setzen der Uhrzeit des Datenbanksystems oder das Löschen der kompletten Datenbank.

Im Unterschied zur SEL Datenbank existieren noch drei weitere Befehle, von denen zwei das Einschalten und das Ausschalten des „Update“-Modus steuern. Es handelt sich dabei um einen Zustand der SDR Datenbank, der eine Aktualisierung der SDR Datenbank erlaubt, ohne dass Konflikte mit parallel laufenden Anfragen an die Datenbank auftreten können. Der dritte zusätzliche Befehl aktiviert den Initialisierungsagenten, der beim Start des Systems notwendige Initialisierungsschritte durchführt. Der Initialisierungsagent arbeitet beim Start oder beim Neustart des Systems die Liste der in der SDR Datenbank hinterlegten Sensoren ab und sorgt gleichzeitig für einen definierten Ausgangszustand des Systems. Hierzu zählen vor allem die verschiedenen Grenzwerte der Grenzwert-Sensoren. Der Initialisierungsagent wird ebenfalls

aufgerufen, wenn das System aus einem Stromsparmodus aufgeweckt wird. In diesem speziellen Fall kann es notwendig sein, nicht den definierten Ausgangszustand einzustellen, sondern den Zustand vor dem Herunterfahren des Systems. Eine Kopplung mit dem ACPI hilft gleichzeitig beim Erkennen einer Neuinitialisierung des Systems.

6.4.12 Sensorbefehle

Auch für die einzelnen Sensoren im System existieren verschiedene Befehle, über die sowohl Informationen ausgelesen als auch Konfigurationsparameter eingestellt werden können. Neben einem grundlegenden Befehl, mit dessen Hilfe allgemeine Informationen über den Sensor und dessen SDR Eintrag in Erfahrung gebracht werden können, existieren noch weitere Befehle, welche das Auslesen oder das Verändern verschiedener Parameter erlauben. Dazu zählen vor allem der Sensortyp, die Grenzwerte oder ganz einfach der aktuelle Messwert des Sensors, der mit einem ebenfalls auslesbaren Faktor verknüpft sein kann. Bezüglich der Grenzwerte kann für die Messwerte eine Hysterese eingestellt werden. Diese entspricht einem separat definierbaren Bereich oberhalb und unterhalb des Grenzwertes, innerhalb dessen nach einmaliger Benachrichtigung keine neuen Alarmmeldungen an das System gesendet werden. Erst wenn der Messwert auch den Bereich der Hysterese verlassen hat, ist der Sensor wieder „scharfgeschaltet“ und kann neue Alarme erzeugen und versenden.

7

IEEE 802.1X Port-basierte Netzwerk Zugriffskontrolle

Bei den meisten heutigen Netzwerken handelt es sich um IP Netzwerke, die auf dem Protokoll des Internets beruhen. Die Identifizierung von Komponenten im Netzwerk wird auf der Ebene des IP Protokolls anhand der IP Adressen durchgeführt. Öffentlich über das Internet erreichbare Netzwerkgeräte erhalten in vielen Fällen eine statische aber auf jeden Fall eindeutige IP Adresse. So können die einzelnen Komponenten des Internets eindeutig beschrieben werden. Natürlich existieren auch Ausnahmen von dieser Regelung; so erhalten beispielsweise Hosts, die sich temporär über einen Internet Service Provider (ISP) in das Internet einwählen, oftmals eine dynamisch zugewiesene IP Adresse aus einen Pool von Adressen, welche der ISP verwaltet und nach Belieben an seine Kunden temporär verteilen kann. Abseits des Internets in den lokalen Netzwerken können wiederum spezielle IP Adressen vergeben werden, die nur innerhalb des Intranets eindeutig sein müssen. Verschiedene LANs können so dieselben IP Adressen implementieren, die dann nicht mehr eindeutig sind[1].

Beim Schutz von IP Netzwerken vor unbefugten Kommunikationsbeziehungen beschränkt man sich in vielen Fällen auf die Vermittlungsschicht des OSI Referenzmodells, in der sich die IP Adressen befinden. Auch eine Filterung von Netzwerkverkehr in der unmittelbar darüber liegenden Transportschicht, in der sich Protokolle wie UDP und TCP befinden, ist in IP Netzwerken weit verbreitet. Es existiert jedoch keine Möglichkeit, die Netzwerkkomponenten als solche eindeutig zu identifizieren, da jedes Gerät prinzipiell mit jeder beliebigen IP Adresse konfiguriert werden kann.

Ein Schritt zur Lösung dieses Problems läuft über die Sicherungsschicht des OSI Referenzmodells. Dort ist zu jeder Netzwerkkomponente eine eindeutige MAC Adresse definiert. Völlig unabhängig von der Zuweisung von IP Adressen in den höheren Schichten und der Bindung von Netzwerkprotokollen im Allgemeinen sind die MAC Adressen weltweit einmalig. Eine eindeutige

[1] Spätestens zur Kommunikation über das Internet muss dann aber wieder auf eindeutige Adressen zurückgegriffen werden, beispielsweise über Network Address Translation (NAT).

Identifizierung von Netzwerkkomponenten über MAC Adressen wäre deshalb theoretisch möglich. In der Praxis lässt sich jedoch bei vielen Geräten die IP Adresse umkonfigurieren, so dass die Einmaligkeit der MAC Adressen nicht gewährleistet ist. Dieser Umstand ist sogar in einigen Fällen zwingend notwendig. Als Beispiel sei hier eine redundante Firewall-Architektur angeführt. Im Fehlerfall der primären Firewall wird die transparente Übernahme der Arbeit von der sekundären Firewall dadurch erreicht, dass diese die MAC Adressen des primären Gerätes übernimmt. Die anderen Netzwerkkomponenten können so – ohne dass eine Aktualisierung ihres ARP Speichers notwendig ist – weiterhin mit der Firewall kommunizieren.

Die MAC Adresse allein reicht also nicht zur zweifelsfreien Identifizierung von Netzwerkgeräten aus. Erst eine Kombination aus Kryptographie und den MAC Adressen ermöglicht eine Authentifizierung von Netzwerkkomponenten. An dieser Stelle setzt der 802.1X Standard [211] des Institute of Electrical and Electronics Engineering (IEEE) [89] an. Er regelt den Port-basierten Zugriff von Netzwerkkomponenten auf Ethernet-Netzwerke mit Hilfe kryptographischer Mechanismen. Der IEEE 802.1X Standard soll im Wesentlichen die Authentifizierung von Netzwerkkomponenten unterstützen, die physikalisch an eine Local Area Network (LAN) angeschlossen werden. Dies schließt explizit auch die Anbindung von Netzwerkgeräten an Access-Points eines Wireless Local Area Network (WLAN) mit ein. Die Port-basierte Netzwerk Zugriffskontrolle hilft entscheidend dabei, ein Netzwerk vor unbekannten und nicht autorisierten Geräten zu schützen. Spätestens seit der Einführung von WLAN und der damit verbundenen schwer zu schützenden Luftschnittstelle ist IEEE 802.1X ein wichtiger Mechanismus aus dem Bereich Netzwerksicherheit – der allerdings auch seinen Einfluss auf das Netzwerkmanagement hat.

7.1 Rollenkonzept

Für den Port-basierten Netzwerk Zugriffsschutz des IEEE 802.1X Standards sind grundsätzlich drei verschiedene Typen von Rollen definiert, wobei ein physikalisches System durchaus auch die Rolle von zwei oder sogar allen drei Rollen übernehmen kann. In vielen Fällen handelt es sich bei dem zur Authentifizierung verwendeten System jedoch um drei separate Netzwerkkomponenten, die jeweils eine der Rollen „Bittsteller“ (Supplicant), „Authentifizierer“ (Authenticator) und „Authentifizierungs-Server“ (Authentication Server) übernehmen[2]. Abbildung 7.1 veranschaulicht, wie diese drei verschiedenen Rollen im Netzwerk zusammenspielen.

[2] Die englischen Begriffe ‚Supplicant‘, ‚Authenticator‘ und ‚Authentication Server‘ übersetzen sich zu den deutschen Begriffen ‚Bittsteller‘, ‚Authentifizierer‘ und ‚Authentifizierungs-Server‘. Diese Bezeichnungen könnten aber verwirrend wirken, so dass an dieser Stelle weiterhin die englischen Originalbegriffe verwendet werden sollen.

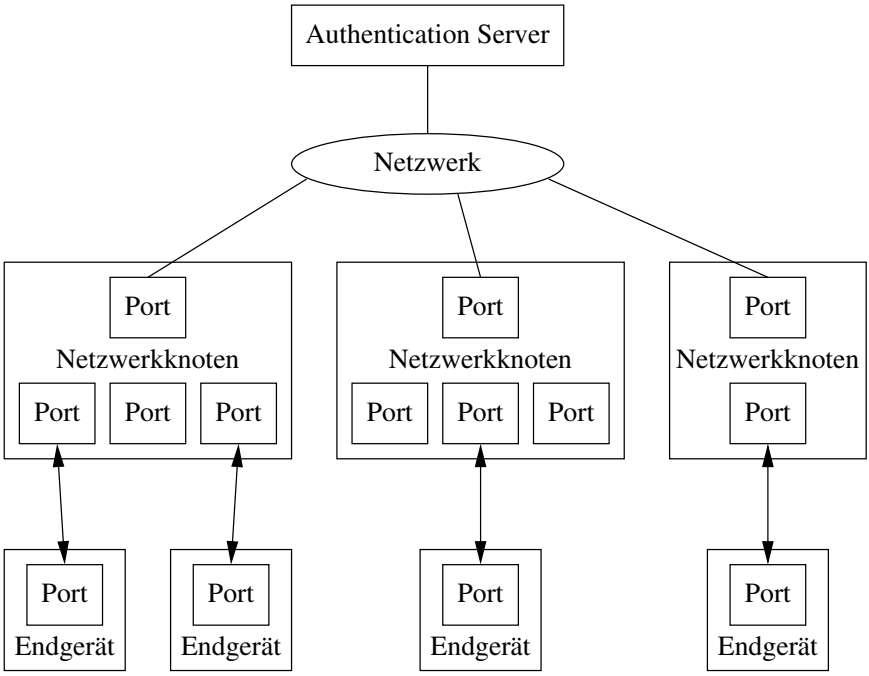

Abb. 7.1. Im IEEE 802.1X Standard spielen die Ports eine entscheidende Rolle. Verschiedene Geräte besitzen eine unterschiedliche Anzahl von Ports. Endgeräte sollen als Geräte mit nur einem Port definiert sein, und Netzwerkknoten sollen als Geräte mit mehr als einem Port definiert sein. Der Authentication Server ist ein Gerät im Netzwerk, welches die Aufgabe der Authentifizierung übernimmt.

7.1.1 Port

Bei der IEEE 802.1X Authentifizierung spielen die Ports der Netzwerkkomponenten die entscheidende Rolle. Ein Endgerät, welches Zugriff auf ein Netzwerk erhalten will, verbindet den eigenen Port mit einem freien Port des Netzwerkes. Allerdings kann diese Verbindung erst nach einer erfolgreichen Authentifizierung genutzt werden. Dazu müssen die beteiligten Ports unterschiedliche Rollen übernehmen. Das Endgerät, welches Zugriff auf das Netzwerk erhalten möchte, tritt gegenüber dem Netzwerk als Bittsteller auf, so dass der Port des Endgerätes die Aufgabe des Supplicants erhält. Auf der anderen Seite der Punkt-zu-Punkt Verbindung steht der Netzwerkknoten. Dieser nimmt die Anfrage des Supplicants entgegen und bildet den Vermittler zwischen dem Endgerät und dem Authentifizierungs-Server. Der entsprechende Port des Netzwerkknotens erhält in diesem Fall die Aufgabe eines Authenticator.

Ports müssen nicht zwangsweise physikalische Ports sein, wie man sie bei Netzwerkswitches oder MAC Bridges vorfindet. Ein Access-Point eines WLAN

bietet drahtlosen Endgeräten ebenfalls einen Port zur Verbindung mit dem Netzwerk an. In diesem Fall sind sowohl der Port auf der Seite des Endgerätes als auch auf der Seite des Netzwerkknotens als logische Ports zu verstehen. Die Definitionen für einen Supplicant und für einen Authenticator sind jedoch vollständig unabhängig von der konkreten Form des Ports. Im Sinne des 802.1X Standards sind Ports lediglich Schnittstellen, zwischen denen verschiedene Geräte eines Netzwerkes Punkt-zu-Punkt Verbindungen aufbauen können.

7.1.2 Supplicant

Als einen Supplicant bezeichnet man den Port eines Netzwerkgerätes, welcher eine Verbindung zu einem IEEE 802.1X kontrollierten Netzwerk herstellen möchte. In diesem Fall tritt die Netzwerkkomponente als Bittsteller gegenüber dem Netzwerk auf, da sie die Herstellung einer Verbindung wünscht. Zur erfolgreichen Verbindung muss ein Supplicant verschiedene Aufgaben erfüllen können.

Antworten auf Authentifizierungsanfragen eines Authenticators

Damit ein Netzwerkgerät sich mit einem 802.1X kontrollierten Netzwerk erfolgreich verbinden kann, muss es auf die Anfragen des Authenticators zur Authentifizierung antworten können. Die Antwort besteht im Wesentlichen in der Übermittlung der Authentifizierungsparameter an den Authentifizierungs-Server. Übermittelte Authentifizierungsparameter können beispielsweise eine Kombination aus Benutzernamen und Passwort, ein Einmal-Passwort oder auch ein Zertifikat sein.

Antworten auf Reauthentifizierungsanfragen eines Authenticators

Im IEEE 802.1X Standard ist es vorgesehen, dass die Identität und die Authentizität jeder zum Netzwerk hinzugefügten Komponente nach Ablauf einer definierbaren Zeitspanne erneut überprüft werden kann. Zu diesem Zweck kann der Authenticator eine Reauthentifizierung initiieren. Um weiterhin auf das Netzwerk zugreifen zu können, müssen sich die Netzwerkkomponenten dann erneut beim Authentication Server authentifizieren.

Initiieren einer Authentifizierung

Wenn ein Netzwerkgerät sich mit einem 802.1X kontrollierten Netzwerk verbinden möchte, muss es gegebenenfalls eine Authentifizierung selbst initiieren. Falls das Gerät eingeschaltet wird und die in diesem Augenblick vom Authenticator gesendeten Anfragen zur Authentifizierung noch nicht empfangen oder ausgewertet werden können, muss die Netzwerkkomponente nach Beendigung seines Initialisierungsvorgangs autark den Authentifizierungsvorgang

einleiten können. Auf diese Weise kann sichergestellt werden, dass eine Authentifizierung auf jeden Fall stattfindet und der Zugriff zum Netzwerk damit auch ermöglicht werden kann. Gleichzeitig muss der Supplicant aber auch mit Netzwerken umgehen können, die nicht IEEE 802.1X kontrolliert sind. Falls vom Netzwerk keine Authentifizierungsanfragen gestellt werden und auch die aktive Initiierung der Authentifizierung vom Supplicant unbeantwortet bleibt, muss der Supplicant von einem freien Zugriff zum Netzwerk ausgehen und die Verbindung als aufgebaut betrachten.

Beenden einer authentifizierten Verbindung

Aus Sicherheitsgründen sollte ein 802.1X authentifiziertes Netzwerkgerät bei einer Änderung seines Zustandes, seiner Konfiguration oder seiner Aufgaben, die eine direkte Auswirkung auf die Authentizität des Gerätes, der laufenden Dienste oder angemeldeten Benutzer hat, eine Beendigung der authentifizierten Verbindung einleiten. Beispielsweise könnte sich ein Benutzer von einem System abmelden, so dass danach auch andere Benutzer Zugriff auf das System und dessen Dienste haben. In einem solchen Fall ändert sich nicht die MAC Adresse des Systems und auch die Verbindung zum Netzwerk wird zu keiner Zeit unterbrochen, so dass der Authenticator die Änderung im System nicht feststellen kann. Die neu angemeldeten Benutzer können allerdings über restriktivere Berechtigungen verfügen. Um diesem Problem vorzubeugen, ist vom Supplicant beim Abmelden eines Benutzers auch die authentifizierte Verbindung zu beenden. Dazu ist lediglich der Authenticator über die Beendigung der Verbindung zu informieren.

7.1.3 Authenticator

Die Rolle des Authenticators wird in 802.1X kontrollierten Netzwerken ebenfalls von einem Port übernommen. Ein Authenticator ist allerdings nicht ein Port eines Systems, das Zugang zu einem Netzwerk erbittet, sondern ein Authenticator ist immer bereits Teil des Netzwerkes und bietet anderen Geräten eine Möglichkeit zur Verbindung mit dem Netzwerk an. Jeder Supplicant, der mit einer Punkt-zu-Punkt Verbindung Zugriff zu einem Netzwerk sucht, muss demnach bei einem Authenticator eine Zugriffserlaubnis einholen. Zur Erfüllung seiner Aufgaben in einem IEEE 802.1X kontrollierten Netzwerk muss ein Authenticator verschiedene Aufgaben übernehmen können.

Senden von Authentifizierungsanfragen an neue Supplicants

Eine wichtige Aufgabe für einen 802.1X Authenticator besteht in der Abwicklung und Initiierung der Authentifizierung des Supplicants. Wird ein neues Netzwerkgerät dem Netzwerk hinzugefügt, so hat der Authenticator sofort Authentifizierungsanfragen an den Supplicant zu stellen. Der Authenticator

ist gleichzeitig für die Wiederholungen der Anfragen verantwortlich, um Übertragungsprobleme bei der Punkt-zu-Punkt Verbindungen zu kompensieren. Bei dem hinzugefügten Gerät kann es sich jedoch auch um eine Komponente handeln, die nicht 802.1X-fähig ist. Auf eine solche Situation kann ein Authenticator unterschiedlich reagieren. Im Normalfall ist dem neuen System der Zugriff auf das Netzwerk gänzlich zu verweigern und es werden auch keine weiteren Authentifizierungsanfragen gesendet. Handelt es sich beim Authenticator jedoch um eine intelligente Netzwerkkomponente, die auch den Virtual Local Area Network (VLAN) [212] Mechanismus unterstützt, so kann dieser einen eingeschränkten Zugriff auf das Netzwerk erlauben. In der Praxis ordnet man einfach alle nicht-authentifizierten Geräte in ein besonderes VLAN ein, in dem dann nur eingeschränkte Dienste zur Verfügung stehen wie beispielsweise ein Dynamic Host Control Protocol (DHCP) Dienst.

Senden von Reauthentifizierungsanfragen an bereits authentifizierte Supplicants

Im IEEE 802.1X Standard ist es vorgesehen, dass die Identität und die Authentizität jeder zum Netzwerk hinzugefügten Komponente nach Ablauf einer definierbaren Zeitspanne erneut überprüft werden kann. Die Aufgabe eines Authenticators besteht in diesem Zusammenhang in der Verwaltung der authentifizierten Verbindungen zum Supplicant. Nach einer definierten Zeitspanne muss der Authenticator gegebenenfalls eine Reauthentifizierung des Supplicants einleiten, die ansonsten identisch zur initialen Authentifizierung abläuft.

Weiterleiten von Paketen zur Authentifizierung zwischen Supplicant und Authentication Server

Haben Authenticator oder Supplicant einen Authentifizierungsvorgang eingeleitet, findet anschließend die eigentliche Authentifizierung zwischen Supplicant und Authentication Server statt. Dem Authenticator wird in diesem Prozess die wichtige Rolle des Vermittlers zuteil, da er die einzelnen Pakete des Supplicants an den Authentication Server und die Antworten zurück an den Supplicant weiterleiten muss. Außerdem muss der Authenticator die Pakete noch transformieren, da die beteiligten Kommunikationspartner unterschiedliche Protokolle verwenden. Hauptursache für die uneinheitliche Sprache ist die Tatsache, dass der Supplicant zum Zeitpunkt der Authentifizierung keinen Zugriff auf das Netzwerk hat. Deshalb haben auch wichtige Vorgänge aus den höheren Protokollen noch nicht stattgefunden wie beispielsweise die Anfrage bei einem DHCP Server zur Zuteilung einer IP Adresse. Das im IEEE 802.1X Standard vorgesehene Protokoll zur Authentifizierung ist das Extensible Authentication Protocol (EAP) [19]. Mittels des Protokolls EAP kann der Authenticator die Authentifizierungsparameter des Supplicant an den Authentication Server übermitteln und dessen Antworten empfangen. Der Supplicant

selber kann aber kein EAP verwenden. Zur Lösung dieses Problem führt der 802.1X Standard das EAP over LAN (EAPOL) Protokoll ein, mit dessen Hilfe der Supplicant die notwendigen Daten an den Authenticator senden kann. Die Aufgabe des Authenticators beschränkt sich explizit auf die Protokolltransformation; die Inhalte der Pakete dürfen in keiner Weise verändert werden.

Verwaltung des Netzwerkzugriffs vom Supplicant in Abhängigkeit vom Ergebnis der (Re-)Authentifizierung

Der Authenticator ist im 802.1X Standard die verwaltende Einheit, welche den Zugriff von Supplicants auf das Netzwerk kontrollieren. Der Authenticator muss also auf verschiedene Ereignisse reagieren und die Verbindung zum Supplicant entsprechend einschränken. Häufig hat der Authenticator dabei auch das vom Authentication Server gesendete Ergebnis einer Authentifizierung zu interpretieren. Besonders auf folgende Situationen muss ein Authenticator im Verlauf einer Verbindung mit einem Supplicant reagieren:

Ein Supplicant verbindet sich mit dem Authenticator. Beim Hinzufügen eines neuen Gerätes zum Netzwerk muss die Verbindung zum Gerät zunächst geschlossen und auf die zur Authentifizierung notwendige Kommunikation beschränkt werden.

Die Authentifizierung war erfolgreich. Nach erfolgreicher Authentifizierung hat der Supplicant uneingeschränkten Zugriff auf das Netzwerk und die Verbindung wird vollständig geöffnet. Je nach den Möglichkeiten des Authenticators kann auch ein differenzierter Zugriff zum Netzwerk gewährt werden. Beispielsweise kann bei Verwendung von VLANs auch ein Zugriff auf einzelne VLANs gestattet werden.

Die Authentifizierung war erfolglos. War die Authentifizierung eines Supplicant erfolglos oder hat gar keine Authentifizierung stattgefunden, so wird die Verbindung geschlossen und verbleibt in einem Zustand wie beim Hinzufügen einer neuen Komponente. Die Initiierung einer erneuten Authentifizierung vom Supplicant ist jederzeit möglich.

Eine Reauthentifizierung war erfolgreich. Am Zustand der Verbindung muss bis zur nächsten Reauthentifizierung nichts geändert werden.

Eine Reauthentifizierung war erfolglos. Kann sich ein Supplicant auf Verlangen des Authenticators nicht erneut authentifizieren, muss die Verbindung geschlossen werden. Die Initiierung einer erneuten Authentifizierung vom Supplicant ist jederzeit möglich.

Der Supplicant beendet die Verbindung. Wünscht der Supplicant explizit die Beendigung der Verbindung, so muss der Authenticator darauf reagieren und die Verbindung schließen. Die Initiierung einer erneuten Authentifizierung vom Supplicant ist jederzeit möglich.

Übernahme der Schnittstelle zwischen Netzwerkmanagement und IEEE 802.1X Mechanismus

Jede an einem IEEE 802.1X kontrollierten Netzwerk aktiv beteiligte Komponente, die entweder die Aufgabe eines Supplicant oder eines Authenticator übernimmt, muss gleichzeitig auch eine Schnittstelle für das Managementsystem bereitstellen. Im Falle des Supplicants müssen nur sehr wenige Objekte an das Netzwerkmanagement zum Lesen oder Schreiben freigegeben werden. Ein Authenticator hingegen muss wesentlich mehr Möglichkeiten für das Netzwerkmanagement zur Verfügung stellen. Der IEEE 802.1X Standard definiert auch eine MIB mit verschiedenen Objekten für die jeweiligen Rollen. Diese MIB ist von den 802.1X-fähigen Geräten entsprechend umzusetzen und zu unterstützen. Die Funktionalitäten, die für das Netzwerkmanagement zur Verfügung gestellt werden müssen, beziehen auch Supplicants mit ein, die nicht authentifiziert sind und demnach auch keinen Zugriff zum Netzwerk haben. Es muss in diesem Fall dennoch eine Möglichkeit vorgesehen werden, wie spezielle Nachrichten des Netzwerkmanagements den Supplicant erreichen können.

7.1.4 Authentication Server

Beim Authentication Server handelt es sich im Unterschied zum Supplicant und dem Authenticator nicht um einen besonderen Port des 802.1X kontrollierten Netzwerks. Der Authentication Server muss nicht zwangsweise selbst ein 802.1X-fähiges Gerät sein, jedoch muss er Teil des Netzwerkes sein und muss vom Authenticator erreichbar sein. Dies ist deshalb notwendig, weil eine Authentifizierung von Supplicants nur mit der im Authentifizierungs-Server vorgehaltenen Datenbank vorgenommen werden kann. Authenticator und Authentication Server können im selben Netzwerkgerät implementiert sein; in der Praxis findet sich diese Lösung aber eher selten. Vielfach wird ein Remote Authentication Dial In User Service (RADIUS) Server [173] als Authentication Server eingesetzt, es können jedoch auch andere Authentication Server verwendet werden. Das im 802.1X Standard verwendete EAP Protokoll erlaubt überdies auch eine Verwendung von verschiedenen Authentifizierungsmethoden, so dass für eine konkrete Implementierung noch mehrere Freiheitsgrade verfügbar sind.

Die Kommunikation zwischen einem Supplicant und dem Authentication Server verläuft zweigeteilt. Der Supplicant sendet seine Authentifizierungsparameter mit Hilfe des durch den IEEE 802.1X Standard definierten Protokolls EAPOL, während der Authentication Server die über das Protokoll EAP erhaltenen Parameter mit seiner Autorisierungsdatenbank abgleicht. Als Vermittler fungiert der Authenticator, welcher die EAP in EAPOL Pakete übersetzt und umgekehrt.

7.2 Kontrollierte und unkontrollierte Ports

Die Port-basierte Zugriffskontrolle des IEEE 802.1X Standards unterscheidet Netzwerk Ports nicht nur nach ihren Aufgaben, sondern auch nach ihrem Authentifizierungsstatus. Zu diesem Zweck spricht man von kontrollierten und unkontrollierten Ports. Unter einem unkontrollierten Port versteht man keinesfalls eine Schnittstelle, die sich der Kontrolle des Netzwerkadministrators entzieht; es handelt sich dabei lediglich um den Zugang der Schnittstelle zum Netzwerk, der nicht unter der Kontrolle von 802.1X steht. Das Sicherheitsmodell entscheidet letztendlich darüber, was authentifizierten und nichtauthentifizierten Benutzern und Systemen zur Verfügung steht. Grundsätzlich kommuniziert ein neu zum Netzwerk hinzugefügtes System zunächst über den unkontrollierten Port, der nur ausgewählte Dienste anbietet. Erst nach erfolgreicher Authentifizierung kann das neue System über den kontrollierten Port auch andere Dienste im Netzwerk nutzen. Abbildung 7.2 veranschaulicht den Zusammenhang zwischen kontrolliertem und unkontrolliertem Port sowie den verschiedenen Steuerungsmechanismen zur Zugriffskontrolle.

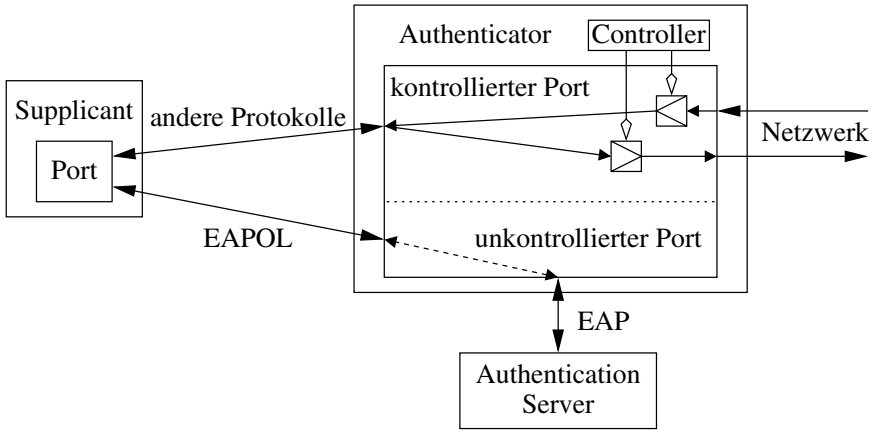

Abb. 7.2. Nach dem IEEE 802.1X Standard kann ein Supplicant jederzeit den Authentication Server über den unkontrollierten Port erreichen. Jede andere Kommunikation verläuft über den kontrollierten Port. Dort übernimmt ein Controller die Zugriffssteuerung auf das Netzwerk in beiden Kommunikationsrichtungen.

7.2.1 Unkontrollierter Port

Wird eine neue Komponente zu einem Netzwerk hinzugefügt, so beginnt die Port-basierte Zugriffskontrolle des 802.1X Standards damit, dass sich das System beim Netzwerk authentifizieren muss. Zu diesem Zweck werden die Authentifizierungsparameter vom Supplicant an den Authentication Server

gesendet. Da zu diesem Zeitpunkt selbstverständlich noch keine Authentifizierung erfolgt ist, dürfte die neue Komponente prinzipiell gar nicht auf das Netzwerk und damit auch nicht auf den Authentication Server zugreifen. Um dieses Problem zu umgehen, muss ein Kommunikationsweg geschaffen werden, welcher an der Zugriffskontrolle vorbei direkt in das Netzwerk verläuft. Damit gleichzeitig der Zugriffsschutz nicht ad absurdum geführt wird, muss diese „unkontrollierte" Kommunikation dennoch eingeschränkt werden.

Genau spezifiziert ist ein Zugriff auf den Authentifizierungs Server über die Protokolle EAP und EAPOL. Ein Supplicant übermittelt nach der Initiierung einer Authentifizierung die notwendigen Authentifizierungsparameter über den unkontrollierten Port an den Authentication Server. Eine gleichzeitige Kontrolle dieses Kommunikationsweges findet automatisch statt, da der Authenticator bei der Authentifizierung als Protokolltransformierer zwischen EAP und EAPOL fungiert. Eine andere Art von Kommunikation über den unkontrollierten Port ist zunächst einmal nicht vorgesehen. Der Vollzugriff auf das Netzwerk findet erst nach erfolgreicher Authentifizierung des Supplicant über den kontrollierten Port statt. Zur Steuerung und Aufrechterhaltung der Zugangsberechtigung beispielsweise bei einer erforderlichen Reauthentifizierung ist selbst dann noch ein EAPOL Zugriff des Supplicant auf den Authenticator über den unkontrollierten Port möglich.

7.2.2 Kontrollierter Port

Der Datenverkehr über den kontrollierten Port eines Authenticators wird abhängig vom Zustand der Authentifizierung des Supplicant gesteuert. Zusätzlich kann auch das Managementsystem manuell den Status von kontrollierten Ports steuern. Vor allem, damit das Netzwerkmanagement auch Kontakt zu nicht-authentifizierten Netzwerkkomponenten aufnehmen kann, lässt sich außerdem der Datenverkehr über den kontrollierten Port in beiden Richtungen zumindest teilweise unterschiedlich konfigurieren.

Authentifizierungsstatus

Beim Hinzufügen einer neuen Komponente zum Netzwerk befindet sich der Status *AuthControlledPortStatus* des kontrollierten Ports im Zustand *unauthorized*. In dieser Situation ist keine Kommunikation zwischen Supplicant und dem Netzwerk möglich. Verläuft der Authentifizierungsvorgang erfolgreich, ändert sich der Status *AuthControlledPortStatus* des kontrollierten Ports in den Zustand *authorized*. Diese Statusänderung hat direkten Einfluss auf das Verhalten des Ports, denn im Zustand *authorized* wird der kontrollierte Port für alle Kommunikation geöffnet und der Supplicant erhält vollen Zugriff auf das Netzwerk. VLAN-fähige Authenticator können zusätzlich den Zugriff noch weiter differenzieren. Werden beispielsweise nacheinander verschiedene Netzwerkkomponenten an einen Authenticator angeschlossen, so können die Systeme nach erfolgreicher Authentifizierung jeweils unterschiedlichen VLANs

zugeordnet werden, so dass sie jeweils andere Zugriffsberechtigungen auf das Netzwerk erhalten.

Portstatus

Der allgemeine 802.1X Status des kontrollierten Ports eines Authenticators kann durch das Netzwerkmanagement separat administriert werden. Auf diese Weise kann der IEEE 802.1X Mechanismus aktiviert und deaktiviert werden. Zu diesem Zweck kann der Zustandswert *AuthControlledPortControl* drei verschiedene Werte annehmen:

Auto. Der 802.1X Mechanismus ist wie beschrieben aktiviert und der Zustand des Authentifizierungsstatus wird vom Ausgang des Authentifizierungsvorgangs zwischen Supplicant und Authentication Server abhängig gemacht.

ForceAuthorized. Der Port-basierte Zugriffskontroll-Mechanismus 802.1X ist deaktiviert und ein Authentifizierungsstatus des kontrollierten Ports von *authorized* wird erzwungen. Der Supplicant erhält in diesem Fall unabhängig von seiner Berechtigung ohne Authentifizierung vollen Zugriff auf das Netzwerk.

ForceUnauthorized. Der IEEE 802.1X Mechanismus ist deaktiviert und ein Authentifizierungsstatus des kontrollierten Ports von *unauthorized* wird erzwungen. Der Supplicant erhält in diesem Fall keinen Zugriff auf das Netzwerk, selbst wenn er über die notwendigen Berechtigungen und Authentifizierungsparameter verfügt.

Richtungen der Zugangskontrolle

Die Zugriffskontrolle des kontrollierten Ports lässt sich in den beiden Kommunikationsrichtungen zumindest teilweise unterschiedlich konfigurieren. Speziell vom Netzwerk ausgehende Pakete in Richtung Supplicant unterliegen manchmal gänzlich anderen Bedingungen als eingehende Pakete vom Supplicant. Die Intention des IEEE 802.1X Standards liegt in der Einschränkung des Zugriffs von Netzwerkkomponenten auf ein Netzwerk. Diese Kommunikationsrichtung unterliegt immer der Port-basierten Zugangskontrolle – selbstverständlich abhängig von den Zustandswerten *AuthControlledPortStatus* und *AuthControlledPortControl*. Die Einschränkung des Zugriffs vom Netzwerk auf die Komponente liegt jedoch nicht im Fokus des 802.1X Standards. Dementsprechend kann diese Kommunikationsrichtung auch unabhängig konfiguriert werden. Zu diesem Zweck kann der Parameter *AdminControlledDirections* auf einen der beiden folgenden Werte gesetzt werden:

1. *Both*: Zugangskontrolle findet symmetrisch in beiden Richtungen statt.
2. *In*: Zugangskontrolle findet nur in eingehender Richtung vom Supplicant zum Authenticator statt.

7.3 Authentifizierung

Die 802.1X Authentifizierung verwendet als Basis das Extensible Authentication Protocol (EAP), welches verschiedene Authentifizierungsmethoden unterstützt. Da ein neu zu einem Netzwerk hinzugefügter Supplicant zuerst eine Authentifizierung durchführen muss, bevor er Zugriff auf das Netzwerk mit dessen höheren Protokollen erhält, kann eine direkte Kommunikation zwischen Supplicant und Authentication Server nicht unbedingt stattfinden. In dem häufigen Fall, dass sich Authenticator und Authentication Server nicht im selben Netzwerkgerät befinden, muss diese Kommunikation anders erfolgen. Der IEEE 802.1X Standard definiert für diesen Zweck das EAP over LAN (EAPOL) Protokoll, welches auf der Sicherungsschicht des OSI Referenzmodells arbeitet. Die Authentifizierung erfolgt dann über den Authenticator als Vermittler und Pakettransformierer.

7.3.1 EAP

Das im 802.1X Standard verwendete Authentifizierungsprotokoll EAP ist in RFC 2284 [19] bzw. in RFC 3748 [1] definiert. Mögliche Varianten für die Authentifizierung können Kombinationen aus Benutzername und Passwort, Einmal-Passwörter oder auch Zertifikate enthalten. Beispiele für verschiedene EAP Methoden sind das Zertifikat-basierte Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) [2] oder das Extensible Authentication Protocol Tunneled Transport Layer Security (EAP-TTLS) [73] zur Unterstützung einer klassischen Authentifizierung mittels Benutzernamen und Passwort über einen verschlüsselten Kanal. Zusätzlich sind verschiedene herstellerabhängige Authentifizierungsmethoden im Umlauf, bei denen der Versuch einer Standardisierung bislang nicht erfolgreich war:

- Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)
- Extensible Authentication Protocol – Generic Token Card (EAP-GTC)
- Extensible Authentication Protocol – Message Digest Number 5 (EAP-MD5)
- Extensible Authentication Protocol – Microsoft Challenge Handshake Authentication Protocol Version 2 (EAP-MSCHAPv2)
- Extensible Authentication Protocol – One Time Password (EAP-OTP)
- Extensible Authentication Protocol – Subscriber Identification Module (EAP-SIM)
- Protected Extensible Authentication Protocol (PEAP)
- Lightweight Extensible Authentication Protocol (LEAP)

Welche der vielen Methoden zu verwenden ist, wird vom 802.1X Standard nicht vorgegeben. Dies gilt in gleichem Maße für den zu verwendenden Authentication Server. Zwar wird im 802.1X Standard der Remote Authentication

Dial In User Service (RADIUS) Server als Beispiel genannt, es können jedoch auch andere Authentication, Authorization, and Accounting (AAA) Server wie beispielsweise Diameter [26] oder TACACS[3] [69] verwendet werden.

Das EAP Protokoll lässt sich auf vier verschiedene Pakettypen reduzieren, mit denen eine Authentifizierung von Supplicants durchgeführt werden kann:

1. *Request*: Der Authentication Server fordert den im *Request* spezifizierten Authentifizierungsparameter vom Supplicant an.
2. *Response*: Der Supplicant antwortet auf einen *Request* des Authentication Servers.
3. *Success*: Nach der Authentifizierung informiert der Authentication Server den Supplicant über die erfolgreiche Beendigung des Authentifizierungsvorgangs.
4. *Failure*: Nach der Authentifizierung informiert der Authentication Server den Supplicant über die erfolglose Beendigung des Authentifizierungsvorgangs.

7.3.2 EAPOL

Zum Transport von EAP Paketen über die Sicherungsschicht des OSI Referenzmodells existieren verschiedene Kapselungen von EAP in anderen Datagrammen. Neben der wohl bekanntesten Kapselung im Point-to-Point Protocol (PPP) [195] definiert der 802.1X Standard eine Kapselung in IEEE 802 LAN Pakete mit dem Namen EAPOL. Jedes EAPOL Paket besteht dabei aus fünf verschiedenen Bereichen:

- Angabe des Ethernet Typs (beispielsweise Ethernet oder Token Ring)
- Version des verwendeten EAP Protokolls (zur Zeit immer gleich „1“)
- EAPOL Pakettyp (wie unten definiert)
- Länge des Datenbereichs
- Datenbereich mit den Nutzdaten.

Zur differenzierten Kommunikation des Supplicants mit dem Authentication Server werden insgesamt fünf verschiedene Typen von EAPOL Paketen definiert.

EAP-Paket

Ein EAPOL Paket vom Typ „EAP-Paket“ enthält in seinem Datenbereich ein vollständiges EAP Paket. Ein Authenticator, der mit einem Authentication

[3] Die Bedeutung der Abkürzung TACACS ist nicht mehr zweifelsfrei zu rekonstruieren. Sie geht aber vermutlich auf den Ausdruck ‚Terminal Access Controller Access Control System‘ zurück. Dies erklärt auch den etwas ungewöhnlichen Titel der RFC 1492 [69], welche den TACACS Dienst beschreibt: ‚An Access Control Protocol, Sometimes Called TACACS‘

Server über das EAP Protokoll kommuniziert und gleichzeitig zum Supplicant hin das EAPOL Protokoll verwendet, kapselt die EAP Pakete des Authentication Servers in EAPOL Pakete für den Supplicant.

EAP-Start

Wenn der Supplicant eine Authentifizierung initiieren will, sendet er ein EAPOL Paket vom Typ „EAP-Start" an den Authenticator, der anschließend die Authentifizierung nach dem EAP Schema durchführt.

EAP-Logoff

Möchte ein Supplicant explizit die authentifizierte Verbindung zum Netzwerk beenden, so sendet er ein EAPOL Paket des Typs „EAP-Logoff". Der Authenticator nimmt dieses Paket entgegen und setzt daraufhin den Status des kontrollierten Ports auf *Unauthorized*.

EAP-Schlüssel

In einem IEEE 802.1X kontrollierten Netzwerk kann mit einem EAPOL Paket vom Typ „EAP-Schlüssel" ein Schlüssel von einem Authenticator zum entsprechenden Supplicant verschickt werden. Der Austausch von Schlüsseln ist typischerweise nur bei einer Verbindung zum Netzwerk über ein unsicheres Medium notwendig. Deshalb werden EAPOL Pakete des Typs „EAP-Schlüssel" vorrangig in 802.11 WLAN Umgebungen eingesetzt.

EAP-gekapselter ASF-Alarm

Alerting Standard Formats (ASF) [70] Nachrichten wie SNMP Nachrichten (Traps) spielen eine wichtige Rolle im Netzwerkmanagement. Die Aufgaben des Netzwerkmanagements beinhalten auch die Überwachung und Kontrolle von Endgeräten im Netzwerk. Durch einen Port-basierten Zugriffsschutz in der Sicherungsschicht des OSI Referenzmodells ist jedoch eine ständige Kommunikationsmöglichkeit zwischen den Managementstationen und den Supplicants nicht gewährleistet. Kann sich ein Supplicant nicht erfolgreich am Authenticator authentifizieren, so wird jegliche Kommunikation zwischen diesen beiden Geräten durch den 802.1X Mechanismus unterbunden – mit Ausnahme der EAPOL Pakete. Damit in einem solchen Fall das Netzwerkmanagement weiterhin funktionieren kann, müssen die dazu notwendigen Pakete den einzigen zur Verfügung stehenden Mechanismus der EAPOL Pakete nutzen. Die entsprechenden Pakete haben den Typ „EAP-gekapselter ASF-Alarm" und erlauben unabhängig vom Authentifizierungsstatus der Verbindung eine Management-Kommunikation in beiden Richtungen.

Fernwartung. Bei der Fernwartung benötigt eine Managementstation eingeschränkten Zugriff auf ein Endgerät im Netzwerk. Mittels der in EAPOL gekapselten ASF Nachrichten kann das Netzwerkmanagement beispielsweise Funktionen wie „Wake-On-LAN" durchführen, bei denen keine authentifizierte Verbindung des Supplicant zum Netzwerk notwendig ist.

Fehlerbenachrichtigung. Die Fehlerbenachrichtigung erfordert den Versand von EAPOL gekapselten ASF Nachrichten vom Supplicant an die Managementstation, damit das Endgerät auch ohne authentifizierte Verbindung zum Netzwerk erfolgreich einen Alarm senden kann. Ein konkretes Beispiel für ein derartiges EAPOL Paket wäre eine gekapselte SNMP Nachricht (Trap) des Supplicant. Ohne authentifizierte Verbindung wird eine vom Endgerät generierte SNMP Nachricht vom Authenticator verworfen und nicht an das Netzwerkmanagement weitergeleitet. Damit die Nachricht dennoch garantiert das Netzwerkmanagement erreicht, kann ein Supplicant zusätzlich zum SNMP Trap noch eine EAPOL gekapselte ASF Nachricht versenden. Erhält ein Authenticator eine derartige ASF Nachricht über EAPOL von einem Supplicant, so muss er das Paket entkapseln und die erhaltene Nachricht an das Netzwerkmanagement weiterleiten[4].

7.4 IEEE 802.1X MIB

Im IEEE 802.1X Standard sind neben der Funktionalität zur Authentifizierung auch die Anforderungen an das Management beschrieben, die von 802.1X-fähigen Geräten zu unterstützen sind. Explizit wird dort auch eine 802.1X-MIB spezifiziert, über welche die verschiedenen Managementaufgaben mit Hilfe von SNMP zu implementieren sind. Abbildung 7.3 verdeutlicht die Stelle der 802.1X-MIB im globalen SNMP Objektbaum. Zum Vergleich sind auch die Position der MIB-I und MIB-II sowie die Position der privaten MIBs abgebildet.

Die 802.1X-MIB unterteilt sich noch einmal in drei unabhängige Unterbäume, die allgemeine Objekte von 802.1X Systemen oder spezielle Objekte von Authenticator und Supplicant Systemen beinhalten. Abbildung 7.4 veranschaulicht die innere Struktur der 802.1X-MIB mit den drei Untergruppen.

[4] Die Weiterleitung der ASF Nachrichten wird vom IEEE 802.1X Standard nicht zwingend vorgeschrieben. Da der Authenticator den Authentifizierungsstatus der aktuellen Verbindung zum Supplicant kennt, kann er im Zustand *authenticated* das EAPOL Paket verwerfen. Die Information erreicht dennoch das Netzwerkmanagement, da der Supplicant parallel zum EAPOL Paket die eigentliche ASF Nachricht erfolgreich versenden kann.

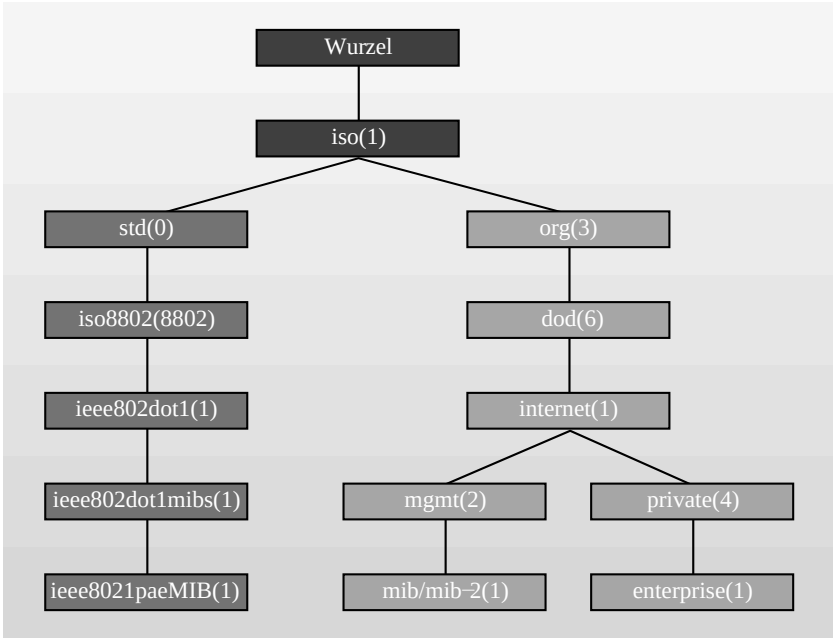

Abb. 7.3. Position der 802.1X-MIB im globalen SNMP Objekt Baum im Vergleich zu den Standard MIBs MIB-I und MIB-II sowie den privaten MIBs.

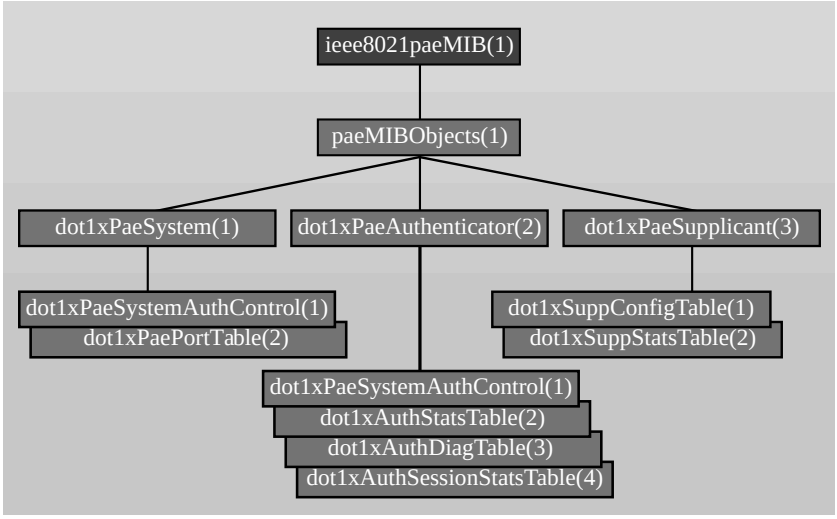

Abb. 7.4. Struktur der 802.1X-MIB mit den drei Unterbäumen für Systeme, Authenticator Systemen und Supplicants.

7.4.1 Allgemeiner MIB-Zweig für alle IEEE 802.1X Systeme

Der *dot1xPaeSystem* Zweig der IEEE 802.1X-MIB befindet sich an der Position *paeMIBObjects.1*. Der System Zweig enthält zwei Objekte, über die globale Angaben zum 802.1X System ausgelesen und gesetzt werden können.

***dot1xPaeSystemAuthControl* (*dot1xPaeSystem.1*).** Der Parameter *dot1xPaeSystemAuthControl* kann einen der beiden gültigen Werte *enabled* oder *disabled* annehmen. Über diesen Parameter lässt sich global einstellen, ob das System überhaupt den IEEE 802.1X Standard verwendet. Steht der Wert auf *disabled*, so wird der Zustand der Authentifizierung aller Authenticator Ports auf *authenticated* gesetzt und eine Kommunikation zwischen Supplicant und Authenticator ist ohne Zugangskontrolle möglich.

***dot1xPaePortTable* (*dot1xPaeSystem.2*).** Beinhaltet eine Tabelle mit Angaben zu den einzelnen Ports des Systems. Die Tabelle ermöglicht sowohl das Auslesen der von den einzelnen Ports unterstützten 802.1X Rollen als auch das Administrieren der Ports. Die fünf Untereinträge unterhalb des Objekts *dot1xPaePortEntry* (*dot1xPaePortTable.1*) für die Spaltendefinitionen lauten:

- *dot1xPaePortNumber(1)* Index des zu dieser Zeile gehörenden Ports.
- *dot1xPaePortProtocolVersion(2)* EAPOL Protokoll Version, die von diesem Port verwendet wird.
- *dot1xPaePortCapabilities(3)* Angaben darüber, welche der beiden 802.1X Rollen Authenticator oder Supplicant dieser Port übernehmen kann.
- *dot1xPaePortInitialize(4)* Kontrollparameter zur Neuinitialisierung des zugehörigen Ports.
- *dot1xPaePortReauthenticate(5)* Kontrollparameter, mit dessen Hilfe sich eine Reauthentifizierung des Supplicant an diesem Port erzwingen lässt.

7.4.2 MIB-Zweig für Authenticator Systeme

Der zweite Unterzweig *dot1xPaeAuthenticator* der 802.1X-MIB zielt speziell auf Authenticator Systeme und befindet sich an der Position *paeMIBObjects.2*. Die vier in dieser Gruppe vorhandenen Tabellen ermöglichen eine detaillierte Überwachung und Konfiguration der einzelnen Authenticator Ports.

***dot1xAuthConfigTable* (*dot1xPaeAuthenticator.1*).** Beinhaltet eine Tabelle mit Angaben zu den einzelnen Ports des Authenticator Systems. Unterhalb des Objekts *dot1xAuthConfigEntry* für die Spaltendefinition befinden sich die insgesamt 14 Einträge für jeden einzelnen Authenticator.

- *dot1xAuthPaeState(1)* Status des virtuellen Authenticator Zustandsautomaten, welcher die Werte *initialize(1)*, *disconnected(2)*, *connecting(3)*, *authenticating(4)*, *authenticated(5)*, *aborting(6)*, *held(7)*, *forceAuth(8)* und *forceUnauth(9)* annehmen kann.

- *dot1xAuthBachendAuthState(2)* Status des virtuellen Backend Authentication Server Zustandsautomaten. Mögliche Werte sind *request(1)*, *response(2)*, *success(3)*, *fail(4)*, *timeout(5)*, *idle(6)* und *initialize(7)*.
- *dot1xAuthAdminControlledDirections(3)* Status der administrativ vorgegebenen richtungsabhängigen Zugriffskontrolle des Ports mit den möglichen Werten *both(0)* oder *in(1)*.
- *dot1xAuthOperControlledDirections(4)* operativer Status der richtungsabhängigen Zugriffskontrolle des Ports mit den möglichen Werten *both(0)* oder *in(1)*. Ein Unterschied zum administrativen Parameter kann sich nur ergeben, wenn *dot1xAuthAdminControlledDirections* auf *in* gesetzt ist, am Port hängt ein weiteres 802.1X-fähiges Gerät und es ist wechselseitige Authentifizierung konfiguriert.
- *dot1xAuthAuthControlledPortStatus(5)* Zustand der Authentifizierung des kontrollierten Ports dieses Authenticators. Mögliche Werte sind *authorized(1)* oder *unauthorized(2)*.
- *dot1xAuthAuthControlledPortControl(6)* Aktueller Port Status mit den drei möglichen Werten *forceUnauthorized(1)*, *auto(2)* und *forceAuthorized(3)*.
- *dot1xAuthQuietPeriod(7)* Zeitspanne in Sekunden nach einem fehlgeschlagenen Authentifizierungsversuch des angeschlossenen Supplicants, in welcher der Authenticator keine weiteren EAPOL Pakete vom Supplicant entgegennimmt. Auf diese Weise lassen sich beispielsweise DoS Attacken eingrenzen.
- *dot1xAuthTxPeriod(8)* Zeitspanne in Sekunden, nach der ein Authenticator ein erneutes EAP-Paket vom Typ *Request* an den Supplicant sendet, falls dieser nicht auf den vorhergehenden *Request* geantwortet hat.
- *dot1xAuthSuppTimeout(9)* Zeitspanne in Sekunden, auf welche der Authenticator auf Antworten vom Supplicant wartet, bevor er seine letzte Nachricht wiederholt und erneut versendet.
- *dot1xAuthServerTimeout(10)* Zeitspanne in Sekunden, auf welche der Authenticator auf Antworten vom Authentication Server wartet, bevor er seine letzte Nachricht wiederholt und erneut versendet.
- *dot1xAuthMaxReq(11)* Liefert die Anzahl der Versuche, die ein Authenticator zur Initialisierung eines Authentifizierungsvorgangs des Supplicants unternimmt.
- *dot1xAuthReauthPeriod(12)* Zeitspanne in Sekunden nach der letzten erfolgreichen Authentifizierung, nach welcher der Authenticator eine Reauthentifizierung initiiert.
- *dot1xAuthReauthEnabled(13)* Gibt an, ob nach der im obigen Parameter *dot1xAuthReauthPeriod* angegebenen Zeitspanne eine Reauthentifizierung des Supplicants notwendig ist.
- *dot1xAuthKeyTxEnabled(14)* Gibt an, ob der Austausch von Schlüsseln zur geschützten Kommunikation mittels EAPOL Paketen erlaubt ist.

***dot1xAuthStatsTable* (*dot1xPaeAuthenticator.2*).** Beinhaltet eine Tabelle mit statistischen Daten zu jedem Authenticator Port des Systems. Über die zwölf Untereinträge unterhalb des Objekts *dot1xAuthStatsEntry* (*dot1xAuthStatsTable.1*) für die Spaltendefinitionen können die gesammelten Werte ausgelesen werden.

- *dot1xAuthEapolFramesRx(1)* Anzahl aller vom Authenticator empfangenen gültigen EAPOL Pakete.
- *dot1xAuthEapolFramesTx(2)* Anzahl aller vom Authenticator gesendeten EAPOL Pakete.
- *dot1xAuthEapolStartFramesRx(3)* Anzahl aller vom Authenticator empfangenen gültigen EAPOL Pakete vom Typ EAP-Start.
- *dot1xAuthEapolLogoffFramesRx(4)* Anzahl aller vom Authenticator empfangenen gültigen EAPOL Pakete vom Typ EAP-Logoff.
- *dot1xAuthEapolRespIdFramesRx(5)* Anzahl aller vom Authenticator empfangenen gültigen EAP-Response Pakete, welche jeweils die Identität eines Supplicants angeben.
- *dot1xAuthEapolRespFramesRx(6)* Anzahl aller vom Authenticator empfangenen gültigen EAP-Response Pakete, welche nicht die Identität eines Supplicants angeben.
- *dot1xAuthEapolReqIdFramesTx(7)* Anzahl aller vom Authenticator gesendeten EAP-Request Pakete, welche die Identität eines Supplicants erfragen.
- *dot1xAuthEapolReqFramesTx(8)* Anzahl aller vom Authenticator gesendeten EAP-Request Pakete, welche nicht die Identität eines Supplicants erfragen.
- *dot1xAuthInvalidEapolFramesRx(9)* Anzahl aller vom Authenticator empfangenen EAPOL Pakete, die einen ungültigen Typ aufweisen.
- *dot1xAuthEapLengthErrorFramesRx(10)* Anzahl aller vom Authenticator empfangenen EAPOL Pakete mit einer ungültigen Angabe für die Paketlänge.
- *dot1xAuthLastEapolFrameVersion(11)* Version des zuletzt empfangenen EAPOL Paketes.
- *dot1xAuthLastEapolFrameSource(12)* MAC Quelladresse des zuletzt empfangenen EAPOL Paketes.

***dot1xAuthDiagTable* (*dot1xPaeAuthenticator.3*).** Enthält eine Tabelle mit statistischen Daten zum Zustandsautomaten des Authenticators und des Authentication Servers. Über die 18 Untereinträge unterhalb des Objekts *dot1xAuthDiagEntry* (*dot1xAuthDiagTable.1*) für die Spaltendefinitionen können im Wesentlichen Angaben darüber entnommen werden, wie oft sich die Zustände des Zustandsautomaten geändert haben.

- *dot1xAuthEntersConnecting(1)* Anzahl der Übergänge des Authenticator-Zustandsautomaten in den Status *Connecting*.

- *dot1xAuthEapLogoffsWhileConnecting(2)* Anzahl Übergänge des Zustandsautomaten des Authenticators in den Status *Disconnected* durch den Empfang eines EAPOL Paketes vom Typ EAP-Logoff.
- *dot1xAuthEntersAuthenticating(3)* Anzahl Übergänge des Authenticator-Zustandsautomaten in den Status *Authenticating* durch den Empfang eines EAP-Response Paketes, welche die Identität eines Supplicants angibt.
- *dot1xAuthAuthSuccessWhileAuthenticating(4)* Anzahl der Übergänge des Authenticator-Zustandsautomaten in den Status *Authenticated* durch den Empfang eines EAP-Success Paketes vom Authentication Server.
- *dot1xAuthAuthTimeoutsWhileAuthenticating(5)* Anzahl der Übergänge des Authenticator-Zustandsautomaten in den Status *Aborting* durch den Empfang eines EAP Paketes vom Authentication Server, das auf einen Zeitüberlauf bei der Authentifizierung hindeutet.
- *dot1xAuthAuthFailWhileAuthenticating(6)* Anzahl der Übergänge des Zustandsautomaten des Authenticators in den Status *Held* durch den Empfang eines EAP-Fail Paketes vom Authentication Server.
- *dot1xAuthAuthReauthsWhileAuthenticating(7)* Anzahl der Übergänge des Authenticator-Zustandsautomaten in den Status *Aborting* durch den Empfang einer Management Anweisung zur Reauthentifizierung.
- *dot1xAuthAuthEapStartsWhileAuthenticating(8)* Häufigkeit der Übergänge des Authenticator-Zustandsautomaten in den Status *Aborting* durch den Empfang eines EAPOL Paketes vom Typ EAP-Start vom Supplicant.
- *dot1xAuthAuthEapLogoffWhileAuthenticating(9)* Häufigkeit der Übergänge des Authenticator-Zustandsautomaten in den Status *Aborting* durch den Empfang eines EAPOL Paketes vom Typ EAP-Logoff vom Supplicant.
- *dot1xAuthAuthReauthsWhileAuthenticated(10)* Anzahl der Übergänge des Authenticator-Zustandsautomaten in den Status *Connecting* durch den Empfang einer Management Anweisung zur Reauthentifizierung.
- *dot1xAuthAuthEapStartsWhileAuthenticted(11)* Anzahl der Übergänge des Authenticator-Zustandsautomaten in den Status *Connecting* durch den Empfang eines EAPOL Paketes vom Typ EAP-Start vom Supplicant.
- *dot1xAuthAuthEapLogoffWhileAuthenticted(12)* Anzahl der Übergänge des Authenticator-Zustandsautomaten in den Status *Disconnected* durch den Empfang eines EAPOL Paketes vom Typ EAP-Logoff vom Supplicant.
- *dot1xAuthBackendResponses(13)* Anzahl der Versuche zur Kontaktaufnahme mit dem Authentication Server, die vom Authenticator durch Senden eines initialen Paketes unternommen worden sind.
- *dot1xAuthBackendAccessChallenges(14)* Anzahl der erfolgreich mit dem Authentication Server aufgebauten Verbindungen, die durch den Empfang eines Antwortpaketes angezeigt werden.
- *dot1xAuthBackendOtherRequestsToSupplicant(15)* Anzahl der vom Authentication Server gesendeten EAP-Request Pakete, welche dem Aushandeln der zu verwendenden EAP Methode gedient haben.

- *dot1xAuthBackendNonNakResponsesFromSupplicant(16)* Anzahl der positiven Antworten des Supplicant auf EAP-Request Pakete des Authentication Servers, welche dem Aushandeln der zu verwendenden EAP Methode gedient haben.
- *dot1xAuthBackendAuthSuccesses(17)* Anzahl der vom Authentication Server gesendeten EAP-Success Pakete, die eine erfolgreiche Authentifizierung des Supplicant anzeigen.
- *dot1xAuthBackendAuthFails(18)* Anzahl der vom Authentication Server gesendeten EAP-Fail Pakete, die eine erfolglose Authentifizierung des Supplicant anzeigen.

***dot1xAuthSessionStatsTable* (*dot1xPaeAuthenticator.4*).** Beinhaltet eine Tabelle mit statistischen Daten zu den Verbindungen jedes im System vorhandenen Authenticators. Über die neun Untereinträge unterhalb des Objekts *dot1xAuthSessionStatsEntry* (*dot1xAuthSessionStatsTable.1*) für die Spaltendefinitionen können die ermittelten Werte ausgelesen werden.

- *dot1xAuthSessionOctetsRx(1)* Anzahl der in dieser Verbindung empfangenen Oktette in Paketen mit Nutzerdaten.
- *dot1xAuthSessionOctetsTx(2)* Anzahl der in dieser Verbindung gesendeten Oktette in Paketen mit Nutzerdaten.
- *dot1xAuthSessionFramesRx(3)* Anzahl der in dieser Verbindung empfangenen Pakete mit Nutzerdaten.
- *dot1xAuthSessionFramesTx(4)* Anzahl der in dieser Verbindung gesendeten Pakete mit Nutzerdaten.
- *dot1xAuthSessionId(5)* Eindeutige Identifikationszeichenkette dieser Verbindung.
- *dot1xAuthSessionAuthenticMethod(6)* Angaben zu der bei dieser Verbindung verwendeten Authentifizierungsmethode.
- *dot1xAuthSessionTime(7)* Dauer dieser Verbindung in Sekunden.
- *dot1xAuthSessionTerminateCause(8)* Angabe für den Grund der Beendigung dieser Verbindung, sofern sie nicht mehr aufgebaut ist. Mögliche Werte sind *supplicantLogoff(1)*, *portFailure(2)*, *supplicantRestart(3)*, *reauthFailed(4)*, *authControlForceUnauth(5)*, *portReInit(6)*, *portdminDisabled(7)* und *notTerminatedYet(999)*.
- *dot1xAuthSessionUserName(9)* Benutzername stellvertretend für die Identität des Supplicants dieser Verbindung.

7.4.3 MIB-Zweig für Supplicants

Der *dot1xPaeSupplicant* Zweig zielt speziell auf Supplicants und rundet somit die 802.1X-MIB ab. Unterhalb des an der Position *1eee.8021paeMIB.1.3* liegendenden Zweiges befinden sich zwei Tabellen, die eine Überwachung und Konfiguration der einzelnen Supplicant Ports ermöglichen.

***dot1xSuppConfigTable* (*dot1xPaeSupplicant.1*).** Enthält eine Tabelle mit Angaben zum Status des Zustandsautomaten des Supplicants, über welche auch administrative Aufgaben durchgeführt werden können. Das Unterobjekt *dot1xSuppConfigEntry* für die Spaltendefinitionen enthält fünf Angaben:

- *dot1xSuppPaeState(1)* Aktueller Status des Zustandsautomaten des Supplicants. Mögliche Werte sind *disconnected(1)*, *logoff(2)*, *connecting(3)*, *authenticating(4)*, *authenticated(5)*, *aquired(6)* und *held(7)*.
- *dot1xSuppHeldPeriod(2)* Zeitspanne in Sekunden nach einem fehlgeschlagenen Authentifizierungsversuch, in welcher der angeschlossene Authenticator keine weiteren EAPOL Pakete entgegennimmt.
- *dot1xSuppAuthPeriod(3)* Zeitspanne in Sekunden, die der Supplicant auf eine Antwort vom Authenticator wartet, bevor er eine erneute Authentifizierung anfordert.
- *dot1xSuppStartPeriod(4)* Zeitspanne in Sekunden, die ein Supplicant auf eine Authentifizierungsanfrage des Authenticators wartet, bevor er selbst aktiv ein EAPOL Paket vom Typ EAP-Start versendet.
- *dot1xSuppMaxStart(5)* Anzahl der vom Supplicant unternommenen Versuche zum Initiieren einer Authentifizierung beim Authenticator, bevor er den Zugang zum Netzwerk als nicht 802.1X-kontrolliert betrachtet.

***dot1xSuppStatsTable* (*dot1xPaeSupplicant.2*).** Beinhaltet eine Tabelle mit statistischen Daten zu jedem Supplicant Port des Systems. Die zwölf Einträge unterhalb des Objekts *dot1xSuppStatsEntry* (*dot1xSuppStatsTable.1*) für die Spaltendefinitionen ermöglichen das Auslesen der gesammelten Werte.

- *dot1xAuthEapolFramesRx(1)* Anzahl aller vom Supplicant empfangenen gültigen EAPOL Pakete.
- *dot1xAuthEapolFramesTx(2)* Anzahl aller vom Supplicant gesendeten EAPOL Pakete.
- *dot1xAuthEapolStartFramesTx(3)* Anzahl aller vom Supplicant gesendeten EAPOL Pakete vom Typ EAP-Start.
- *dot1xAuthEapolLogoffFramesTx(4)* Anzahl aller vom Supplicant gesendeten EAPOL Pakete vom Typ EAP-Logoff.
- *dot1xAuthEapolRespIdFramesTx(5)* Anzahl aller vom Supplicant gesendeten EAP-Response Pakete, welche die eigene Identität angeben.
- *dot1xAuthEapolRespFramesTx(6)* Anzahl aller vom Supplicant gesendeten EAP-Response Pakete, welche nicht die eigene Identität angeben.
- *dot1xAuthEapolReqIdFramesRx(7)* Anzahl aller vom Supplicant empfangenen gültigen EAP-Request Pakete, welche die eigene Identität anfordern.
- *dot1xAuthEapolReqFramesRx(8)* Anzahl aller vom Supplicant empfangenen gültigen EAP-Request Pakete, welche nicht die eigene Identität anfordern.
- *dot1xAuthInvalidEapolFramesRx(9)* Anzahl aller vom Supplicant empfangenen EAPOL Pakete, die einen ungültigen Typ aufweisen.

- *dot1xAuthEapLengthErrorFramesRx(10)* Anzahl der vom Supplicant empfangenen EAPOL Pakete mit einer ungültigen Angabe für die Paketlänge.
- *dot1xAuthLastEapolFrameVersion(11)* Version des zuletzt empfangenen EAPOL Paketes.
- *dot1xAuthLastEapolFrameSource(12)* MAC Quelladresse des zuletzt empfangenen EAPOL Paketes.

8

Andere Kommunikationsformen und -wege des Netzwerkmanagements

Wie bereits in Kapitel 1 definiert, besteht Netzwerkmanagement sowohl aus der Netzwerkkonfiguration als auch aus der Netzwerküberwachung. Das in Kapitel 4 näher beschriebene Simple Network Management Protocol (SNMP) erfüllt beide Aufgaben gleichermaßen gut. Es lassen sich sowohl umfangreiche Statusinformationen eines einzeln Hosts abfragen als auch Konfigurationseinstellungen auf dem Host vornehmen. Die Aufgaben der Netzwerküberwachung stellen im Normalfall den weitaus größeren Anteil am Netzwerkmanagement. Je größer ein zu überwachendes Netzwerk wird, desto schwieriger wird eine vollständige Überwachung aller Netzwerkkomponenten von einer einzigen zentralen Managementstation aus. Eine Verteilung der Überwachungsfunktionen auf mehrere Managementstationen hilft in großem Maße dabei, dieses Problem zu entschärfen. Eine speziell zu diesem Zweck entwickelte Lösung ist das Remote Monitoring (RMON), das in RFC 2819 [218] definiert ist. Andere Techniken erlauben ebenfalls die Überwachung einer größeren Anzahl an Netzwerkgeräten. Beispielsweise können über den Unix SYSLOG Mechanismus Informationen aller Art zeitnah an einen zentralen Log-Server übermittelt werden. Der SYSLOG Mechanismus erlaubt darüber hinaus sogar das unmittelbare Auslösen beliebiger Aktionen beim Eintreffen spezieller Log-Meldungen. Aber auch proprietäre Lösungen existieren, die ihre eigenen Transportmechanismen entwickelt haben. Dieses Kapitel soll eine kurze Beschreibung zu den neben SNMP bestehenden Werkzeugen abliefern.

8.1 RMON

Das Remote Network Monitoring (RMON) stellt ein Werkzeug im Zusammenhang von SNMP dar, mit dem sich vorrangig statistische Informationen über ein Netzwerk zentral sammeln und aufbereiten lassen, um sie anschließend an eine Netzwerkmanagementstation weiterzuleiten. Zu diesem Zweck wird ein Netzwerkmonitor wie eine Sonde direkt im zu überwachenden Netzwerk platziert. Dort kann der Netzwerkmonitor verschiedene Statistiken nicht nur

über ein einzelnes Gerät, sondern über das gesamte Teilnetzwerk sammeln. Dies ist auch dann noch möglich, wenn durch normale oder unvorhergesehene Umstände die entfernt stehende Netzwerkmanagementstation vorübergehend nicht mit dem Netzwerk verbunden ist. Der Netzwerkmonitor sammelt in dieser Zeit ungehindert weiter die gewünschten Daten. Sobald die Netzwerkmanagementstation wieder erreichbar ist, kann sie die gesammelten Daten in zusammengefasster Form beim RMON Monitor abrufen. Typisches Einsatzgebiet eines RMON Netzwerkmonitors sind große Netzwerke. Um den aus der Überwachung der einzelnen Komponenten resultierenden Netzwerkverkehr nicht vollständig an eine zentrale Managementstation übermitteln zu müssen, was mit einer messbaren Beeinträchtigung des Netzwerkes verbunden wäre, können einzelne Netzwerkabschnitte mit RMON Sensoren ausgerüstet werden. Vorteilhaft ist dies vor allem dann, wenn das Teilnetzwerk nicht ständig mit dem zentralen Managementnetzwerk verbunden ist, beispielsweise bei einer Wählverbindung. Aber auch bei einer Festverbindung kann die Netzwerkanbindung unplanmäßig unterbrochen werden, so dass die in diesem Zeitraum anfallenden statistischen Daten ohne RMON Sensoren verloren gehen würden.

Da der RMON Netzwerkmonitor speziell für die Aufgabe der Informationssammlung abgestellt ist, kann er die erfassten Daten vor der Weitergabe an die Netzwerkmanagementstation aufbereiten und zusammenfassen. So kann die Managementstation – ohne eigene umfangreiche Berechnungen und Datenauswertungen durchführen zu müssen – direkt den RMON Sensor nach Geräten befragen, die für einen bestimmten statistischen Wert die Spitzenreiter sind. Ein Beispiel wäre dasjenige Gerät, welches den meisten Netzwerkverkehr erzeugt oder empfängt.

Ein weiterer Vorteil von RMON ist die Tatsache, dass im überwachten Teilnetz nur ein einziger Sensor die Daten sammeln muss, die anschließend jedoch an beliebig viele Netzwerkmanagementstationen weitergegeben werden können. Dadurch wird verhindert, dass eine Aufteilung des Netzwerkmanagements auf mehrere Geräte einen nachteiligen Einfluss auf die Datenkonsistenz hat. Da die Daten außerdem im RMON Monitor bereits zusammengefasst und ausgewertet werden können, wird gleichzeitig der durch das Netzwerkmanagement verursachte zu übertragende Netzwerkverkehr erheblich reduziert.

RMON ist eine Erweiterung des SNMP Rahmenwerkes. Genauer gesagt ist RMON eine MIB, die von den RMON Sensoren unterstützt wird. Mittlerweile ist ein größeres Rahmenwerk um die beiden in RFC 2819 [218] und RFC 2021 [217] definierten Versionen RMONv1 und RMONv2 entstanden. RFC 3577 [221] liefert einen Überblick über die verschiedenen RFCs und Standards, die zum Thema Remote Monitoring existieren. Der *rmon* Unterzweig des hierarchischen MIB Baums befindet sich an der Stelle *mib-2.16* (*1.3.6.1.2.1.16*). RFC 3737 [226] enthält einen statischen Ausschnitt aller zur Zeit der Definition dieser RFC im April 2004 definierten OIDs, die auch in Tabelle 8.1 aufgelistet sind.

Tabelle 8.1. IN RFC 3737 definierte Remote Network Monitoring OIDs und deren Herkunft.

RMON OID	Name	RFC
rmon.0	rmonEventsV2	RFC 2819 [218]
rmon.1	statistics	RFC 2819
rmon.2	history	RFC 2819
rmon.3	alarm	RFC 2819
rmon.4	hosts	RFC 2819
rmon.5	hostTopN	RFC 2819
rmon.6	matrix	RFC 2819
rmon.7	filter	RFC 2819
rmon.8	capture	RFC 2819
rmon.9	event	RFC 2819
rmon.10	tokenRing	RFC 1513 [216]
rmon.11	protocolDir	RFC 2021 [217]
rmon.12	protocolDist	RFC 2021
rmon.13	addressMap	RFC 2021
rmon.14	nlHost	RFC 2021
rmon.15	nlMatrix	RFC 2021
rmon.16	alHost	RFC 2021
rmon.17	alMatrix	RFC 2021
rmon.18	usrHistory	RFC 2021
rmon.19	probeConfig	RFC 2021
rmon.20	rmonConformance	RFC 2021
rmon.21	mediaIndependentStats	RFC 3273 [219]
rmon.22	switchRMON	RFC 2613 [224]
rmon.23	apm	RFC 3729 [220]
rmon.26	dsmonMIB	RFC 3287 [12]
rmon.27	interfaceTopNMIB	RFC 3144 [180]
rmon.29	hcAlarmMIB	RFC 3434 [16]

8.1.1 RMONv1

Die in RFC 2819 [218] näher beschriebene RMONv1 MIB enthält insgesamt neun verschiedene Gruppen von Informationen, die sich mit einem Netzwerkmonitor ermitteln lassen. Eine Erweiterung der vorrangig auf Ethernet ausgerichteten RMON MIB findet sich in RFC 1513 [216]. Dort werden sowohl ergänzende Tabellen zu den vorhandenen Gruppen aus *rmon* als auch eine neue zehnte Gruppe für IEEE 802.5 Token Ring Netzwerke definiert.

statistics (*rmon.1*)

Die *statistics* Gruppe besteht ursprünglich nur aus der einzigen Tabelle *etherStatsTable*, die jeweils eine Zeile pro Ethernet-Schnittstelle des Gerätes enthält. Zu jeder Schnittstelle werden dann insgesamt 21 verschiedene Informa-

tionen gesammelt, beispielsweise die Anzahl an Paketen, Oktetten, Multicast-Paketen, Broadcast-Paketen, Paketkollisionen oder auch die Anzahl an Paketen, die nicht mehr als 64 Oktette enthalten haben. Durch die Token Ring Erweiterungen aus RFC 1513 werden zwei zusätzliche Tabellen ergänzt. In der Tabelle *tokenRingMLStatsTable* werden grundsätzlich ähnliche Daten zu den beobachteten Paketen gesammelt. Es werden jedoch einige zusätzliche Statistiken erstellt, die typisch für Token Ring Netzwerke sind. Der Unterschied zur dritten Tabelle *tokenRingPStatsTable* liegt lediglich in der Art und Weise, die zur Erfassung der Pakete verwendet wird. In die Tabelle *tokenRingMLStatsTable* werden nur Pakete eingetragen, die über die Media Access Control (MAC) Adresse aus der Sicherungsschicht des OSI Referenzmodells der jeweiligen Netzwerkschnittstelle gesammelt wurden. In der Tabelle *tokenRingMLStatsTable* landen Pakete, die über den promiskuitiven Modus beobachtet wurden, der alle Pakete erfasst.

history (*rmon.2*)

In der *history* Gruppe befinden sich zwei Tabellen, die statistische Informationen aus vergangenen Zeiträumen enthalten. Die Tabelle *historyControlTable* enthält Informationen, an welcher Schnittstelle und für welche Zeiträume jeweils Messungen durchzuführen sind. In der *etherHistoryTable* Tabelle stehen dann die Messwerte der einzelnen Schnittstellen und Messzeiträume zur Verfügung. Die Art der gemessenen Werte ist ähnlich zur *statistics* Gruppe aber nicht identisch. So werden zwar auch Werte für die Anzahl an Paketen, Oktetten, Multicast-Paketen oder Broadcast-Paketen ermittelt und gespeichert, die Unterscheidung der gemessenen Pakete nach ihrer Größe wird aber nicht vorgenommen. Zusätzlich zu diesen beiden in RFC 2819 definierten Tabellen existieren noch die durch RFC 1513 ergänzten Tabellen *tokenRingMLHistoryTable* und *tokenRingPHistoryTable*. Wie schon in der *statistics* Gruppe werden hier konsequente Erweiterungen der vorhandenen Ethernet-bezogenen Tabellen vorgenommen. Der Unterschied zwischen den beiden Token Ring Tabellen liegt wieder in der Art der Datenerfassung über die MAC Adresse oder im promiskuitiven Modus der Netzwerkschnittstelle.

alarm (*rmon.3*)

Um beim Beobachten von bestimmten konfigurierbaren Ereignissen eine Netzwerkmanagementstation benachrichtigen zu können, sind im RMON Monitor verschiedene Sensoren untergebracht, welche das Überschreiten oder das Unterschreiten von Schwellwerten einzelner Messwerte identifizieren können. Die Sensoren werden über die *alarmTable* Tabelle konfiguriert, welche das einzige Objekt in der *alarm* Gruppe ist. Zu den konfigurierbaren Attributen gehören das Messintervall genauso wie der obere und der untere Schwellwert.

hosts (*rmon.4*)

In der *host* Gruppe sammelt ein RMON Monitor Informationen über Hosts, die im Netzwerk bekannt werden. Die Informationen der einzelnen Komponenten werden abhängig von der Schnittstelle des RMON Monitors gesammelt. Jeder neu entdeckte Host wird parallel in die beiden Tabellen *hostTable* und *hostTimeTable* eingetragen. Zu den gemessenen Statistiken zählt beispielsweise die Anzahl der ein- und ausgehenden Pakete und Oktette zu und von diesem Host oder auch die Anzahl der fehlerhaften Pakete von diesem Gerät. Der einzige Unterschied zwischen den beiden Tabellen *hostTable* und *hostTimeTable* liegt in der Sortierung. Während die *hostTable* nach MAC Adressen sortiert ist, ist die *hostTimeTable* nach den Zeitpunkten sortiert, zu denen die einzelnen Netzwerkkomponenten bekannt wurden. Die dritte in der *hosts* Gruppe enthaltene Tabelle *hostControlTable* steuert, an welchen Schnittstellen die Messungen vorgenommen werden sollen und wie groß die *hostTable* und *hostTimeTable* maximal werden dürfen, bevor Einträge aus der Tabelle zwangsweise gelöscht werden.

hostTopN (*rmon.5*)

Die *hostTopN* Gruppe zeichnet hauptsächlich für das Erstellen von „Top-Ten"-Listen verantwortlich. Dabei ist der Begriff „Top-Ten" nicht wörtlich zu nehmen, denn die Anzahl der in den verschiedenen Listen enthaltenen Einträge ist variierbar. Diese und andere Einstellungen für die Messwertnahme können in der *hostTopNControlTable* vorgenommen werden. Die einzelnen Listen werden dann in der zweiten Tabelle dieser Gruppe gespeichert. Mögliche Messwertgrößen, für die in der *hostTopNTable* Statistiken der einzelnen Hosts hinterlegt werden, sind ähnlich zu den Statistiken in der *history* Gruppe.

matrix (*rmon.6*)

In der *matrix* Gruppe werden Statistiken zu einzelnen Kommunikationsbeziehungen gesammelt. Für jedes Paar aus einem Sender und einem Empfänger werden parallel in den beiden Tabellen *matrixSDTable* und *matrixDSTable* die Anzahl an Paketen, Oktetten und fehlerhaften Paketen zur Kommunikation der beiden Geräte gespeichert. Die beiden Tabellen unterscheiden sich nur nach ihrer Indizierung. Während die *matrixSDTable* nach Source (Sender) und Destination (Empfänger) sortiert sind, werden in der *matrixDSTable* die Einträge umgekehrt sortiert. In der *matrixControlTable* können Konfigurationseinstellungen zur Ermittlung dieser Statistiken durchgeführt werden, beispielsweise die maximale Größe, welche die beiden anderen Tabellen erreichen dürfen.

filter (*rmon.7*)

Die *filter* Gruppe ermöglicht weitaus komplexere statistische Daten zu erheben. Durch die Definition von Filtern in der *filterTable* können sämtliche Pakete ermittelt werden, die einem frei definierbaren Suchmuster entsprechen. Diese Filter ermöglichen also Statistiken über den Inhalt der einzelnen Pakete. Wenn die Netzwerkmanagementstation über das Eintreffen neuer Pakete informiert werden soll, welche diesem Suchmuster entsprechen, so können in der zweiten Tabelle *channelTable* die Rahmenbedingungen für das Versenden einer entsprechenden SNMP Nachricht konfiguriert werden.

capture (*rmon.8*)

Alle Pakete, die in der *filter* Gruppe einem dort definierten Suchmuster entsprechen, können nicht nur gezählt, sondern sogar komplett in der *captureBufferTable* gespeichert werden. In der zweiten Tabelle der *capture* Gruppe mit der Bezeichnung *bufferControlTable* können zusätzlich Konfigurationsangaben über die Speicherung der Pakete gemacht werden. Dies betrifft beispielsweise den zu speichernden Ausschnitt der Pakete oder die maximale Anzahl an Oktetten, die vom Inhalt eines Pakets gespeichert werden können.

event (*rmon.9*)

In der *event* Gruppe schließlich befinden sich die beiden Tabellen *eventTable* und *logTable*, in denen Informationen zu den vom RMON Monitor generierten SNMP Nachrichten und den optional dazu erstellten Log-Meldungen gespeichert werden.

tokenRing (*rmon.10*)

Die sechs Tabellen der *tokenRing* Gruppe werden allesamt durch die RFC 1513 zur bestehenden RMON MIB ergänzt. In der *ringStationTable* werden verschiedene statistische Daten zu allen bekannten Token Ring Hosts gesammelt. Ähnlich der *hosts* Gruppe existiert auch bei Token Ring eine Tabelle, in der Konfigurationsangaben zur Datenermittlung getroffen werden. Eine typische Information, die in der *ringStationControlTable* gespeichert wird, ist die maximal mögliche Größe der *ringStationTable*. Eine Besonderheit von Token Ring ist die Anordnung aller Netzwerkkomponenten in einer ringförmigen Struktur. In der Tabelle *ringStationOrderTable* werden alle bekannten Hosts in der Reihenfolge sortiert, wie sie im Token Ring Netzwerk aufeinander folgen. Informationen, die weniger als Statistiken sondern mehr als Konfigurationsparameter zu bezeichnen sind, werden in der vierten Tabelle gespeichert. Wie die Daten für die *ringStationConfigTable* Tabelle gesammelt werden sollen, lässt sich zusätzlich in der *ringStationConfigControlTable* definieren. Die letzte Tabelle in der *tokenRing* Gruppe bezieht sich speziell auf Source Routing

Bridges. In der Tabelle *sourceRoutingStatsTable* werden beispielsweise Informationen über die Anzahl der Routing-Schritte (Hops) gesammelt, welche die einzelnen Pakete zurückgelegt haben.

8.1.2 RMONv2

Nach den Erfahrungen mit der RMONv1 MIB wurde eine Ergänzung notwendig, um andere sinnvolle und hilfreiche Werte von den RMON Sensoren messen und via SNMP übermitteln zu können. Aus diesem Grund wurde RMONv2 in der RFC 2021 [217] eingeführt, das nicht als Ersatz sondern als konsequente Erweiterung zu RMONv1 zu betrachten ist. Neben der Tatsache, dass in allen bestehenden Tabellen aus RFC 2819 und RFC 1513 zwei zusätzliche Werte hinzugefügt wurden, werden in RMONv2 weitere neun neue Gruppen definiert, die vorrangig das Ermitteln von statistischen Daten aus weiter oben liegenden Schichten des OSI Referenzmodells (bis hin zur Anwendungsschicht) ermöglichen sollen.

protocolDir (*rmon.11*)

Die *protocolDir* Gruppe ist hauptsächlich eingeführt worden, um auch Netzwerkprotokolle höherer Schichten des OSI Referenzmodells messen zu können. Dies betrifft insbesondere die Vermittlungsschicht und die Transportschicht, in denen sich beispielsweise große Teile der TCP/IP Protokollfamilie befinden. Alle unterstützten und erkannten Protokolle werden in der *protocolDirTable* eingetragen. In den Request for Comments RFC 2895 [15], RFC 2896 [14], RFC 3395 [13] und RFC 3919 [65] sind Beispiele für verschiedene gültige Protokolldefinitionen angegeben.

protocolDist (*rmon.12*)

Die Statistiken, die zu den in der *protocolDir* definierten Protokollen gehören, werden in der *protocolDist* Gruppe gehalten. Es lassen sich allerdings nur zwei verschiedene Werte messen; die Tabelle *protocolDistStatsTable* speichert lediglich die Anzahl der gemessenen Pakete und Oktette der jeweiligen Protokolle. Zusätzlich ermöglicht die Tabelle *protocolDistControlTable* eine Konfiguration der zugehörigen Sensoren.

addressMap (*rmon.13*)

In der *addressMap* Gruppe werden die zu physikalischen Adressen gehörenden Netzwerkadressen von allen bekannten Hosts gespeichert. Während die einzelnen Einträge in der *addressMapTable* landen, steht zur Konfiguration der Datensammlung die *addressMapControlTable* zur Verfügung. Außerdem sind in der *addressMap* Gruppe noch separate Zähler implementiert, welche die Anzahl der Ereignisse festhalten, bei denen ein Eintrag der Tabelle hinzugefügt oder aus der Tabelle entfernt wurde.

nlHost (*rmon.14*)

Die Gruppe *nlHost* beinhaltet Statistiken zu Paketen, die zu und von einzelnen Host-Adressen der Netzwerkschicht des OSI Referenzmodells gesendet wurden. Während die Tabelle *nlHostTable* die eigentlichen statistischen Daten enthält, dient die Tabelle *hlHostControlTable* der Konfiguration der jeweiligen Sensoren. Auch in der *nlHost* Gruppe beschränken sich die Statistiken auf jeweils gesendete und empfangene Pakete und Oktette.

nlMatrix (*rmon.15*)

Die Gruppe *nlMatrix* ist vergleichbar mit der Kombination aus den beiden in RMONv1 bereits definierten Gruppen *hostTopN* und *matrix*. Im Unterschied zu den beiden schon vorhandenen Gruppen beziehen sich die in der *nlMatrix* Gruppe gesammelten Daten jedoch auf die Netzwerkschicht des OSI Referenzmodells und nicht auf die physikalischen Adressen der Sicherungsschicht.

Die jeweils zur *matrix* Gruppe vergleichbaren Tabellen sind die *hlMatrixControlTable* mit Konfigurationsangaben zur Sammlung der Statistiken in den beiden Tabellen *nlMatrixSDTable* und *nlMatrixDSTable*. Die beiden letzten Tabellen unterscheiden sich wieder nur durch die Indizierung nach Quell-Ziel-Paaren und Ziel-Quell-Paaren.

Angelehnt an die *hostTopN* Gruppe sind schließlich die beiden Tabellen *nlMatrixTopNControlTable* und *nlMatrixTopNTable*. Auch hier werden „Top-Ten"-Listen zu den Statistiken von Paketen aus der Netzwerkschicht ermittelt und zur Verfügung gestellt.

alHost (*rmon.16*)

Ein direkt zu *nlHost* vergleichbare Gruppe ist die *alHost* Gruppe. Diese enthält nur eine Tabelle, die mit Daten zu Paketen aus der Anwendungsschicht des OSI Referenzmodells gefüllt wird. Wie viele andere Tabellen der RMONv2 MIB auch, macht die *alHostTable* intensiven Gebrauch von der weniger bekannten Möglichkeit für MIB-basierte Tabellen, Indizes aus anderen Tabellen zu verwenden. Insgesamt werden Indizes aus den vier anderen Tabellen *hlHostControlTable*, *alHostTable*, *nlHostTable* und *protocolDirTable* der RMONv2 MIB verwendet.

alMatrix (*rmon.17*)

Eine weitere Gruppe mit Daten ähnlich zu den Gruppen *hostTopN* und *matrix* ist die Gruppe *alMatrix*. In diesem Fall beziehen sich die Angaben jedoch auf Protokolle der Anwendungsschicht des OSI Referenzmodells. Die zugehörigen Tabellen sind *alMatrixTopNControlTable* und *alMatrixTopNTable* für die „Top-Ten" Statistiken sowie die Tabellen *alMatrixSDTable* und *alMatrixDSTable* mit Statistiken über Pakete, die zwischen je zwei Hosts gesendet wurden.

usrHistory (*rmon.18*)

Die *usrHistory* Gruppe ist eine Erweiterung der *history* Gruppe um die Möglichkeit, nicht nur andere als Ethernet-Pakete messen zu können, sondern auch benutzerdefinierte Statistiken sammeln zu können. Die Tabelle *usrHistoryControlTable* dient der Konfiguration der jeweiligen Sensoren. Anders als in der *history* Gruppe werden die eigentlichen Daten aber in zwei weiteren Tabellen gespeichert. Dies ist zum einen die Tabelle *usrHistoryObjectTable*, in der lediglich die zu messenden MIB Instanzen angegeben sind. Zum anderen handelt es sich um die Tabelle *usrHistoryTable*, in welcher die gemessenen Daten zu den Objekten der ersten Tabelle gespeichert werden.

Ein zusätzlicher Vorteil der *usrHistory* Gruppe findet sich in der Art, wie die Daten ermittelt werden. Musste bei RMONv1 noch die Netzwerkmanagementstation periodisch die Daten aus der *history* Gruppe abrufen und weiterverarbeiten, so kann diese Aufgabe bei RMONv2 direkt vom RMON Monitor übernommen werden. Auf diese Weise wird die notwendige Netzwerklast noch einmal reduziert.

probeConfig (*rmon.19*)

In der *probeConfig* Gruppe werden keine gesammelten Statistiken gespeichert oder deren Datenerfassung konfiguriert. Diese Gruppe enthält viel mehr drei Tabellen und elf weitere skalare Objekte, die Angaben zum RMON Monitor selber machen oder Konfigurationsparameter für den RMON Monitor bieten. Zu diesen Angaben zählen beispielsweise die Hardware und Software-Versionen der Sonde, die einzelnen Fähigkeiten und unterstützten RMON Gruppen, Angaben zur Modem-Unterstützung oder auch Konfigurationsparameter für das Trivial File Transfer Protocol (TFTP). Ein Kontrollobjekt für den RMON Sensor ist beispielsweise ein schreibbarer Eintrag, über den ein Warmstart oder sogar ein Kaltstart des Monitors initiiert werden kann oder ein Eintrag zum Herunterladen und Aktivieren einer neuen Firmware für den RMON Monitor.

8.2 Proprietäre Client-Server-Lösungen

Prinzipiell können Netzwerkmanagementsysteme jede beliebige Kommunikationsform für die Erfüllung ihrer Aufgaben verwenden. Die bisher beschriebenen Mechanismen verwenden entweder den SNMP Transportmechanismus mit den beiden UDP Ports 161 und 162 oder sie verwenden den SYSLOG Mechanismus mit dem UDP Port 514. Denkbar ist allerdings auch eine Verwendung ganz anderer Transportmechanismen. Beispielsweise kann ein Managementsystem einen anderen Transportmechanismus einsetzten, um die Problematik der Unzuverlässigkeit der Nachrichtenauslieferung zu umgehen. Manchmal können aber auch einzelne Dienste über proprietäre Kommunikationswege überwacht und gesteuert werden.

8.2.1 Netzwerkmanagement mittels Web-Schnittstelle

Gerade über die Visualisierung von Netzwerkmanagement-Informationen und Aufgaben macht keiner der Standards eine Aussage. Schließlich beschreiben die Standards lediglich die technischen Merkmale der jeweiligen Mechanismen; sie machen aber keine Vorgaben für die Implementierung, Visualisierung und Bedienung der Mechanismen. Bei einer Trennung von Funktionalität und Bedienung kommt bei vielen Werkzeugen häufig eine Web-Schnittstelle zum Einsatz. Über diese lassen sich bequem selbst dynamische Informationen darstellen. Andererseits können über die Web-Schnittstelle auf eine einfache Art und Weise Eingaben des Benutzers entgegengenommen und weitergeleitet werden. Schließlich garantiert die Verwendung einer Web-Schnittstelle die Unabhängigkeit von Betriebssystemen und Plattformen. In der heutigen Zeit lassen sich auf nahezu jedem elektronischen Gerät, das über ein Display verfügt, Web-Inhalte anzeigen und Eingaben entgegennehmen. Herausragende Beispiele für die Vielfalt der Möglichkeiten sind neuere Mobiltelefone oder der Personal Digital Assistent (PDA). Zugleich ermöglicht die Verwendung eines Standards für die Bedienung der Eingabe und Ausgabe des Netzwerkmanagements eine ortsunabhängige Überwachung und Konfiguration des Netzwerks. Abbildung 8.1 veranschaulicht die Trennung zwischen dem Netzwerkmanagementprotokoll und der Komponente zur Bedienung und Visualisierung mittels Web-Schnittstelle. Die Netzwerkmanagementstation befindet sich im verwalteten Netzwerk, damit sie Zugriff auf alle Agenten erhält. Das verwendete Protokoll innerhalb des Netzwerkes ist nicht festgelegt und könnte beispielsweise SNMP sein. Der Administrator befindet sich außerhalb der verwalteten Netzwerks und greift mittels des Hypertext Transfer Protocol (HTTP) [68] auf die Netzwerkmanagementstation zu, um die einzelnen Geräte im Netzwerk zu verwalten.

Sicherheitsprobleme der Web-Schnittstelle

Wie bereits beschrieben bringt die Auslagerung der Visualisierung und Bedienung der Netzwerkmanagementstationen einige Vorteile mit sich. Hier sind vor allem die Flexibilität, Plattformunabhängigkeit und Ortsunabhängigkeit zu nennen. Demgegenüber stehen die sicherheitsbedenklichen Aspekte der Verwendung einer Web-Schnittstelle. Im Folgenden sollen die einzelnen Aspekte näher betrachtet und Lösungsmöglichkeiten für die daraus resultierenden Probleme diskutiert werden.

Unsicherer Transportweg

Das HTTP Protokoll gilt grundsätzlich als unsicher, da die Informationen unverschlüsselt zwischen dem Server und den Clients übertragen werden. Eine schnelle Lösung dieses Problems erhält man durch den Einsatz des sichere-

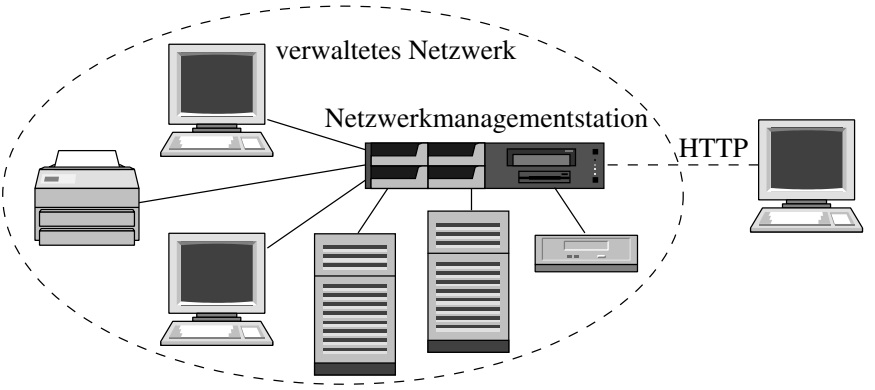

Abb. 8.1. Entfernte Verwaltung des Netzwerks über ein beliebiges Web-fähiges System, das sich auch außerhalb des verwalteten Netzwerkes befinden kann. Die Kommunikation zwischen dem entfernten System und der Netzwerkmanagementstation verläuft über das Web-Protokoll HTTP.

ren Protokolls Secure Hypertext Transfer Protocol[1] (HTTPS) [172] bei dem eine vollständige Verschlüsselung der übertragenen Daten gewährleistet ist. Die Möglichkeit der Server-Zertifizierung, die das HTTPS Protokoll ebenfalls mitbringt, hilft einen „Man-in-the-Middle" Angriff zu vermeiden.

Problematisch ist die Verwendung des HTTP(S) Protokolls, wenn der dazu benötigte Web-Server nicht vom Netzwerkmanagementsystem entkoppelt werden kann. In diesen Fällen kann es vorkommen, dass die proprietär in die Managementstation integrierte Web-Server Software HTTPS nicht unterstützt. Eine nachträgliche Erweiterung um das sichere Protokoll ist dann nicht oder nur schwer möglich. In diesem Fall sollte man überlegen, ob nicht eine bessere Alternative zur Web-Schnittstelle existiert, welche die Sicherheitsaspekte besser beleuchtet und ein sicheres Netzwerkmanagement erst ermöglicht.

Ortsunabhängigkeit

Die sich durchaus als vorteilhaft erweisende Ortsunabhängigkeit hat auch ihre Schattenseiten. Da der Zugriff teilweise sogar bewusst von unbekannten Orten aus möglich sein soll, können keine effektiven Filtermechanismen nach IP Adressen oder ähnlichen Eigenschaften durchgeführt werden. Dies verhindert zugleich den kontrollierten Einsatz einer Firewall. Eine Zugangskontrolle kann daher nur auf höheren Ebenen des OSI Referenzmodells stattfinden. Eine Möglichkeit dafür ist eine Authentifizierung direkt an der Netzwerkmanagement-Applikation oder auch am darunterliegenden Web-Server.

[1] HTTPS steht ursprünglich für den Ausdruck „HTTP secured by SSL", also eine Absicherung der Kommunikation mittels Netscapes Secure Socket Layer (SSL) [72] Mechanismus. Der von Netscape patentierte SSL Mechanismus ist in heutigen Standards durch das SSL-kompatible Transport Layer Security (TLS) [60] Protokoll ersetzt worden.

Plattformunabhängigkeit

Was dem Administrator einen bequemen Zugriff von beliebigen Geräten zur Netzwerkmanagementstation ermöglichen soll, erleichtert auf dieselbe Weise auch den Zugriff eines Unbefugten. Im selben Maße, wie die Anforderungen und Voraussetzungen an das über die Web-Schnittstelle zugreifende System weiter herabgesetzt werden, sinken auch die Hindernisse und Hürden für einen Angreifer von außen. Dieser benötigt lediglich ein Web-fähiges System, um die technischen Voraussetzungen zum Zugriff auf das Netzwerk und das Netzwerkmanagement zu erfüllen. Die Sicherheitsmechanismen müssen also an anderer Stelle definiert und implementiert werden.

Unsichere externe Systeme

Die Verwendung des HTTP(S) Protokolls für den Zugriff auf das Netzwerkmanagement ermöglicht es einem Administrator prinzipiell, sich von überall auf der Welt und vor allem auch von jedem beliebigen System eine Verbindung mit dem Netzwerk aufzubauen. Das bedeutet in letzter Konsequenz, dass über die Sicherheit der entfernten Systeme keinerlei Aussagen getroffen werden können. Verwendet man als abschreckendes Beispiel einen beliebigen Rechner eines Internetcafés, so kann mit hoher Wahrscheinlichkeit von einem Sicherheitsproblem ausgegangen werden. Im schlimmsten Fall ist der Rechner mit einem Tastatur-Logger ausgestattet, so dass die über das Internet eingegebenen Passwörter – inklusive der Zugangsberechtigung für das Netzwerkmanagement – abgehört werden. Es ist demnach nicht ratsam, eine Verbindung mit dem Netzwerk oder dem Netzwerkmanagement von einem unkontrollierten System aus aufzubauen. Der Vorteil der Plattformunabhängigkeit sollte daher nur in Bezug auf die verwendete Hardware und Software interpretiert werden, nicht aber in der völligen Unabhängigkeit des Systems. Ein Beispiel für die Einschränkung des Zugriffs auf bestimmte Plattformen wäre die Einrichtung einer Zertifikat-basierten Authentifizierung. Diese lässt sich durch das Vorschreiben eines VPN Tunnels gleichzeitig noch verschlüsseln. Auf diese Weise können nur Plattformen auf das Netzwerkmanagement zugreifen, die über das korrekte Zertifikat verfügen. Gleichzeitig sollte sichergestellt werden, dass nur vertrauenswürdige Systeme in den Besitz des Zertifikats gelangen. Es spricht daher prinzipiell nichts dagegen, dass ein Administrator sein Netzwerk vom Urlaubsstrand aus verwaltet, jedoch sollten sowohl das verwendete Gerät als auch die verwendeten Protokolle sehr gut ausgewählt oder konfiguriert werden.

8.2.2 Netzwerkmanagement mittels Textkonsole

Betrachtet man nicht den Weg zwischen der Managementstation und der Visualisierungskomponente, sondern den Weg von der Managementstation zu den verwalteten Geräten, so ist die Wahl der möglichen verwendbaren Protokolle abhängig von der Systemarchitektur teilweise stark eingeschränkt. In

der heutigen Zeit lassen sich die meisten Geräte mehr oder weniger über einige der verschiedenen Versionen des Protokolls SNMP administrieren und überwachen. Fast alle Systeme bieten aber einen Zugriff auf eine Textkonsole, über die oftmals sogar eine deutlich umfangreichere Konfiguration des Systems möglich ist. Grundsätzlich kann man bei den Textkonsolen drei verschiedene Varianten identifizieren, die teilweise mit unterschiedlichen Voraussetzungen verknüpft sind. Allen drei Varianten ist aber gemein, dass sie auf die eine oder andere Weise lesbare Zeichen zur Eingabe an das verwaltete Gerät senden oder lesbare Zeichen von diesem System als Ausgabe entgegennehmen.

Textkonsole über eine Serielle Schnittstelle

Verwendet man eine Textkonsole über eine Serielle Schnittstelle, so kann man auf höhere Protokolle wie das Internet Protocol (IP) oder das Transmission Control Protocol (TCP) verzichten. Gleichzeitig besteht jedoch die Einschränkung, dass die Konsole nicht mehr von einem beliebigen Ort aus aufgebaut werden kann, sondern es existiert nur noch eine einzige feste Verbindung, die auch noch über eine restriktive Längenbeschränkung verfügt. Diese Art von Verbindung findet sich häufig beim Out-of-Band Management, da es sich um vollständig unabhängige Verbindungswege zu den verwalteten Geräten handelt. In solchen Fällen haben nur noch physikalische Bedrohungen eine hinreichende Stärke. Durch Abhören der (meist unverschlüsselt) über die Leitung übertragenen Daten kann ein Angreifer in den Besitz von geheimen Informationen gelangen. Sind sowohl die überwachten Geräte als auch die Netzwerkmanagementstationen ausreichend physisch abgesichert, so stellt das Management über eine Textkonsole an der Seriellen Schnittstelle eine vergleichsweise sichere Methode dar. Die Nachteile liegen jedoch auch klar auf der Hand. Das Management über die serielle Konsole ist ortsgebunden, da im Normalfall nur von genau einer einzigen Maschine aus die Verwaltungsaufgaben und Überwachungsaufgaben ausgeführt werden können. Außerdem handelt es sich bei dieser Art von Netzwerkmanagement in vielen Fällen nicht um eine integrierte, sondern um eine isolierte Lösung, die nicht mit anderen Mechanismen gekoppelt ist. Erst mit so genannten Wandlern, welche das über die Seriellen Schnittstellen verwendete Protokoll in höhere Protokolle wie die TCP/IP Protokollfamilie übersetzen, können diese Nachteile überwunden werden. Zur gleichen Zeit wird aber das Sicherheitsniveau auf den gleichen Stand gebracht, den auch TELNET Verbindungen aufweisen.

Textkonsole mittels TELNET-Verbindung

Die zweite Kategorie der Verbindungen zu verwalteten Geräten über eine Textkonsole kann sehr leicht mit dem beliebten – aber unsicheren – Werkzeug TELNET durchgeführt werden. In Abbildung 9.13 auf Seite 271 wurde bereits ausführlich dargestellt, wie eine Kommunikation zwischen TELNET-Client und -Server auf der Netzwerkebene verläuft. Dabei wurde auch verdeutlicht, dass

die gesamte Kommunikation unverschlüsselt verläuft. Da dies auch auf die Authentifizierung mit der Übertragung des Benutzernamens und des Passwortes zutrifft, ist das Netzwerkmanagement mit Hilfe von TELNET als durchweg unsicher zu bezeichnen und sollte daher auch nicht eingesetzt werden.

Unglücklicherweise sind es aber häufig Verbindungen über TELNET, die von den verwalteten Geräten bevorzugt angeboten werden. Handelt es sich um proprietäre Systeme, so kann man in vielen Fällen keinen oder nur mit einem Aufpreis versehenen Einfluss auf die Unterstützung alternativer Protokolle nehmen. Bei einigen Netzwerkkomponenten werden verschlüsselte Verbindungen, wie sie das im nachfolgenden Abschnitt beschriebene SSH Werkzeug aufbaut, nur bei Vorhandensein eines entsprechenden optionalen Verschlüsselungsmoduls unterstützt. Findet sich jedoch keine andere und sicherere Möglichkeit der Kommunikation mit einem verwalteten Gerät, so ist auf Netzwerkebene durch die Errichtung von Firewalls und Paketfiltern der Zugriff auf das System so restriktiv wie möglich zu gestalten. Idealerweise kann dann nur noch von der Managementstation auf das verwaltete Gerät zugegriffen werden.

Textkonsole mittels SSH-Verbindung

Die letzte der drei möglichen Kommunikationsverbindungen mit einer Textkonsole basiert auf der verschlüsselten Kommunikation zwischen einem SSH-Client und dem zugehörigen SSH-Server. Der Unterschied zum TELNET Befehl liegt einzig in der Verschlüsselung der Kommunikation sowie einer optionalen parallelen Authentifizierung, die auch mit Zertifikaten realisierbar ist. Durch die Verschlüsselung fallen die Sicherheitsprobleme weg, die bei einem unsicheren Übertragungsweg zwangsweise entstehen. Ein Angreifer ist ohne Weiteres nicht in der Lage, an den Inhalt der übertragenen Daten zu gelangen. In vielen Fällen kann eine TELNET Verbindung durch eine sichere SSH Verbindung ersetzt werden. Sofern dies möglich ist, sollte unbedingt davon Gebrauch gemacht werden. Die meisten der heutigen proprietären Netzwerkkomponenten unterstützen SSH, so dass nur wenige Geräte verbleiben, bei denen eine sichere Kommunikation auf eine andere Weise herzustellen ist. In diesem Fall sollte man darüber nachdenken, ob diese Geräte nicht durch modernere ausgetauscht werden sollten, welche den bekannten Sicherheitsmechanismen mehr Aufmerksamkeit widmen.

8.2.3 Netzwerkmanagement mittels KVM Switch

Wie bereits in Abschnitt 2.5.3 kurz beschrieben, kann eine Alternative zu einer Konsolenverbindung durch die Administration über einen Keyboard-Video-Mouse (KVM) Switch bestehen. In diesem Fall ist die Verbindung zwischen den verwalteten Geräten und dem zentralen Switch durch eine Punkt-zu-Punkt Verbindung realisiert. Durch die Tatsache, dass lediglich die Tastatureingaben und die Mauseingaben sowie die Videoausgaben an den KVM

Switch weitergeleitet werden, ist diese Verbindung zunächst einmal vergleichsweise sicher. Gleichzeitig findet alle Administration über den zentralen KVM Switch statt, was sowohl Vorteile als auch Nachteil haben kann. Ganz klarer Vorteil ist die Möglichkeit der Administration der Geräte über deren gewohnte graphische Oberfläche. Dadurch kann ein Maximum der Funktionalitäten für das Netzwerkmanagement gewonnen werden. Außerdem kann die Managementstruktur durch das sternförmige Design vergleichsweise einfach abgesichert werden. Andererseits stellt der KVM Switch auch einen Single-Point-of-Failure dar; durch den Ausfall des KVM Switch kann also das gesamte Netzwerkmanagement außer Gefecht gesetzt werden.

9

Sicherheit in Netzen

In den vergangenen Jahren hat das Internet einen enormen Zuwachs an Akzeptanz und auch Nutzung gewonnen. In nahezu allen Bereichen des täglichen Lebens lassen sich mittlerweile sogar Abhängigkeiten vom Internet beobachten. Angefangen von der Wissenschaft, die durch ihren hohen Bedarf an weltweitem Informationsaustausch den Grundstein für das heutige Internet gelegt hat, über zahlreiche Prozessketten in Industrie und Wirtschaft, die für einen konkurrenzfähigen Betrieb auch vom Internet abhängen, bis hin zu den einzelnen Menschen, die über den heimischen Internetanschluss Einkäufe tätigen, Reisen buchen oder soziale Kontakte pflegen; überall gewinnt das Internet ständig weiter an Bedeutung. Aber eine steigende Anzahl von Nutzern zieht auch einen Anstieg des Missbrauchs des Internets mit sich. Fest steht: Je größer die Abhängigkeit von einer Technologie ist, desto größer ist auch der Schaden, den eine Störung dieser Technologie verursachen kann. Außerdem bedeuten die Risiken des einen zugleich auch immer eine Chance für einen anderen. Dementsprechend werden sich immer Individuen finden, welche das Internet dazu verwenden, anderen Schaden zuzufügen. Das können gleichermaßen Geheimdienste sein, die durch einen „Cyber-War“ die Feinde der eigenen Nation angreifen oder ausspionieren wollen oder auch Unternehmen, die sich geheime Informationen von der Konkurrenz verschaffen wollen.

Dieses Kapitel widmet sich zunächst den verschiedenen Bedrohungsszenarien. Neben dem totalen Verlust des Zugangs und damit der Nutzungsmöglichkeit des Internets existieren noch weitere Zwischenformen von Bedrohungen wie beispielsweise Informationsverfälschung oder eine unberechtigte Informationsweitergabe. Anschließend werden verschiedene Angriffsformen besprochen, durch welche die Bedrohungen in die Praxis umgesetzt und somit zu einer Gefahr gemacht werden können. Nach einer kurzen Betrachtung der Angriffsziele werden abschließend die Auswirkungen auf das Netzwerkmanagement analysiert. Ziel ist Minimierung der Risiken für das Netzwerk im Allgemeinen und das Netzwerkmanagement im Speziellen.

9.1 Bedrohungen

Grundsätzlich lässt sich in Datennetzen zwischen vier verschiedenen Bedrohungen differenzieren: dem Verlust, der Verfälschung, dem Vortäuschen und dem Bekanntwerden von Informationen. Um die Schwere dieser Bedrohungen richtig einschätzen zu können und um geeignete Gegenmaßnahmen identifizieren zu können, muss bei den verschiedenen Bedrohungsarten noch unterschieden werden, wann die entsprechenden Angriffe bemerkt werden oder ob sie unentdeckt bleiben.

9.1.1 Verlust von Informationen

Der Verlust von Informationen kann entweder bei der Datenhaltung oder bei der Datenübermittlung auftreten. Handelt es sich um eine Nachrichtenübertragung, bei welcher die gesendeten Informationen den Empfänger nicht erreichen, so kann die Ursache für den Informationsverlust im Wesentlichen nur an drei verschiedenen Stellen zu finden sein: beim Sender, beim Übertragungsweg oder beim Empfänger selbst. Eine besondere Form des Informationsverlustes ist ein Ausfall der für die Speicherung oder Übertragung der Daten notwendigen Infrastruktur durch einen so genannten Denial of Service (DoS) Angriff. Ein Verlust von gespeicherten Informationen andererseits kann aus einem Hardware-Fehler oder einem Software-Problem resultieren, wobei auch dies durch einen Angreifer verursacht werden kann. Abbildung 9.1 zeigt schematisch den Idealzustand ohne Informationsverlust.

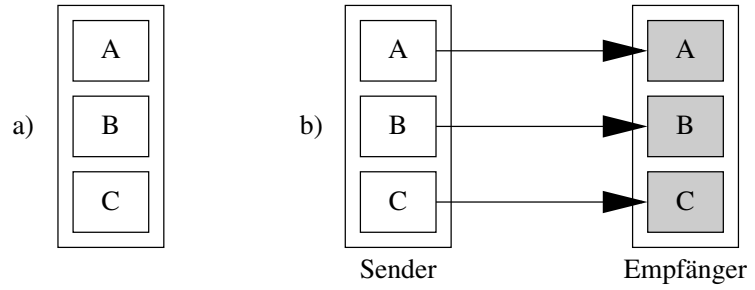

Abb. 9.1. Idealzustand ohne Informationsverlust a) bei der Datenhaltung und b) bei der Datenübermittlung.

Der Sender hat eine zu übermittelnde Nachricht nicht gesendet

Der erste Ansatzpunkt, das Übermitteln von Informationen zu unterbinden, ist der Sender dieser Nachricht (siehe Abbildung 9.2). Ein Angreifer könnte beispielsweise beim Sender die Sensoren manipulieren, die das Aussenden

einer Nachricht initiieren. Als klassisches Beispiel kann man sich eine Alarmanlage vorstellen, die auf Grund von Geräuschen, Bewegungen, Temperaturschwankungen oder vielleicht sogar dem CO_2-Gehalt der Luft einen Einbrecher erkennt und daraufhin einen Alarm auslöst. Hat der Einbrecher zuvor die Sensoren manipuliert – also hat er beispielsweise bei einer Lichtschranke durch Spiegel den Lichtweg umgeleitet, so dass er beim Öffnen von dahinterliegenden Türen und Fenstern nicht mehr unterbrochen wird – so erkennt der Sender (die Alarmanlage) das Ereignis, welches eine Nachrichtenübermittlung auslöst (den Einbruch) nicht mehr. Folgerichtig wird die Nachricht erst gar nicht übermittelt.

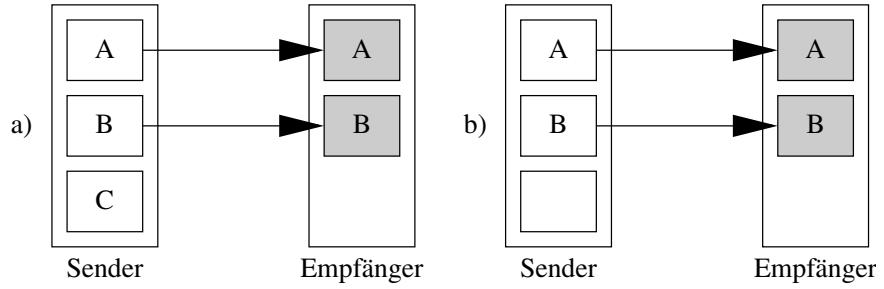

Abb. 9.2. Informationsverlust beim Sender einer Nachricht. a) Der Sender besitzt zwar die Informationen, sendet sie aber nicht. b) Der Sender verfügt nicht mehr über die notwendigen Informationen.

In der Informationstechnik spielt neben den Geräten auch die auf ihnen laufende Software eine bedeutende Rolle. Ist beispielsweise ein Rechner mit einem SNMP-Agenten ausgerüstet (siehe hierzu Kapitel 4), der bei Überschreiten des Grenzwertes für die Datenträgerauslastung eine Nachricht an eine Empfängerstation senden soll, so kann eine Manipulation des SNMP-Agenten das Aussenden dieser Nachricht verhindern. Ein weiteres Beispiel wäre ein manipulierter Mail User Agent (MUA) – also ein E-Mail Client – welcher dem Benutzer das erfolgreiche Versenden der Nachricht vortäuscht.

Für eine richtige Einschätzung der Schwere dieser Bedrohung zählt vor allem der Umstand, ob das Aussenden der Nachricht erwartet wird. Dies hat nämlich entscheidenden Einfluss darauf, ob der resultierende Informationsverlust entdeckt wird. Erwartet der Empfänger die Nachricht, so wird er ihr Ausbleiben zeitnah erkennen und entsprechende Maßnahmen einleiten können. Dies ist vor allem bei regelmäßig wiederholten Nachrichten der Fall. Schwieriger wird es aber bei zeitlich unbestimmten Ereignissen, die durch besondere Ausnahmesituationen hervorgerufen werden. Wenn der Empfänger zwar grundsätzlich die Möglichkeit für das Überlaufen eines Datenträgers einräumt, so wird er dennoch dieses Ereignis nicht unbedingt erwarten und das Ausbleiben einer solchen Nachricht daher auch nicht erkennen können. Bei

dem obigen Beispiel des manipulierten MUA kann auch das Ausbleiben der Antwort auf die Nachricht zum Aufdecken des Informationsverlustes führen. In diesem Fall ist für den Sender nicht eindeutig erkennbar, wo das Problem genau liegt. Schließlich kann ein unterbrochener Übertragungsweg oder ein fehlerhafter Empfänger die Ursache für das Ausbleiben der Antwort sein. Zumindest aber ist der Verlust von Informationen erkannt worden – entweder der Verlust der ursprünglichen Nachricht oder der Verlust der Antwort – und der Sender kann entsprechende Maßnahmen einleiten.

Der Übertragungsweg einer Nachricht ist unterbrochen

Ein Informationsverlust bei einer Nachrichtenübermittlung tritt in den meisten Fällen durch eine Störung oder Unterbrechung des Übertragungsweges auf. Das sind entweder die Kabel, Funkwellen oder Lichtsignale, die dem Informationsfluss zugrunde liegen, oder aber es sind die Geräte auf dem Weg der Übertragungsstrecke, welche den Informationsfluss steuern. In diesen Geräten durchlaufen die Signale verschiedene Schichten des ISO OSI Referenzmodells [51]. Elektromagnetische Signale, Lichtwellen, Kabel, Modulatoren und Demodulatoren, all das gehört zur untersten Schicht des OSI Referenzmodells, der Physikalischen Schicht oder auch Bitübertragungsschicht. Fällt eine dieser Hardware-Komponenten auf dem Weg der Nachricht bei ihrer Übermittlung aus, so geht die Information verloren (siehe Abbildung 9.3).

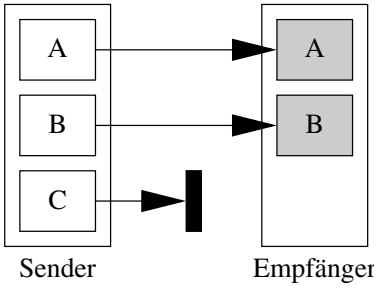

Abb. 9.3. Informationsverlust bei der Übertragung einer Nachricht.

Auch bei Problemen auf dem Übertragungsweg ist es von entscheidender Bedeutung, ob der Informationsverlust bemerkt wird. Unter bestimmten Bedingungen erhalten weder der Sender noch der Empfänger Kenntnis davon. Typischerweise wird dies durch Protokolle in den höheren OSI-Schichten verhindert: In der Transportschicht wird beispielsweise eine Verbindung zwischen Sender und Empfänger aufgebaut, die den Verlust von Informationen auf der Übertragungsstrecke verhindern soll. Zwar kann die Verbindung nach dem Durchtrennen eines Übertragungskabels nicht ohne Weiteres wiederhergestellt werden, jedoch wird das Problem zumindest vom Sender erkannt und es kön-

nen entsprechende Reaktionen eingeleitet werden, beispielsweise der Aufbau einer alternativen Verbindungsstrecke über Satellit.

Für eine Unterbrechung des Übertragungsweges gilt außerdem im Wesentlichen das Gleiche wie für eine Störung des Senders. Insbesondere Nachrichten, die nicht erwartet werden und deren Ausbleiben nichts Ungewöhnliches ist, laufen besondere Gefahr, unbemerkt verloren zu gehen. Dies gilt in besonderem Maße für Nachrichten, die ein verbindungsloses Protokoll zur Übermittlung verwenden.

Der Empfänger ignoriert die Nachricht

Eine auf den ersten Blick etwas seltsam anmutende, aber dennoch plausible Möglichkeit des Informationsverlustes besteht in einem unerwünschten Verhalten des Empfängers. Es kann sich dabei wieder um eine Manipulation der Software handeln, beispielsweise ein Programm, welches den korrekten Empfang der Nachricht vortäuscht und anschließend den Inhalt der Nachricht ignoriert (siehe Abbildung 9.4). Auf das weiter oben bereits geschilderte Beispiel einer Alarmanlage würde dies bedeuten, dass die Anlage zwar den Einbrecher erkennt und einen Alarm auslöst, der sogar das Sicherheitsunternehmen erreicht, dort aber wird der Alarm ignoriert und es werden keine weiteren Schritte eingeleitet. Ein Einbrecher kann also auch durch geschickte Manipulation des Empfängers bei der Sicherheitsfirma, welche den Alarm entgegennimmt, unbemerkt seiner „Arbeit“ nachgehen.

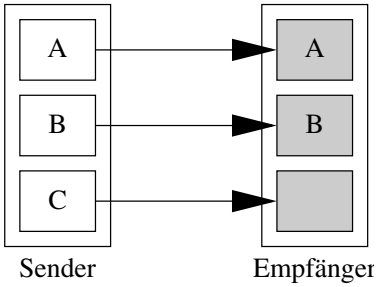

Abb. 9.4. Informationsverlust beim Sender einer Nachricht.

In diesem Fall ist es nicht der Empfänger, der vom Verlust der Information keinerlei Kenntnis erhält, sondern es ist der Sender, der fälschlicherweise von der korrekten Informationsübermittlung ausgeht. Diesen Umstand kann man durch das explizite Anfordern einer automatischen Antwort auf die Nachricht nicht verhindern, denn ein manipulierter Empfänger kann auch die Antwort korrekt senden und den Inhalt der Nachricht trotzdem ignorieren. Die Wahrscheinlichkeit eines unentdeckten Übertragungsproblems kann so jedoch verringert werden.

Gespeicherte Daten gehen verloren

Ein letztes Szenario, bei dem Informationen verloren gehen können, ist mit der Speicherung der Daten verbunden (siehe Abbildung 9.5). Kann ein Angreifer wichtige Daten löschen, so kann der daraus resultierende Schaden stark von der Zeitspanne abhängen, in welcher der Informationsverlust unentdeckt bleibt. Voraussetzung ist natürlich eine hinreichende Sicherung der Daten in Form von Backups. Im einfachen Fall eines Datei-Servers reichen hier Sicherungsbänder aus, von denen die verlorenen Daten wiederhergestellt werden können[1]. Je länger der Informationsverlust vor seiner Entdeckung zurückliegt, desto schwieriger wird eine vollständige Wiederherstellung der Daten. Man kann sich leicht vorstellen, dass Daten, die vor mehreren Jahren unbefugt gelöscht wurden, heute nicht zwangsweise vollständig zurückgeholt werden können. In diesem Fall bleibt allerdings die Frage zu klären, welchen Wert Daten besitzen, deren Löschung jahrelang nicht bemerkt wurde. Wird der Verlust der Informationen niemals bemerkt, dann haben diese Daten offensichtlich auch keinerlei Wert.

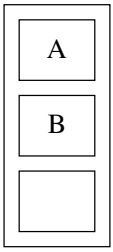

Abb. 9.5. Informationsverlust bei der Datenhaltung.

Schwieriger sieht es mit Daten aus, die nur dynamisch und flüchtig gehalten werden, und deren Sicherung kaum einen Sinn machen würde, weil der hierfür zu betreibende Aufwand sogar den Wert der Informationen übersteigen kann.

Zur Verdeutlichung sei an dieser Stelle das einfache Beispiel einer redundanten Firewall-Architektur erwähnt. Die zweite Firewall übernimmt dabei eine passive Aufgabe, bis sie eine Störung oder den Ausfall der primären Firewall beobachtet. In diesem Fall übernimmt das redundante Gerät die Aufgabe der ersten Firewall, damit für die Nutzer des Netzes die Störung möglichst nicht erkennbar wird. Zum erfolgreichen und korrekten Funktionieren der Firewall gehört allerdings auch das aktive Verfolgen der Verbindungen, die durch die Firewall aufgebaut werden. Dieser „Stateful Inspection“ Modus der Firewall soll verhindern, dass ein Angreifer beliebige Datenpakete an

[1] Das Thema Backup und Archivierung ist in der Realität etwas komplexer, als es hier zur Veranschaulichung präsentiert wird. Nur als ein Beispiel spielen bei der Archivierung auch Redundanzen eine große Rolle.

einen autorisierten Nutzer des Systems senden kann, auch wenn dieser durch eine zuvor von ihm gesendete Anfrage auf eine Antwort wartet. Die Firewall verfolgt nämlich die einzelnen zur Verbindung gehörenden Pakete und kann somit zwischen gültigen und ungültigen Paketen unterscheiden. Die Daten zu den bestehenden Verbindungen werden im Speicher der primären Firewall gehalten und sind nur von kurzer Gültigkeit. Dementsprechend werden diese Daten auch nach vergleichsweise kurzer Zeit wieder gelöscht. Fällt nun die aktive Firewall im Betrieb aus, so gehen auch die Verbindungsdaten verloren. Selbst wenn die sekundäre Firewall sofort den Betrieb übernimmt, so werden ohne passende Vorkehrungen alle aktiven Verbindungen getrennt und die Nutzer erfahren einen Verbindungsabbruch. In der Praxis kann sogar diesem Problem begegnet werden: Da die beiden Firewalls zur erfolgreichen Erkennung einer Störung einen direkten Kommunikationsweg besitzen, können sie auch Informationen über die aktuellen Zustände zwischen ihnen austauschen. So kann die redundante Komponente dieselbe Zustandstabelle aufbauen und deshalb auch den Betrieb zustandsabhängig übernehmen.

Ist ein redundantes Firewall-System nicht für den Austausch der Zustandsdaten ausgelegt, so kann ein Angreifer durch permanentes Initiieren einer Störung des Firewall-Systems einen nicht unerheblichen Anteil der Nachrichtensendungen über das verbindungsorientierte Transmission Control Protocol (TCP) trotz Redundanz erfolgreich verhindern.

9.1.2 Bekanntwerden von Informationen

Neben dem Verlust von Informationen kann manchmal auch das Gegenteil von Nachteil sein, nämlich das Bekanntwerden von Informationen. Auch hier kann es wiederum von entscheidender Bedeutung sein, ob und wann das unerwünschte Bekanntwerden der Daten bemerkt wird. Für eine korrekte Einschätzung des Ernstes der Bedrohung ist vor allem aber die Gruppe wichtig, die diese Informationen unberechtigt erhalten. Abbildung 9.6 veranschaulicht den Sollzustand, bei dem weder bei der Datenhaltung noch bei der Datenübermittlung Informationen an Unberechtigte gelangen.

Diebstahl von Informationen

Unter dem Diebstahl von Informationen soll an dieser Stelle das Bekanntwerden von Informationen für einen begrenzten Personenkreis verstanden werden (siehe Abbildung 9.7). Ein typisches Beispiel ist die Industriespionage, bei der sich die Konkurrenz gegenseitig um die Informationen der Mitbewerber bemüht. Fallen beispielsweise wichtige Entwicklungsdaten eines Unternehmens an einen direkten Mitbewerber, so ist dieser mit großer Wahrscheinlichkeit in der Lage, einen Vorteil daraus zu ziehen und somit seine Marktposition zu verbessern. Der entstehende Schaden kann mitunter beträchtlich sein. Gelangen die unrechtmäßig erworbenen Daten jedoch in den Besitz von Personen, die

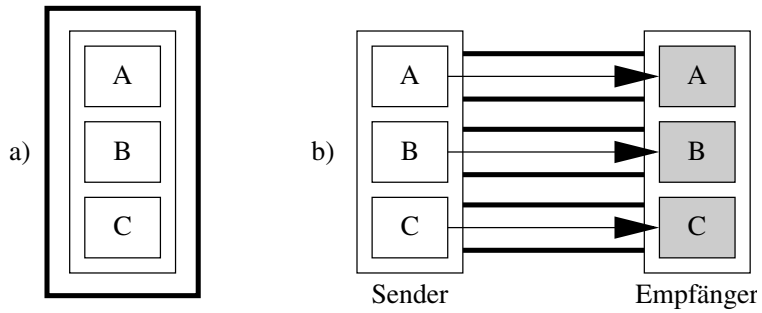

Abb. 9.6. Sollzustand: Es gelangen keinerlei Informationen an Unberechtigte a) bei der Datenhaltung und b) bei der Datenübermittlung.

daraus keinerlei Vorteile ziehen können oder wollen, so kann sich der entstandene Schaden stark reduzieren. Ein einfaches Beispiel wäre ein so genannter „White Hat", also ein Hacker, der in offiziellem Auftrag einen Einbruch in ein System durchführen soll. Die gewonnenen Informationen werden in diesem Fall aber nicht unberechtigt verwendet oder weitergegeben.

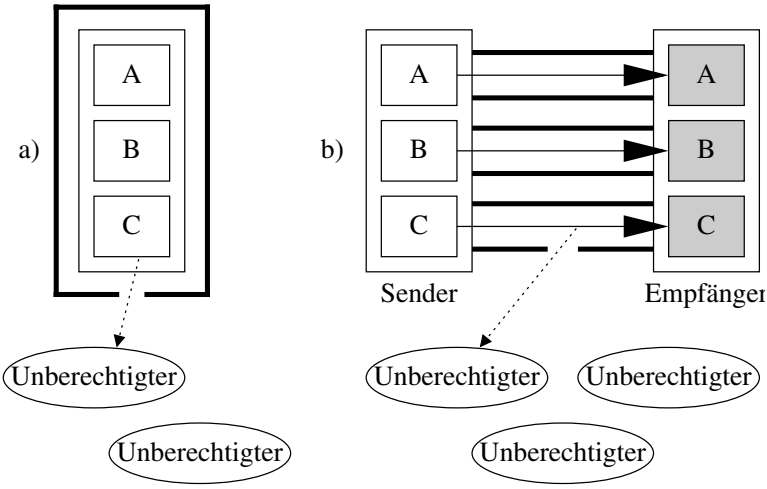

Abb. 9.7. Beim Diebstahl von Informationen gelangen Informationen entweder a) direkt oder b) bei der Datenübermittlung an einige, aber nicht alle Unberechtigte.

Gravierende Auswirkungen kann es haben, wenn der Diebstahl von Informationen nicht bemerkt wird. Insbesondere dann, wenn es sich bei den Informationen um Passwörter handelt. Die resultierenden Folgeschäden können immens sein. Ein Angreifer, der unrechtmäßig in Besitz eines Passwortes gelangt, repräsentiert alle der in diesem Kapitel vorgestellten Bedrohungsarten. Er ist

nämlich in der Lage, Informationen zu löschen, zu stehlen, zu veröffentlichen oder zu verfälschen.

Es gibt kein wirkungsvolles Mittel, Informationen vor Diebstahl effektiv zu schützen. Auch das Erkennen eines derartigen Zwischenfalls kann durch keine Maßnahme wirklich sichergestellt werden. Ein regelmäßiges Ändern von Passwörtern hilft, das Risiko eines Datenverlustes klein zu halten. Einen endgültigen Schutz garantiert es aber dennoch nicht. Schließlich kann ein erfahrener Angreifer, der nur kurz in den Besitz eines Passwortes gelangt ist, das entsprechende System mit Hintertüren versehen, die ihm den Zugang auch ohne Kenntnis des ursprünglichen Passwortes ermöglichen.

Veröffentlichung von Informationen

Die Veröffentlichung von Informationen unterscheidet sich grundsätzlich vom Diebstahl von Informationen. Zum Ersten sind veröffentlichte Informationen nach ihrem Bekanntwerden nicht nur einem eingeschränkten Kreis zugänglich, sondern der gesamten Öffentlichkeit. Zum Zweiten bleibt die Veröffentlichung verständlicherweise niemals unentdeckt. Abbildung 9.8 veranschaulicht die Veröffentlichung von Informationen.

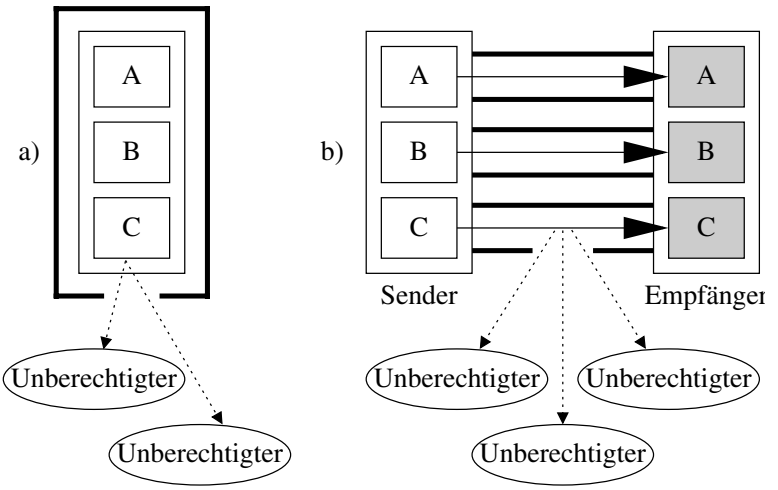

Abb. 9.8. Bei der Veröffentlichung von Informationen gelangen Informationen entweder a) direkt oder b) bei der Datenübermittlung an die gesamte Öffentlichkeit.

Diese spezielle Form der Bedrohung zielt oft auf den Ruf und das öffentliche Ansehen des Geheimnisträgers. Das Bekanntwerden bestimmter ungesetzlicher oder unmoralischer Tatsachen kann nicht nur einzelnen Unternehmen Schaden zufügen, sondern sogar ganzen Nationen. Gleichzeitig sind die Folgen der Veröffentlichung nur schwer wieder zu beseitigen. Gerade die öffent-

liche Meinung ist nur schwer zu beeinflussen und das Ansehen kann nachhaltig beeinträchtigt werden. Die Veröffentlichung von Informationen kann aber manchmal auch nur zu finanziellen Einbußen führen. Ein gutes Beispiel für ein solches Szenario stellt der Software Entwickler Valve[2] dar, der sich hauptsächlich mit der Erstellung von Action Computerspielen beschäftigt. Der Veröffentlichungstermin des aktuellen Bestseller Spiels Half-Life 2[3] unterlag einer beträchtlichen Verschiebung, weil bei einem Einbruch in das Netzwerk der Spieleentwickler große Teile des Quellcodes von Half-Life 2 in die Hände der Angreifer fiel. Anschließend wurde der Quellcode, an mehreren Stellen im Internet veröffentlicht [228]. Der Quellcode enthielt unter anderem auch den Netzzugangsteil, über den die Entwickler sich vor Raubkopien schützen wollten. Die Veröffentlichung dieses Geheimnisses führte dazu, dass Valve den relevanten Programmteil neu entwickeln musste, um die Integrität der Netzzugangskomponente weiterhin gewährleisten zu können.

9.1.3 Verfälschung von Informationen

Die wohl schlimmste Bedrohung geht von einer unbemerkten Verfälschung von Informationen aus. Ausgehend von Abbildung 9.1 als Sollzustand veranschaulicht Abbildung 9.9 die Verfälschung von Informationen bei der Datenhaltung oder der Datenübermittlung.

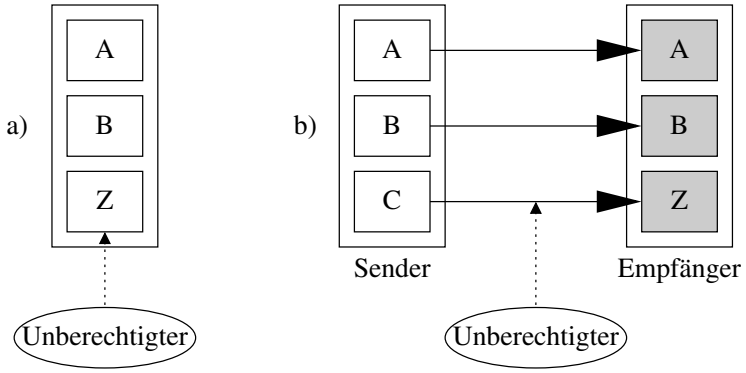

Abb. 9.9. Eine Verfälschung von Informationen kann sowohl a) bei der Datenhaltung als auch b) bei der Datenübermittlung auftreten.

Im extremsten Fall einer Informationsverfälschung kann der genaue Zeitpunkt der Verfälschung nicht mehr ausgemacht werden; gleichzeitig sind die falschen Informationen durch Abhängigkeiten in andere Informationsbereiche geflossen, die sich nur schwer oder vielleicht gar nicht mehr ausmachen lassen. Ein

[2] http://www.valvesoftware.com
[3] http://www.half-life.com

einfaches Beispiel sind die Datenfelder eines Kalkulationsprogramms, die mit Formeln untereinander verknüpft sind. Die Änderung eines einzigen Wertes kann sich unter ungünstigen Umständen auf alle anderen Werte auswirken, so dass die gesamte Kalkulation fehlerhaft wird. Wenn gleichzeitig die Ergebnisse dieser Kalkulation wieder als Eingang für weitere Berechnungen dienen, so pflanzt sich der Fehler auch in andere Datensätze fort. Wird die ursprüngliche Verfälschung des einzelnen Datensatzes nicht bemerkt, so sind auch die daraus resultierenden Folgefehler nur schwerlich auszumachen. Sollte der Fehler jedoch bemerkt werden, so kann der Aufwand zur Korrektur der Informationsverfälschung beträchtlichen Schaden verursachen.

Es ist nicht sehr einfach, diesem Problem effektiv zu begegnen. Handelt es sich eher um statische Daten, so kann über Prüfsummen und Signaturen eine Verfälschung sichtbar gemacht werden. Sind die Informationen jedoch sehr kurzlebig, so kann das ursprünglich manipulierte Datum bereits ungültig geworden und gelöscht worden sein. Die Folgefehler können möglicherweise aber weiterhin vorhanden sein. In einem solchen Fall ist die Korrektur der Verfälschung äußert schwierig und oft nur durch Überführen des Systems in einen bekannten und korrekten Zustand möglich – beispielsweise durch eine Neuinitialisierung oder das Einspielen eines Backup. Ein einfaches Beispiel für ein solches System ist ein simpler Zähler, der ständig – beispielsweise durch Eintreffen von entsprechenden Nachrichten – um einen Betrag erhöht wird, der auch vom Inhalt der Nachricht abhängen kann. Gelingt es einem Angreifer nun, den Wert des Zählers zu manipulieren, so sorgen alle weiteren eintreffenden Nachrichten für ein korrektes Hochzählen. Dennoch pflanzt sich der induzierte Fehler in alle zukünftigen Zustände des Zählers fort: Er zeigt einen falschen Wert an.

Eine weitere Art der Verfälschung von Informationen kann in der einfachen Umordnung oder Wiederholung der einzelnen Pakete eines Datenstroms bestehen. Damit Pakete beim Empfänger korrekt eingeordnet und Duplikate leicht erkannt und verworfen werden können, hilft in erster Lösung bereits die Einführung eines eindeutigen Index zur Durchnummerierung der Pakete.

9.1.4 Vortäuschung von Informationen

Das Vortäuschen von Informationen kann sich entweder auf gespeicherte Daten oder auf gesendete Nachrichten beziehen. Vor dem Hintergrund des in Abbildung 9.1 gezeigten Idealzustandes veranschaulicht Abbildung 9.10 den Fall der Informationsvortäuschung. In beiden gezeigten Fällen kann über eine Signatur die Herkunft der Daten zweifelsfrei identifiziert werden, so dass vorgetäuschte Informationen leicht erkannt und wieder gelöscht werden können. Vortäuschung von Informationen beinhaltet manchmal auch das Vortäuschen einer autorisierten Identität. Letztlich ist die Vortäuschung von Informationen nur ein Spezialfall der Verfälschung von Informationen. Schließlich können sich auch hier Folgefehler in anderen Bereichen des Systems ergeben. Nimmt

man beispielsweise Pakete eines Zugangsrouters, die der Abrechnung des Datenverkehrs dienen, so kann das Vortäuschen von solchen Paketen zu einem höheren beobachteten Datenvolumen führen, welches dann fälschlicherweise zusätzlich in Rechnung gestellt wird. Zumindest für den Kunden des Internet Service Provider (ISP) ergibt sich daraus ein direkter finanzieller Schaden.

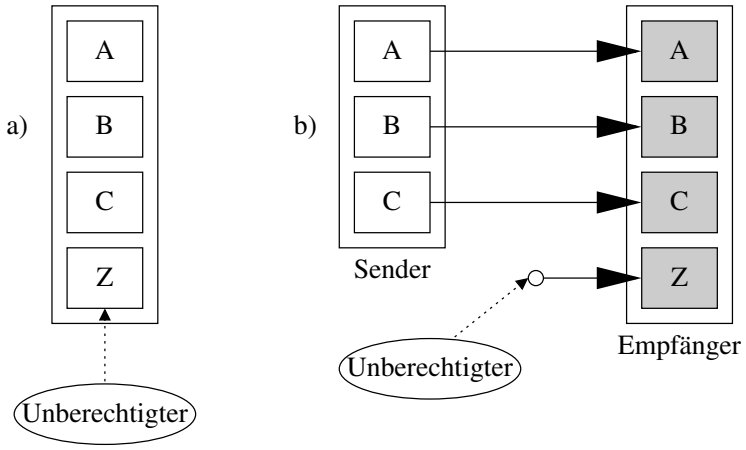

Abb. 9.10. Informationen können sowohl a) bei der Datenhaltung als auch b) bei der Datenübermittlung vorgetäuscht werden.

9.2 Angriffsformen

Die Angriffsformen in Datennetzen wurden in den letzten Jahren stark verfeinert und haben sich teilweise grundlegend verändert. Noch vor kurzer Zeit dominierten die physikalischen Angriffe, bei denen Techniken zur Informationsgewinnung auf physikalischem Wege eingesetzt werden – also beispielsweise durch das Messen von elektrischen Strömen auf Leitungen oder das Empfangen von Strahlungen über geeignete Antennen. Unabhängig davon existiert eine zweite Angriffsform – die Gruppe der logischen Angriffe – die sich auf einer höheren Ebene ohne den Einsatz zusätzlicher physikalischer Vorgänge abspielen. Beispiele für logische Angriffe sind das „Hacken“ eines Systems, also das unberechtigte Verschaffen des Zugangs zu einem System sowie das Entschlüsseln einer kodierten Nachricht oder Kommunikationsverbindung durch geeignetes Verschaffen des Schlüssels.

Durch die Weiterentwicklung in der Datenübertragungstechnik auf der einen Seite und durch die fortschreitende Vernetzung und Globalisierung auf der anderen Seite spielen die physikalischen Bedrohungen in der heutigen Zeit nur noch eine untergeordnete Rolle. Im selben Maße haben gleichzeitig die lo-

gischen Bedrohungen an Bedeutung gewonnen und wurden deshalb auch systematisch weiterentwickelt und verfeinert. Die „effektivste“ Angriffsart – also diejenige Angriffsform, welche den meisten Schaden bei geringstem Einsatz erzeugen kann – findet sich in den verschiedenen automatisierten Angriffs-Werkzeugen.

9.2.1 Physikalische Angriffe

Noch vor kurzer Zeit wurden vielfach Koaxial-Kupferkabel zur Datenübertragung verwendet. Dabei erreichten die Datenübertragungsgeschwindigkeiten gerade einmal 10 MBit/s. Zum Einsatz kamen entweder das so genannte „Thick-Cable“ (RG-8) mit einer maximalen Kabellänge von 500 m oder das „Thin-Cable“ (RG-58) mit einer maximalen Kabellänge von 185 m. Abbildung 9.11 veranschaulicht den inneren Aufbau dieser Rundkabel.

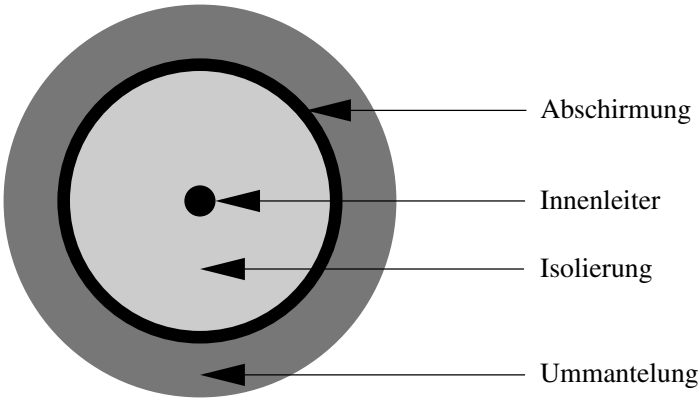

Abb. 9.11. Vierschichtiger Aufbau eines Koaxialkabels aus Innenleiter, Isolierung, Abschirmung und Ummantelung.

Netzwerke, die aus Koaxialkabeln aufgebaut sind, verwenden häufig eine Bus-Topologie. Bei einem physikalischen Angriff auf ein derart aufgebautes Netzwerk lässt sich die Tatsache ausnutzen, dass die Daten über ein einzelnes Koaxialkabel zwischen den Netzwerkkomponenten ausgetauscht werden. Wie in Abbildung 9.11 zu erkennen ist, kann man sich ein Koaxialkabel vereinfacht als einen stromführenden – und damit auch datenführenden – Leiter vorstellen, der von einem Kupferdrahtgeflecht umgeben ist. Zur Isolierung befindet sich eine Trennschicht zwischen der inneren Ader und der äußeren Abschirmung: das so genannte Dielektrikum. Größe und Beschaffenheit dieser Isolierschicht besitzen einen entscheidenden Einfluss auf die maximal mögliche Kabellänge. Aus diesem Grund können die elektrischen Signale mit dem dickeren Thick-Cable über eine größere Distanz transportiert werden als mit dem dünneren Thin-Cable.

Die spezielle Form der Abschirmung des Koaxialkabels verfolgt zwei Ziele: Zum einen sollen störende Strahlungen aus der Umgebung von den mit hoher Frequenz übertragenen elektrischen Signalen im Inneren ferngehalten werden. Dies sichert eine hinreichende Fehlerunanfälligkeit. Zum anderen sollen die übertragenen Daten nicht in die Umgebung abgestrahlt werden. Durch die Bündelung der elektrischen Signale im Kabel wird eine Übertragung der Daten über eine so große Distanz erst möglich gemacht. Gleichzeitig wird ein Übersprechen der Signale zwischen zwei nebeneinander verlegten Kabeln verhindert. Vor diesem Hintergrund lässt sich leicht nachvollziehen, dass sich durch Entfernen der Abschirmung eines Koaxialkabels die übertragenen Daten und Informationen physikalisch abhören lassen. Will man diese Art von „Lauschangriff" professionell praktizieren, so kann direkt in das zu überwachende Kabel eine Weiche eingebaut werden. Dazu wird ein T-Stück in das Kabel eingesetzt, über das sich beliebige Geräte in das überwachte Netzwerk integrieren lassen, die dann sämtliche Daten auf der Leitung abhören können. Gleichzeitig wird man bei dieser Methode durch den Umstand unterstützt, dass Koaxialkabel typischerweise bei einer Bus-Topologie eingesetzt werden, die ein Abhören sämtlicher Daten aller Netzwerkgeräte an jeder Stelle des Kabels erlaubt.

Darüber hinaus lassen sich Daten auch auf den anderen Transportstrecken abhören. Der „ECHELON-Bericht" des europäischen Parlamentes [188] beschreibt eindrucksvoll, welche Methoden zur Spionage und insbesondere zur Industriespionage existieren. Weltweit verteilte Satelliten-Empfangsanlagen sind dabei in der Lage, die übertragenen Daten sämtlicher Kommunikationssatelliten abzuhören. Die so gewonnenen Informationen werden nicht nur für geheimdienstliche Zwecke ausgewertet, sondern anfallende Informationen aus Forschung und Wirtschaft werden den Industrieunternehmen des eigenen Landes zur Verfügung gestellt.

Bei diesem globalen Lauschangriff stellen moderne interkontinentale Unterwasserkabel eine neue Hürde dar. Die in der Vergangenheit verwendeten Kupferkabel konnten störungsfrei und damit auch unentdeckt von U-Booten aus abgehört werden. Diese Technik haben die Vereinigten Staaten von Amerika bereits im Jahr 1971 eindrucksvoll mit ihrem U-Boot *USS Halibut* unter Beweis gestellt [202]. Durch das Anzapfen eines russischen Unterseekabels konnte die dort unverschlüsselt übertragene militärische Kommunikation über viele Jahre unentdeckt abgehört werden.

Mittlerweile sind die Datenübertragungsgeschwindigkeiten für einzelne Kabel auf 1 GBit/s und mehr angestiegen, so dass die herkömmlichen Koaxialkabel auf Grund ihrer vergleichsweise hohen Dämpfung nicht mehr verwendet werden können. Zwar finden sich noch immer Kupferleitungen, die zur Datenübertragung eingesetzt werden und deshalb anfällig für ein Abhören sind. Immer häufiger kommen jedoch Glasfaserkabel zum Einsatz, die sich nur mit erheblich höherem Aufwand abhören lassen. Mit vergleichsweise geringem Aufwand könnte ein Unterseekabel an den Endpunkten an Land abgehört werden, jedoch befinden sich hier oftmals Hochsicherheitsgebäude, die sogar

einem Atomschlag Stand halten könnten. Bleiben noch die Signal-Verstärker übrig, die bei modernen Glasfaserkabeln etwa alle 400 Kilometer in die Leitung zwischengeschaltet werden, um der Abschwächung der Signale entgegenzuwirken. In diesem Fall ergibt sich die Möglichkeit zum passiven Abhören dadurch, dass hier oftmals die optischen Signale zur Verstärkung in elektrische Signale umgewandelt werden. Diese lassen sich dann wiederum durch die gleichzeitig auf Grund der hohen Frequenzen entstehende Strahlung abhören.

Aber auch Glasfaserkabel selbst stellen mittlerweile kein unüberwindliches Hindernis mehr dar. Zwar ist aktives Abhören mit einem Durchtrennen des Kabels verbunden und wird aus diesem Grund dem Betreiber des Kabels sofort auffallen. Besser ist da ein passives Abhören, das mit entsprechend empfindlichen Messgeräten möglich ist. Zum besseren Verständnis hilft wieder ein Ausflug in die Physik. Die Lichtstrahlen in einem Glasfaserkabel unterliegen der so genannten „Inneren Reflexion", da sie sich im optisch dichteren Medium befinden. Da es sich im Regelfall um parallel verlaufende Laserstrahlen handelt, treffen nur dann Lichtstrahlen auf die Wand des Glasfaserkabels, wenn dieses gebogen wird. Solange dabei nicht der Grenzwinkel Θ_c[4] überschritten wird, findet an der Kabelwand eine Totalreflexion der Lichtstrahlen statt, welche deshalb im Kabel verbleiben. Wird das Glasfaserkabel jedoch noch weiter gebogen, so beginnt ein Teil des Lichtes das Kabel zu verlassen. Abbildung 9.12 veranschaulicht diesen Sachverhalt. Während in Abbildung 9.12a der Lichtstrahl in einem steilen Winkel größer dem Grenzwinkel Θ_c einfällt und deshalb ein Teil des Lichtes das optisch dichtere Medium verlassen kann, ist der Einfallswinkel in Abbildung 9.12b kleiner als Θ_c. In diesem Fall findet eine Totalreflexion statt und Licht kann das optisch dichtere Medium nicht verlassen. Für das Abhören muss ein Glasfaserkabel also lediglich über den Grenzwinkel Θ_c hinaus gebogen werden. Wenn nun an dieser Stelle die reflektierende Ummantelung entfernt wird, so können die dort abgestrahlten Lichtsignale durch hochempfindliche optische Sensoren gemessen und in Daten umgewandelt werden.

Bei all diesen Abhörverfahren bleibt allerdings ein Problem bestehen: Um ein heutiges Breitband-Glasfaserkabel vollständig abhören zu können, benötigt man ein mindestens genauso breitbandiges Kabel, welches die abgehörten Daten zur Echtzeitanalyse und Auswertung an einen Ort übermittelt. Diese Tatsache macht das Abhören von Unterwasserkabeln zu einem äußerst aufwändigen und kostspieligen Verfahren, das nur in absoluten Ausnahmesituationen angewendet wird.

Die beschriebenen physikalischen Angriffe führen in der heutigen Zeit immer seltener zu einem unmittelbaren Gewinn an geheimen Informationen und sind höchstens noch ein Mittel zum Zweck. Spätestens mit der großflächigen

[4] Der Grenzwinkel Θ_c beschreibt den Einfallswinkel eines Lichtstrahls, bei dem eine Totalreflexion einsetzt. Alle Lichtstrahlen, die in einem flacheren Winkel einfallen, werden vollständig an der Oberfläche reflektiert. Lichtstrahlen, die in einem größeren Winkel auftreffen, können das optische Medium teilweise verlassen.

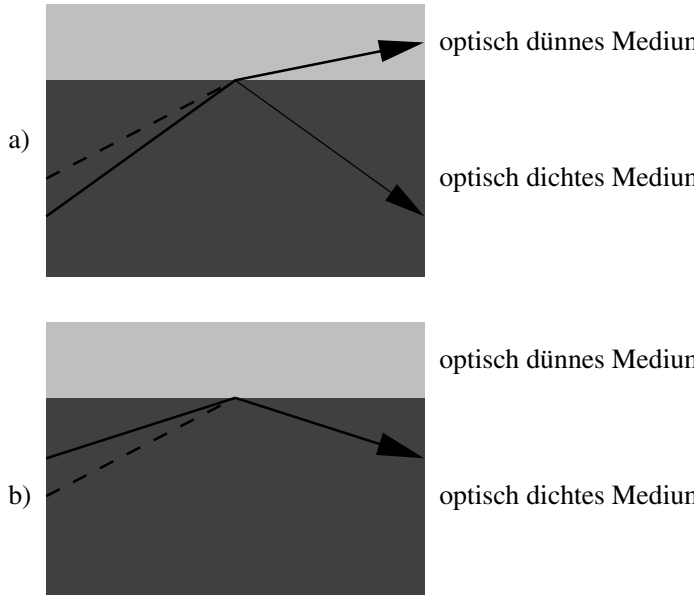

Abb. 9.12. Brechung und Reflexion eines Lichtstrahls an der Grenzfläche eines optisch dichteren Mediums zu einem optisch dünneren Medium. Der Grenzwinkel Θ_c ist durch eine gestrichelte Linie symbolisiert. a) Ein Teil des Lichtstrahls verlässt das optisch dichtere Medium, da der Einfallswinkel größer als der Grenzwinkel Θ_c ist. b) Es kommt zur Totalreflexion, da der Einfallswinkel kleiner als der Grenzwinkel Θ_c ist.

Einführung von Datenübertragung über Satelliten und andere Funkstrecken wurde das Bewusstsein bezüglich der physikalischen Bedrohungen so stark geschärft, dass andere Schutzmechanismen für unumgänglich erachtet und folgerichtig auch eingerichtet wurden. Gerade im hochgradig dynamischen System Internet kann der Weg eines Datenpaketes nie mit absoluter Sicherheit vorhergesagt werden. Die Vermittlung der Pakete wird nämlich nicht von der Länge der Übertragungsstrecken abhängig gemacht, sondern vielmehr von ihrer Bandbreite und Auslastung. Aus diesem Grund werden heutzutage verstärkt Verschlüsselungsverfahren eingesetzt. Das Problem besteht dann nicht mehr darin, die Daten auf physikalischem Wege zu erhalten, sondern darin, den verschlüsselten Datenstrom zu dekodieren und richtig zu interpretieren. Mitunter kann dies das eigentliche Problem werden. Kann ein Abhörzentrum die abgehörten Daten nicht in Echtzeit verarbeiten, so bricht es binnen kürzester Zeit unter der gewaltigen sich anstauenden Datenmenge zusammen.

9.2.2 Logische Angriffe

Auch die logischen Angriffe haben sich im Laufe der Zeit gewandelt. Sowohl die Techniken, mit denen man sich vor Angriffen zu schützen versucht als

auch die Methoden, mit denen man wiederum diese Hindernisse zu umgehen versucht, entwickeln sich unaufhaltsam weiter. Hat man früher noch auf Passwörter und einfache Verschlüsselungsalgorithmen gebaut, so erlauben es heutige Hochleistungsrechner, diese „einfachen" Hürden mit vergleichsweise geringem Aufwand zu umgehen. Durch eine Automatisierung wird zusätzlich für einen Angreifer nicht nur der Aufwand erheblich reduziert, sondern auch das vorausgesetzte Wissen und die erforderlichen Fähigkeiten sinken auf ein Minimum.

Angriffe gegen Passwörter

Die Passwortsicherheit ist ein besonderes Thema, vor allem weil Passwörter von Menschen verwendet werden. Betrachtet man den Sachverhalt zunächst rein technisch, so existieren bereits objektive Qualitätskriterien für Passwörter. Damit gemeint sind vor allem die Anzahl und die Art der im Passwort verwendeten Zeichen. Somit ließe sich ein Mindeststandard für die Passwortsicherheit leicht durch automatisches Generieren von Passwörtern einhalten. Dabei könnte beispielsweise darauf geachtet werden, dass die Passwörter nicht nur einer Mindestlänge genügen, sondern auch aus einer geeigneten Kombination von Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen bestehen.

Notizzettel. In der Praxis kommt zu den technischen Aspekten noch der Faktor Mensch hinzu. Ein automatisch generiertes, sicheres Passwort läuft allzuoft Gefahr, dass sich der Benutzer dieses Passwort nicht merken kann. Oft wird in so einem Fall das Passwort notiert und an einer aus Sicht des Anwenders geeigneten Stelle hinterlegt: Also beispielsweise auf einem Notizzettel am Monitor oder unter der Tastatur. Hat nun ein Angreifer zumindest für einen kurzen Augenblick physikalischen Zugriff auf das System und den Arbeitsplatz, so kann er leicht in den Besitz des Passwortes gelangen. Auf diese Art und Weise erfüllt das Passwort nicht mehr seinen ursprünglichen Zweck und es ist alles andere als sicher – unabhängig von seiner Qualität.

Zu einfache Passwörter. Dieses Problem kann nur dadurch beseitigt werden, dass die Passwörter nicht automatisch generiert werden, sondern vom Anwender selber. Überlässt man es aber einem Benutzer, sich ein für ihn gut zu merkendes Passwort auszudenken, so wird dieses häufig genug nicht dem minimalen Sicherheitsstandard genügen. Gerade unbedarfte Benutzer wählen für ihre Passwörter noch immer einzelne Wörter in Kleinbuchstaben, was keinen ausreichenden Schutz vor unberechtigtem Zugriff bietet. Von der steigenden Performanz heutiger Computer profitiert auch das bekannte Programm JOHN THE RIPPER. Dieser vollautomatische Passwort-Cracker arbeitet in zwei verschiedenen Modi. Im ersten Modus wird versucht, die verschlüsselt vorliegenden Passwörter mit einem Wörterbuch-Angriff (Dictionary Attack) zu erraten. Mit einer hinreichend großen Bibliothek ist die Wahrscheinlichkeit zum

Erraten derart einfacher, aus nur einem einzelnen Wort bestehenden Passwörter vergleichsweise groß. Falls nach Abarbeiten des Wörterbuchs das Passwort noch nicht gefunden wurde, so werden die Begriffe anschließend mit einzelnen Ziffern oder Sonderzeichen kombiniert oder ausgetauscht. Auf diesem Wege werden auch Passwörter gefunden, die auf eine unzureichende Art und Weise sicherer gestaltetet worden sind. Im zweiten Modus werden schließlich alle möglichen Zeichenkombinationen unterschiedlichster Länge ausprobiert. Vermeintlich sichere aber zu kurze Passwörter können so ebenfalls aufgespürt werden. Der Rechenaufwand steigt exponentiell mit der Länge der Zeichenketten an, so dass dieser Methode – abhängig vom verwendeten Rechensystem – klare Grenzen gesetzt sind.

Vermutlich kann nur durch ein umfangreiches Informieren der Benutzer diese Problematik erfolgversprechend angegangen werden. Schließlich existieren mehrere Möglichkeiten zur Erzeugung von sicheren und gleichzeitig leicht zu merkenden Passwörtern. Besonders vorteilhaft auf die Qualität und gleichzeitig auch auf die Merkbarkeit eines Passwortes wirkt sich das zusätzliche Ergänzen oder Austauschen von Buchstaben durch Ziffern oder Sonderzeichen aus. Doch Vorsicht ist auch hier geboten: Viele Ersetzungsmöglichkeiten sind nicht nur einfach zu merken, sondern auch den Angreifern bekannt. Tabelle 9.1 zeigt einige der gängigen Austauschmöglichkeiten für einzelne Buchstaben. Diese Ersetzungen beruhen im Wesentlichen auf der Ähnlichkeit der verschiedenen Zeichen, bekannt vor allem aus der so genannten „hacker eleet"-Sprache – genauer gesagt der „h4x0r 31337"-Sprache.

Tabelle 9.1. Gängige Möglichkeiten, einzelne Buchstaben in Passwörtern durch Ziffern oder Sonderzeichen zu ersetzen.

Buchstabe	Ziffer	Sonderzeichen
a,A	4	@
b,B	6,8	|3
c,C		(,{,[
D		),|)
e,E	3	
g,G	9,6	
H		|-|
i,I	1	!,|
k.K		|<
l	1	!,|
o,O	0	()
R		|2
s,S	5	$,§
t,T	7	+
w,W		vv,VV
x,X		><
z,Z	2	

Ein Beispiel für ein sicheres Passwort soll hier kurz angeführt werden. Ausgangsbasis für das Passwort ist ein Satz oder Ausspruch, den sich der Benutzer gut merken kann. Von Vorteil ist es, wenn der Satz für sich allein genommen nur wenig Sinn macht, sondern erst in einem anderen Kontext einen Zusammenhang ergibt. Das berühmte Zitat von William Shakespeare ‚Sein oder nicht sein, das ist hier die Frage' ist vermutlich eine schlechte Wahl. Besser wäre da beispielsweise die Antwort Albert Einsteins auf das Erscheinen eines Buches mit dem Titel ‚100 Autoren gegen Einstein':

Warum einhundert? Wenn ich Unrecht hätte, wäre einer genug!

Nun wählt man – unter Berücksichtigung der Klein- und Großschreibung – die Anfangsbuchstaben der Wörter dieses Satzes. Gegebenenfalls vorhandene Satzzeichen eignen sich ideal zur Ergänzung des Passwortes mit Sonderzeichen. Im oben gewählten Satz taucht des Weiteren eine Zahl auf, welche in Form von Ziffern direkt in das Passwort übernommen werden kann. Man gelangt also schließlich zu folgendem Passwort[5]:

W100?WiUh,weg!

Das in diesem Beispiel erstellte Passwort besteht aus 14 Zeichen und setzt sich neben den Kleinbuchstaben auch noch aus jeweils drei Großbuchstaben, Ziffern und Sonderzeichen zusammen. Nach diesem Schema lassen sich vergleichsweise einfach Passwörter generieren, die sich leicht merken lassen, und die gleichzeitig einem minimalen Sicherheitsstandard genügen. Dem geneigten Leser sei hier des Weiteren eine Veröffentlichung mehrerer Sicherheitsexperten aus dem Jahr 1996 empfohlen, welche sich mit der minimalen Länge und der Beschaffenheit von Passwörtern beschäftigt [17].

Tastatur-Logger. Egal wie sicher oder unsicher ein Passwort letztlich auch sein mag, manchmal existieren ganz einfache Methoden zur Aneignung der Passwörter. Eine Möglichkeit besteht beispielsweise in der Verwendung eines so genannten Tastatur-Loggers (Key-Logger). Hierbei handelt es sich um ein kleines Programm, das von einem Angreifer unbemerkt auf einem fremden Rechner installiert wird. Das Programm läuft anschließend unsichtbar im Hintergrund und protokolliert sämtliche Tastatureingaben. Das Protokoll kann anschließend vom Angreifer ausgelesen werden oder aber es wird sofort über das Netzwerk an ihn übermittelt. Wird nun von einem beliebigen Benutzer dieses Systems ein Passwort über die Tastatur eingegeben, so wird dieses ebenfalls protokolliert und erreicht damit auch den Angreifer.

Ein umstrittenes Beispiel aus dieser Kategorie stellt das Programm MAGIC LANTERN dar [187]. Das amerikanische Federal Bureau of Investigation (FBI) setzt neben dem Paket-Sniffer CARNIVORE auch diesen Tastatur-Logger ein. CARNIVORE alleine kann zwar die gesamte Kommunikation einer Internet-Leitung abhören, aber verschlüsselte Daten bleiben nach wie vor verborgen.

[5] Es versteht sich von selbst, dass das hier angeführte Passwort alles andere als sicher ist, da es schließlich in diesem Buch veröffentlicht worden ist!

Dieses Problem löst das Programm MAGIC LANTERN, mit dem die Tastaturanschläge eines zugehörigen Rechners abgehört werden können. So lassen sich vom Benutzer eingegebene Passwörter mitprotokollieren, mit denen anschließend die verschlüsselten Nachrichten leicht dekodiert werden können.

Unverschlüsselte Passwörter im Netzwerk. Kritisch ist die Situation ebenfalls, wenn Passwörter unverschlüsselt über das Netzwerk gesendet werden. Das bekannte und beliebte, auf dem *Telnet* Protokoll [163] basierende Programm TELNET fällt beispielsweise in diese Kategorie. Mit einem TELNET-Client kann sich ein Benutzer über das Netzwerk an einem entfernten Rechner anmelden, sofern dort der TELNET-Dienst läuft. Die gesamte Kommunikation läuft dabei unverschlüsselt ab. Das bei der Anmeldung über das Netzwerk gesendete Passwort ist für einen Angreifer, der die Netzwerkpakete abfangen kann, im Klartext lesbar.

Zur Erklärung hilft die Tatsache, dass beim TELNET-Protokoll die Benutzereingaben zeichenweise an den Server übergeben werden. Außerdem werden sämtliche Eingaben, die vom Anwender getätigt werden, vom Server wiederholt und an den Client zurück gesendet. Einzig davon ausgenommen ist das Passwort: Dieses wird zwar zeichenweise im Klartext an den Server gesendet, dieser wiederholt die empfangenen Zeichen jedoch nicht mehr. Der dadurch erreichte Schutz vor dem Abhören des Passwortes ist kaum erkennbar.

Abbildung 9.13 zeigt ein Beispiel, bei dem sich ein Benutzer `peter` mittels eines TELNET-Client bei einem Server anmeldet. Das verwendete Passwort lautet `unsicher`. Nach erfolgreicher Anmeldung führt der Benutzer den einzelnen Befehl PWD[6] auf dem entfernten Rechner aus.

Nachdem der Server seine Begrüßungsmeldung im Klartext an den Client gesendet hat, folgt eine Zeile mit einer Eingabeaufforderung, die im Beispiel den Inhalt `Login:` trägt. Jedes einzelne Zeichen des Benutzernamens, das der Anwender nun eingibt, wird jeweils in einem einzelnen Paket an den Server übermittelt. Im gezeigten Beispiel sind dies die Buchstaben `p`, `e`, `t`, `e` und `r` sowie das abschließende `[Enter]`. Jede einzelne Eingabe bestätigt der Server durch eine exakte Kopie, also dasselbe Zeichen im Klartext. Anschließend folgt vom Server die Eingabeaufforderung für das Passwort mit der Zeile `Password:`. Wieder wird jedes einzelne vom Benutzer eingegebene Zeichen in einem separaten Paket im Klartext an den Server übermittelt. Zwar wiederholt der Server die Zeichen nicht, jedoch ändert das nichts an der Unsicherheit dieses Verfahrens. Nach erfolgreichem Anmeldevorgang sendet der Server seine Begrüßungsmeldung. Der im Anschluss daran ausgeführte Befehl PWD wird – wie schon der Benutzername – Zeichen für Zeichen im Klartext übermittelt und vom Server quittiert. Das Ergebnis des ausgeführten Befehls

[6] Der Befehl PWD steht für die Abkürzung „Present Working Directory“ und liefert als Ergebnis das Verzeichnis, in dem sich der Benutzer gerade befindet. Direkt nach dem Öffnen einer neuen Verbindung zu einem System wird ein Benutzer vom System in sein Heimatverzeichnis gesetzt

1	Client	→	Server	**telnet 172.17.12.1**
2	Client	←	Server	Welcome!
3	Client	←	Server	Login:
4	Client	→	Server	**p**
5	Client	←	Server	p
6	Client	→	Server	**e**
7	Client	←	Server	e
8	Client	→	Server	**t**
9	Client	←	Server	t
10	Client	→	Server	**e**
11	Client	←	Server	e
12	Client	→	Server	**r**
13	Client	←	Server	r
14	Client	→	Server	**[Return]**
15	Client	←	Server	[Return]
16	Client	←	Server	Password:
17	Client	→	Server	**u**
18	Client	→	Server	**n**
19	Client	→	Server	**s**
20	Client	→	Server	**i**
21	Client	→	Server	**c**
22	Client	→	Server	**h**
23	Client	→	Server	**e**
24	Client	→	Server	**r**
25	Client	→	Server	**[Return]**
26	Client	←	Server	[Return]
27	Client	←	Server	bash$
28	Client	→	Server	**p**
29	Client	←	Server	p
30	Client	→	Server	**w**
31	Client	←	Server	w
32	Client	→	Server	**d**
33	Client	←	Server	d
34	Client	→	Server	**[Return]**
35	Client	←	Server	[Return]
36	Client	←	Server	/home/peter
37	Client	←	Server	bash$

Abb. 9.13. Client-Server Kommunikation bei einer TELNET-Anmeldung und anschließender Ausführung eines einfachen Befehls.

wird abschließend in diesem Beispiel als Ganzes in einem Paket vom Server übermittelt.

Die Problematik der unverschlüsselt über das Netzwerk übertragenen Passwörter lässt sich durch das Austauschen des TELNET-Dienstes durch einen anderen Dienst lösen, der eine verschlüsselte Kommunikation verwendet. Ein gutes Beispiel ist die „Secure Shell" (SSH) [231], bei der einschließlich des Anmeldevorgangs der gesamte Datenaustausch verschlüsselt wird.

Social Engineering. Andere Arten von Angriffen auf Passwörter zielen auf die „Schwachstelle Mensch". Eine weit verbreitete Methode ist unter dem Namen „Social Engineering" bekannt. Der berühmte amerikanische Hacker Kevin Mitnick, der für seine verschiedenen Einbrüche in Telefon- und Datennetzwerke eine mehrjährige Haftstrafe abgesessen hat, beschreibt in seinem Buch ‚Die Kunst der Täuschung' [131] eindrucksvoll die Methoden, mit denen er erstaunlich häufig und simpel an Passwörter oder andere geheime Informationen gelangte. Ein Angreifer, der sich der Social Engineering Methode bedient, nutzt verschiedene Verhaltensmuster von Menschen zu seinem Vorteil aus. Oft beginnt ein Angriff zunächst passiv mit der Beschaffung von Informationen, die oftmals sogar öffentlich zugängig sind. Viele Unternehmen veröffentlichen beispielsweise eine Liste ihrer Mitarbeiter auf ihrer Internetpräsenz. Eine von vielen Möglichkeiten könnte dann darin bestehen, dass man sich als einer der Mitarbeiter des Unternehmens ausgibt und sich telefonisch bei der Systemadministration meldet. Unter Vortäuschung eines dringenden Notfalls erklärt man nun, dass man sein Passwort vergessen habe. An diesem Punkt setzt der soziale Aspekt des Angriffs ein: Damit der scheinbar verzweifelte Anrufer seinen Job nicht verliert, ist man eher dazu geneigt, seine Hilfe anzubieten. Im konkreten Fall könnte dies bedeuten, dass der Administrator das Benutzerpasswort des Anrufers zurücksetzt und damit einem Fremden ungewollt Zugang zum System verschafft.

Reverse Social Engineering. Eine noch weiter entwickelte Methode des Social Engineering ist das so genannte „Reverse Social Engineering". Durch Schulung und Sensibilisierung kann man Mitarbeiter dazu anhalten, die Identität einer unbekannten Person zunächst einmal zu hinterfragen und zu überprüfen. Der oben beschriebene Angriff würde in diesem Fall vermutlich weniger Aussicht auf Erfolg haben. Dieses Verhaltensmuster greift aber häufig nur dann, wenn die Kontaktaufnahme vom Angreifer ausgeht. Durch eine intensivere Vorbereitungsphase, in der weiterführende Informationen gesammelt werden, kann man jedoch diese Situation als Angreifer geschickt zu seinen Gunsten ändern. Nach der Informationsgewinnung beginnt der Angriff dann damit, dass man sich den Benutzern als Verantwortlicher für bestimmte Aufgaben oder Probleme bekannt macht. Zu einem späteren Zeitpunkt provoziert man dann genau dieses Problem. Dadurch hat man erreicht, dass sich die Benutzer aus freien Stücken beim Angreifer melden. Zur Lösung des Problems sind sie dann auch eher bereit, dem Angreifer Informationen, Geheimnisse oder gar Passwörter preiszugeben.

Standard-Passwörter. Eine ganz andere Art von Angriffen auf Passwörter scheint zunächst sehr zu überraschen: Es gibt Situationen, in denen kennt ein Angreifer bereits das gesuchte Passwort, ohne dass er besondere Insider-Kenntnisse besitzen muss. Gemeint sind die Standard-Passwörter, die sich in vielen Geräten und Systemen wiederfinden. Das können zum einen Passwörter sein, die einem neuen oder einem zurückgesetzten Benutzerkonto zugewiesen werden. Zum anderen kann es sich aber auch um Passwörter handeln, die vom System voreingestellt sind. Alle durch ein Passwort geschützten Systeme, bei denen im Auslieferungszustand die Software bereits vorinstalliert ist, gewähren dem Anwender oftmals den Zugang über ein vorgegebenes Passwort. Dieses Passwort ist im Normalfall dasselbe für alle vom Hersteller ausgelieferten Systeme gleichen Typs. Zwar wird dem Anwender häufig nahegelegt, das vorinstallierte Passwort umgehend zu ändern, allzuoft geschieht dies aber gerade eben doch nicht. Das Standard-Passwort bietet keinerlei Schutz vor unberechtigtem Zugriff, da es einem Angreifer bereits bekannt ist.

Versteckte Passwörter. Eine verschärfte Version der Standard-Passwörter sind versteckte Passwörter. Damit sind Passwörter gemeint, die vom Hersteller einer Software oder einer Firmware in ein System eingebracht werden. Diese versteckten Passwörter dienen entweder der Sicherstellung von Funktionalitäten unabhängig von den Benutzereinstellungen oder auch dem „Notfall", um bei vergessenem Passwort das System in einen bekannten Ausgangszustand überführen zu können. Nicht selten haben diese versteckten Zugänge volle Berechtigung für das System, so dass sich sämtliche Informationen auslesen und alle verfügbaren Funktionen ausführen lassen. Die beiden Hauptprobleme bei den versteckten Passwörtern liegen in der Unwissenheit des Administrators, der über die „Zusatzfunktionalität" des Systems nicht informiert wird oder ist sowie in der Machtlosigkeit des Administrators, der in vielen Fällen nur wenig gegen die versteckten Passwörter unternehmen kann. Am stärksten davon betroffen sind die „Closed-Source" Software-Produkte. Durch den unter Verschluss gehaltenen Quellcode ist die Wahrscheinlichkeit der Entdeckung für die versteckten Passwörter vergleichsweise gering. In OpenSource Software-Produkten bleiben derartige Geheimnisse nicht lange unentdeckt. Außerdem kann der Administrator im Zweifelsfall den Quellcode verändern und die versteckten Passwörter aus dem System entfernen.

Gelangt ein Angreifer in den Besitz von versteckten Passwörtern, so ist die Bedrohung als mindestens genauso schwerwiegend einzustufen, wie bei den Standard-Passwörtern. Einen entscheidenden Einfluss auf die Stärke dieser Bedrohung hat zusätzlich die Kenntnis des Administrators von den versteckten Passwörtern.

Kein Passwort. Zum Abschluss soll hier noch ein Sachverhalt angesprochen werden, der im engsten Sinne gar kein Angriff gegen ein Passwort ist. Manche Systeme besitzen im Auslieferungszustand nicht einmal das oben beschriebene Standard-Passwort. Eine alternative Möglichkeit, dem Benutzer den ersten Zugriff nach der Auslieferung zu garantieren, besteht im gänzlichen Verzicht

auf ein Passwort. Zwar wird meistens auch hier dem Benutzer nach der ersten Anmeldung das Setzen eines Passwortes empfohlen, nicht selten wird darauf aber verzichtet – beispielsweise aus Bequemlichkeit. Auch in vielen anderen Fällen entscheiden sich Benutzer gegen die Verwendung eines Passwortes – selbst wenn sie explizit dazu aufgefordert werden. Rein rechtlich gesehen fällt es unter solchen Umständen sogar schwer, nach einem entsprechenden Vorfall von einem ‚Einbruch' in das System zu sprechen.

Angriffe gegen Verschlüsselungsalgorithmen

Um elektronisch übermittelte Informationen vor dem Zugriff durch Unbefugte zu schützen, bedient man sich im Normalfall eines der zahlreichen Verschlüsselungsalgorithmen. Einzelne Nachrichten können leicht mit asymmetrischen Verschlüsselungsverfahren wie Pretty Good Privacy (PGP) [155] oder Secure Multipurpose Internet Mail Extensions (S/MIME) [198] vor unbefugtem Zugriff geschützt werden. Beide Verfahren basieren auf der Public-Key-Kryptographie, die an dieser Stelle kurz erläutert werden soll.

Beim Public-Key-Verfahren besitzt jeder an der Kommunikation beteiligte Anwender zwei verschiedene Schlüssel: einen öffentlichen und einen geheimen Schlüssel. Wie die Bezeichnungen leicht vermuten lassen, kann der öffentliche Schlüssel jedem anderen Benutzer bekannt gemacht werden, während der geheime Schlüssel niemandem bekannt gemacht werden darf. Da diese beiden Schlüssel asymmetrisch sind, arbeiten sie nicht wie „normale" Türschlüssel. Vielmehr lässt sich etwas, das mit dem einen Schlüssel verschlossen wurde, ausschließlich mit dem anderen Schlüssel wieder aufschließen. Ebenso muss etwas, das mit dem einen Schlüssel aufgeschlossen werden soll, genau mit dem anderen Schlüssel verschlossen werden.

Wenn nun in einem Beispiel zwei Personen – die in der Kryptographie gerne *A*lice und *B*ob genannt werden – eine Nachricht austauschen wollen, dann kann mit den beiden asymmetrischen Schlüsseln mehreres erreicht werden:

1. Alice kann Bob eine verschlüsselte Nachricht schicken. Sie benötigt dafür den öffentlichen Schlüssel von Bob, mit dem sie die Nachricht kodiert. Da die Schlüssel asymmetrisch arbeiten, existiert nur genau ein einziger Schlüssel, mit dem diese Nachricht wieder dekodiert werden kann: der geheime Schlüssel von Bob. Da nur Bob diesen Schlüssel besitzt, kann auch nur er die Nachricht entschlüsseln.
2. Auf die gleiche Weise kann Bob eine verschlüsselte Nachricht an Alice senden. Dazu benötigt er nur ihren öffentlichen Schlüssel.

Neben der Verschlüsselung kann mit dem Public-Key-Verfahren aber noch eine andere Aufgabe durchgeführt werden, nämlich das Signieren. Hierbei geht es nicht darum, eine Nachricht zu verbergen, sondern es geht darum, den Absender der Nachricht eindeutig zu identifizieren.

3. Bob kann Alice eine Nachricht schicken, so dass Alice zweifelsfrei Bob als den Absender der Nachricht identifizieren kann. Dazu verschlüsselt Bob die Nachricht mit seinem geheimen Schlüssel. Da die Schlüssel asymmetrisch arbeiten, existiert nur genau ein einziger Schlüssel, der diese Nachricht wieder dekodieren kann: der öffentliche Schlüssel von Bob. Da dieser Schlüssel allen – und damit auch Alice – bekannt ist, kann Alice sicherstellen, dass nur Bob diese Nachricht gesendet haben kann.
4. Auch Alice kann Bob eine Nachricht senden, die er zweifelsfrei ihr als Absenderin zuordnen kann. Alice verschlüsselt dazu die Nachricht mit ihrem geheimen Schlüssel. Wenn Bob diese Nachricht mit dem öffentlichen Schlüssel von Alice dekodieren kann, so muss die Nachricht mit dem geheimen Schlüssel von Alice kodiert worden sein – und den besitzt nur Alice.

Obwohl jeweils das Wort „Verschlüsseln“ verwendet wurde, so sollte doch klar sein, dass signierte Nachrichten für jeden lesbar sind. Das liegt daran, dass signierte Nachrichten mit dem öffentlichen Schlüssel des Absenders wieder entschlüsselt werden können – und den kennt im Zweifelsfall jeder.

Sowohl PGP als auch S/MIME basieren auf der Public-Key-Kryptographie. Der Hauptunterschied zwischen diesen beiden Verfahren liegt in der zugrunde liegenden Schlüsselinfrastruktur (Public Key Infrastructure, PKI). Während man bei PGP selbst dafür sorgen muss, seinen öffentlichen Schlüssel vertrauenswürdig an alle Kommunikationspartner zu übermitteln, so liegt bei S/MIME eine vertrauenswürdige Zertifizierungsinstanz (Certification Authority, CA) im Hintergrund, über welche die Echtheit von öffentlichen Schlüsseln überprüft und bestätigt werden kann.

Wenn nicht nur einzelne Nachrichten, sondern der gesamte Datenkommunikationsweg verschlüsselt werden soll, bietet sich ein Virtual Private Network (VPN) an. Ein VPN ermöglicht es, beliebige IP Pakete verschlüsselt zwischen zwei bekannten Endpunkten auszutauschen. Speziell dafür entwickelt wurde des IP Security (IPSec) Protokoll. Für eine genauere Betrachtung des IPSec Protokolls sei an dieser Stelle auf [104] verwiesen. Wichtig ist an dieser Stelle nur, dass zum kodierten Datenaustausch Schlüssel eine entscheidende Rolle spielen.

In jedem Fall hängt der erreichte Grad an Sicherheit einer Verschlüsselung sowohl von der Güte des Verschlüsselungsalgorithmus als auch von der Schlüssellänge ab. Für die oben genannten Verfahren können moderne Verschlüsselungsalgorithmen mit Schlüssellängen von 1024 Bit und weit darüber hinaus eingesetzt werden. Zumindest im Moment können derartige Kodierungen mit keiner vorhandenen Maschine in vertretbarer Zeit entschlüsselt werden und können somit als sicher eingestuft werden.

Brute-Force Angriffe. Mit der ständig steigenden Performanz heutiger Rechnersysteme werden so genannte Brute-Force Angriffe[7] gegen verschlüs-

[7] „brute force“ heißt übersetzt soviel wie „rohe Gewalt“

selte Datenströme immer erfolgversprechender. Durch simples Ausprobieren aller möglichen Schlüssel errät man zwangsweise auch den richtigen Schlüssel, der eine Nachricht dekodiert. Die Organisation DISTRIBUTED.NET [62] beschäftigt sich mit dem verteilten Bearbeiten von Computerproblemen über ein Netzwerk – dem Internet. Als eine besonders herausfordernde Aufgabe betrachtet DISTRIBUTED.NET das Entschlüsseln von Nachrichten, die mit einem unbekannten Schlüssel eines bekannten Verschlüsselungsalgorithmus kodiert wurden. Zum Einsatz kommt dabei eine Netzwerk-Software auf Client-Server Basis, die an alle Interessierten im gesamten Internet verteilt wird. Somit steht für die rechenintensive Aufgabe eine nahezu unerschöpfliche Anzahl von Client-Rechnern zur Verfügung, die alle gemeinsam an der Entschlüsselung der gegebenen Nachricht arbeiten.

DISTRIBUTED.NET hat in den vergangenen Jahren mehrere Projekte erfolgreich durchgeführt. Der Data Encrypting Standard (DES) Algorithmus [48] mit einer Schlüssellänge von 56 Bit wurde bereits mehrfach geknackt, indem der gesamte Schlüsselraum bei einem Brute-Force Angriff abgesucht wurde. Im Januar 1999 konnte DISTRIBUTED.NET im *DES III* Projekt den mit einer Schlüssellänge von 56 Bit kodierten Text in etwas mehr als 22 Stunden knacken. Andere Projekte zielten auf den RC5 (Rivest Cipher Version 5) [177] Algorithmus mit verschiedenen Schlüssellängen. Der 56 Bit lange Schlüssel wurde 1997 nach 250 Tagen durch systematisches Absuchen des gesamten Schlüsselraumes erraten. Eine mit einem 64 Bit langen Schlüssel kodierte Nachricht ist im Projekt *RC5-64* ebenfalls bereits geknackt worden. Es wurden allerdings nahezu 5 Jahre benötigt, bis im Jahr 2002 der korrekte Schlüssel gefunden wurde. Seit dem arbeitet DISTRIBUTED.NET am Projekt *RC5-72* in der Hoffnung, den 256-mal größeren Schlüsselraum dank immer schnellerer Hardware in vergleichbarer oder kürzerer Zeit zu finden.

Ein zu überwindendes Problem für die Hersteller findet sich im Regelfall im hohen Performance-Verlust, der beim Verschlüsseln einer Nachricht oder einer ganzen Kommunikationsstrecke unweigerlich auftritt. Je größer der verwendete Schlüssel dabei ist, desto länger benötigt auch die Software für die Operation. Moderne Verfahren gehen deshalb dazu über, die eigentliche Verschlüsselung durch speziell dafür entwickelte Hardware ausüben zu lassen. Eine Vielzahl von verschlüsselten VPN Daten-Tunneln lässt sich sternförmig von so genannten Konzentratoren kontrollieren, die fast ausschließlich die Aufgabe haben, die jeweils durch den Tunnel zu sendenden Daten zu verschlüsseln und die empfangenen Daten zu entschlüsseln.

Angriffe gegen Protokolle. Nicht immer müssen zum Entschlüsseln einer kodierten Nachricht alle möglichen Schlüssel ausprobiert werden. Manchmal weisen die verwendeten Verschlüsselungsalgorithmen bereits Implementierungsfehler auf, die ein Angreifer systematisch ausnutzen kann. Bekanntestes Beispiel ist vielleicht der Wired Equivalent Privacy (WEP) Verschlüsselungsstandard [214] des Wireless Local Area Network (WLAN) Standards [74]. Dieser gilt gemein hin als unsicher. Bereits 2001 wurden einige Schwachstellen von

WEP aufgedeckt, die ein Entschlüsseln der drahtlos übertragenen Nachrichten mittels statistischer und anderer Methoden innerhalb vergleichsweise kurzer Zeit ermöglichen. Die Unsicherheit von WEP entsteht im Wesentlichen durch den so genannten Initialisierungsvektor (IV), der nur aus einem 24-Bit Wert besteht und zudem noch fragwürdig (durch ungeschicktes Zurücksetzen auf Null) verwendet wird [20]. Nach spätestens etwa 16 Millionen übermittelten Paketen wird sich der Initialisierungsvektor zwangsweise wiederholen müssen, was bei einem ausgelasteten WLAN Access-Point bereits nach nur wenigen Stunden der Fall sein kann.

Angriff gegen die Nachricht. Das klingt zunächst etwas befremdlich, aber eine unverschlüsselte Nachricht kann zur Berechnung des anschließend verwendeten Schlüssels verwendet werden. Einfach gesprochen: Wenn man die ursprüngliche Nachricht und den verwendeten Algorithmus kennt, dann lässt sich eventuell aus der kodierten Nachricht der benutzte Schlüssel errechnen. Abhängig von der Kenntnis der Klartextnachricht, der verschlüsselten Nachricht und dem Verschlüsselungsalgorithmus unterscheidet man insgesamt sechs verschiedene Angriffsmethoden auf Nachrichten:

1. Beim *Known Ciphertext Attack* versucht man – allein aus der Kenntnis der verschlüsselten Nachricht heraus – den Schlüssel zu erraten. Dieser Angriff ähnelt dem Brute-Force-Angriff, wobei sich der Angreifer jedoch zusätzlich noch statistischer Methoden bedienen kann.
2. Der *Known Plaintext Attack* ähnelt dem *Known Ciphertext Attack* mit dem Unterschied, dass beim *Known Plaintext Attack* neben der verschlüsselten Nachricht auch noch der Klartext ganz oder teilweise bekannt sein muss. In den Besitz des Klartextes kann man im schlimmsten Fall auch durch Erraten bestimmter Bereiche der Nachricht gelangen, wie beispielsweise Kopf- oder Fußzeilen, Grußformeln und andere Textbausteine. Besteht die Nachricht aus dem Paket eines Netzwerkprotokolls, so kann manchmal der formalisierte Protokollheader leicht erraten werden. Der Angriff selbst ist vergleichbar aufwändig zum *Known Ciphertext Attack*.
3. Beim *Chosen Plaintext Attack* hat der Angreifer die Möglichkeit, die Klartextnachrichten selbst zu bestimmen. Auf diese Weise kann er durch geschickte Wahl von Nachrichten den verwendeten Schlüssel möglicherweise einfacher erraten.
4. Der *Adaptive Chosen Plaintext Attack* ist eine Verfeinerung des *Chosen Plaintext Attack*. Bei dieser Angriffsform kann der Angreifer nicht nur die Klartextnachricht selbst bestimmen, sondern er hat zusätzlich freien Zugriff auf den Verschlüsselungsalgorithmus. Auf diese Weise kann der Angreifer durch adaptives Anpassen der Klartextnachrichten sukzessiv den Schlüssel errechnen.
5. Beim *Chosen Ciphertext Attack* benötigt der Angreifer keinen freien Zugang zur Klartextnachricht oder dem Verschlüsselungsalgorithmus. Der Angreifer hat lediglich die Möglichkeit, eine Nachricht frei zu wählen, die anschließend dekodiert wird.

6. Der *Adaptive Chosen Ciphertext Attack* erweitert den *Chosen Ciphertext Attack* um die Möglichkeit für den Angreifer, Nachrichten nach Belieben entschlüsseln zu können. Durch ein adaptives Vorgehen kann so der Schlüssel zielgerichtet erraten werden. Ein Beispiel hierfür ist eine signierte Nachricht, deren gesamter Inhalt mit dem geheimen Schlüssel einer PKI kodiert wird. In diesem Fall besitzt der Angreifer die Möglichkeit, den verschlüsselten Teil mit dem öffentlichen Schlüssel zu dekodieren.

Für eine detailliertere Betrachtung dieses Problems soll an dieser Stelle auf die zahlreich vorhandene weiterführende Literatur verwiesen werden [189, 127, 197, 25].

Ein Beispiel für einen Angriff gegen Nachrichten ist gegen den geheimen Schlüssel von Benutzern bei den Verfahren PGP und S/MIME gerichtet. Dabei sorgt der Angreifer dafür, dass der Anwender eine dem Angreifer bekannte Nachricht signiert. In Kenntnis sowohl der unverschlüsselten Ausgangsnachricht als auch der signierten Nachricht ist der Angreifer möglicherweise in der Lage, den geheimen Schlüssel berechnen zu können. Moderne PKI-Systeme schützen sich vor dieser Bedrohung, indem sie nicht die gesamte Nachricht signieren, sondern lediglich einen Hashwert der Nachricht. Die oben angegeben Angriffsvarianten sind deshalb nahezu wirkungslos. Das Bekanntwerden des geheimen Schlüssels einer PKI hat fatale Folgen. Die Vertraulichkeit kann nämlich nur durch ein komplett neues Schlüsselpaar wiederhergestellt werden. Bei der Public Key Infrastructure von S/MIME wird man dabei durch die Zertifizierungsinstanz unterstützt, die neben der Ungültigkeit des alten auch direkt den neuen Schlüssel propagiert. Im Falle von PGP ist dies mangels einer zentralen Instanz nicht so einfach möglich.

Automatisierte Angriffe

Die täglich wachsende, kaum mehr überschaubare Anzahl von Rechnern und die vielen auf diesen Geräten laufenden Dienste haben zusammen mit den grundlegenderen Netzwerkkomponenten wie Router und Switches das Internet zu einem hochgradig nichtlinearen und dynamischen System gemacht. Zwar lassen sich mit einem einzelnen Datenpaket weder das Internet noch die daran angeschlossenen Systeme nachhaltig beeinflussen, jedoch kann eine größere Anzahl von Paketen in besonderen Fällen einen deutlichen Einfluss ausüben. Als Beispiel seien hier die Denial of Service (DoS) oder die etwas effektiveren Distributed Denial of Service (DDoS) Angriffe genannt. In beiden Fällen werden Datenpakete koordiniert an ein einzelnes Empfängersystem geleitet. Bei den DDoS Angriffen wird hierzu eine größere Anzahl von Rechnern dazu gebracht, ein vordefiniertes Ziel möglichst zeitgleich anzusprechen. Das Zielsystem wird damit überlastet und steht für andere Anwender nur noch bedingt oder gar nicht mehr zur Verfügung. Der entscheidende Aspekt liegt

in der zeitlichen und örtlichen[8] Synchronisation aller Datenpakete. Ein solcher Angriff ist manuell nur sehr schwer und aufwändig umsetzbar, obwohl die Anonymous Digital Coalition Anfang 1998 einen erfolgreichen „Netzstreik" oder auch „virtuellen Sit-In" bei fünf mexikanischen Banken durchführen konnte. Grundlage war ein öffentlich verbreiteter Aufruf [4], der die Menschen zu einem koordinierten Besuch der Webseiten dieser Banken motivierte. Vermutlich als Folge davon waren die Webseiten der genannten Banken für einige Zeit nicht oder zumindest nur schlecht erreichbar.

Es muss sich aber nicht immer zwangsläufig um einen Angriff handeln, wenn weltweite Aktionen synchronisiert ablaufen. Der im September 1999 veröffentlichte Starr-Report um die Affäre des damaligen amerikanischen Präsidenten Bill Clinton [86] verursachte durch das rege Interesse der Öffentlichkeit eine ähnliche Überlastung einzelner Webseiten – wenn auch das Internet als Ganzes nicht unter der Last zusammengebrochen war.

Eine deutlich einfacher zu realisierende Angriffsform, bei der Rechner aus aller Welt zeitgleich eine zielgerichtete Aktion ausführen, um die Erreichbarkeit eines Servers und dessen Diensten zu verschlechtern oder zu verhindern, ist der bereits angesprochene DDoS Angriff. Hier lassen sich vor allem grundlegende Techniken der Netzwerkinfrastruktur und der jeweiligen Netzwerkprotokolle ausnutzen. Sendet man beispielsweise mit einer gefälschten Absender IP Adresse einen Broadcast[9], so antworten typischerweise alle Rechner des angesprochenen Netzwerkes an die vorgetäuschte Adresse. Das können bei einem Klasse-B Netzwerk $2^{16} - 2 = 65534$ und bei einem Klasse-A Netzwerk rein theoretisch sogar bis zu $2^{24} - 2 = 16777214$ Rechner sein. Wiederholt man die Broadcast-Pakete in schneller Folge – eventuell auch an Broadcast-Adressen mehrerer Netzwerke – so wird der Zielrechner mit Antwortpaketen überflutet und dadurch schlechter erreichbar.

Um derartige automatisierte Angriffe zu vermeiden, reichen bereits einfache Überprüfungen der IP Pakete aus, die in das Internet eingespeist werden. ISPs haben sich deshalb auf eine bessere Umsetzung der so genannten Ingress Filterung geeinigt [66]. Werden bereits beim Übergang vom Kundennetzwerk zum Backbone des ISP Plausibilitätsüberprüfungen der Absender IP Adresse durchgeführt (Anti-Spoofing), so können bereits einige der Angriffe im Keim erstickt werden. Selbstverständlich können durch korrekte Ingress Filterung keine DoS Angriffe unterbunden werden, in denen ein einzelner Rechner mit breitbandiger Internet-Anbindung die Erreichbarkeit eines Zielrechners mit

[8] Unter dem Ort ist hier die IP Adresse eines Systems gemeint. Diese hängt zunächst nur wenig mit dem tatsächlichen Ort des Systems zusammen, selbst wenn die IP Adressen weltweit in verschiedene Regionen unterteilt sind.

[9] Ein Broadcast ist ein IP-Paket, das an die spezielle Broadcast-Adresse eines Netzwerkes gesendet wird. Bei einem Klasse-C Netzwerk (also beispielsweise 192.168.17.0/24) können 254 der 256 möglichen Adressen für Rechner verwendet werden (also 192.168.17.1 - 192.168.17.254). Die spezielle Adresse 192.168.17.0 bezeichnet das Netzwerk, während die ebenfalls spezielle Adresse 192.168.17.255 die Broadcast-Adresse darstellt.

schlechterer Netzanbindung durch gültige IP Pakete verhindert. Allerdings ist diese Art von Angriff auch sehr leicht ausfindig zu machen und wirkungsvolle Gegenmaßnahmen sind vergleichsweise einfach zu implementieren.

9.3 Angriffsziele

Die in diesem Kapitel beschriebenen Bedrohungen, die durch eine oder mehrere der aufgelisteten Angriffsarten in eine unmittelbare Gefahr überführt werden, richten sich entweder gegen Nutzdaten oder die zur Haltung und zum Transport der Daten notwendigen Infrastruktur. Es existieren allerdings auch Angriffe, bei denen die Zielsetzung eine andere ist. Die Hintergründe dieser ungerichteten Angriffe sollen in diesem Abschnitt ebenfalls beleuchtet werden.

9.3.1 Angriffe auf Nutzdaten

Informationen, die ein Netzwerk nur als Transportmedium nutzen und rein theoretisch auch auf einem anderen Weg übermittelt werden könnten, werden im Weiteren als Nutzdaten bezeichnet. Hierunter fallen beispielsweise Webseiten, die über das Hypertext Transfer Protocol (HTTP) [68] übermittelt werden, oder auch Dateien, die über das File Transfer Protocol (FTP) [164] übertragen werden sowie elektronische Nachrichten, die über die Protokolle Simple Mail Transfer Protocol (SMTP), Post Office Protocol Version 3 (POP3) oder Internet Message Access Protocol (IMAP) transferiert werden. In diesem speziellen Fall sind die aufgeführten Beispiel-Protokolle Bestandteil der TCP/IP Protokollfamilie und befinden sich in der vierten Schicht (Transportschicht) des OSI Referenzmodells. Dies bedeutet gleichzeitig, dass die Nutzdaten auf diese und die drei darunterliegenden OSI-Schichten aufbauen, welche für einen reibungslosen und gesicherten Transport der Daten zuständig sind. Die Nutzdaten sind allerdings so definiert, dass sie keinesfalls vom Transportmedium abhängen. So könnte anstelle des Übertragungsweges via FTP auch ein Transport über Datenträger wie Disketten stattfinden, ohne dass die Daten dabei an Bedeutung verlieren.

Die in Abschnitt 9.1 beschriebenen Bedrohungsarten lassen sich unmittelbar auf das Beispiel einer E-Mail anwenden, deren Inhalt verloren geht, verändert wird, veröffentlicht wird oder deren Herkunft nur vorgetäuscht ist. Informationen können für unterschiedliche Gruppen einen jeweils anderen Wert besitzen. Dementsprechend verbergen sich in vielen Fällen finanzielle Motive hinter Angriffen auf Nutzdaten[10].

Auch automatisierte Angriffe können gegen Nutzdaten gerichtet sein. Ein gutes Beispiel hierfür ist der Internet-Wurm „Witty“ [193]. Dass dieser Wurm

[10] Dies betrifft insbesondere den Diebstahl von Informationen. Die Veröffentlichung von Informationen kann durchaus anderweitig motiviert sein; politische oder moralische Gründe sind nur zwei Beispiele.

gegen Nutzdaten gerichtet war, erkennt man vor allem an der destruktiven Nutzlast, welche die Festplatten befallener Rechner sukzessive löscht. Witty ist bislang in vielerlei Hinsicht einzigartig:

- Durch die Zerstörung des befallenen Systems hat sich der Wurm selbst wieder vernichtet.
- Die vergleichsweise kurze Zeitspanne zwischen Bekanntwerden der zugrunde liegenden Schwachstelle und der Verbreitung des Wurms von nur einem Tag zeigt ganz deutlich, dass auch das prompte Einspielen von Sicherheits-Updates nicht vor der Ausnutzung von Verwundbarkeiten schützt.
- Witty richtete sich ausschließlich gegen eine spezielle Firewall-Software, die eigentlich einen Schutz vor Angriffen bieten sollte.
- Die vergleichsweise schnelle Ausbreitung des Wurms basiert nicht nur auf einem ausgeklügelten Algorithmus, sondern auch auf der bislang neuen Technik einer voreingestellten Hitliste; der Wurm wurde nicht auf einem einzelnen Rechner initial gestartet, sondern auf einer Basismenge von etwa 100 Rechnern gleichzeitig.

9.3.2 Angriffe auf die Infrastruktur

Das zweite große Angriffsziel ist die Infrastruktur – oder im globalen Zusammenhang das Internet selbst. Konkretes Ziel kann die Hardware sein, die direkt angegriffen wird, oder aber auch Pakete, die für das Funktionieren eines Netzwerkes unabdingbar sind. Die Nachrichten des Internet Control Message Protocol (ICMP) erfüllen allesamt Aufgaben zur Steuerung des Informationsflusses im Netzwerk. Tabelle 9.2 listet einige der verschiedenen ICMP Nachrichtentypen auf, die ihrerseits noch in mehrere Codes unterteilt sein können.

Viele der automatisierten Angriffe erzeugen in erster Linie eine enorme Netzwerklast, welche bei den beteiligten Geräten einen DoS erzeugt. Als Folge daraus sind die Geräte nicht mehr erreichbar. Ein gutes Beispiel stellt der Wurm „Sapphire“ [225] dar, der auch unter dem Namen „Slammer“ bekannt wurde. Sapphire gilt als der bislang schnellste Internet-Wurm; nach nur etwa 10 Minuten hatte sich der Wurm bereits weltweit ausgebreitet. Für diese rasante Ausbreitungsgeschwindigkeit existieren im Wesentlichen zwei Hauptgründe:

1. **Die Größe der Pakete**
   Sapphire benötigt zur Übertragung und Infizierung eines anderen Rechners exakt 376 Byte Nutzdaten. Diese können in einem einzigen Paket versendet werden.
2. **Die Art der Pakete**
   Der Wurm nutzt eine Schwachstelle aus, bei welcher die Schadroutine über den Port 1434 des User Datagram Protocols (UDP) übertragen wird. UDP

Tabelle 9.2. Nachrichtentypen und Codes von ICMP.

Type	Code	Beschreibung
0	0	Echo Reply
3	0	Network Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Datagram too Big
	5	Source Route Failed
	6	Destination Network Unknown
	7	Destination Host Unknown
	9	Destination Network Administratively Prohibited
	10	Destination Host Administratively Prohibited
	11	Network Unreachable for TOS
	12	Host Unreachable for TOS
	13	Communication Administratively Prohibited
	14	Host Precedence Violation
	15	Precedence Cutoff in Effect
4	0	Source Quench
5	0	Redirect for Network Error
	1	Redirect for Host Error
	2	Redirect for TOS and Network Error
	3	Redirect for TOS and Host Error
8	0	Echo Request
9	0	Normal Router Advertisement
	1	Does not Route Common Traffic
10	0	Router Solicitation
11	0	Time Exceeded
	1	Fragment Reassembly Timeout
12	0	Parameter Problem: IP Header Invalid
	1	Parameter Problem: Required Option Missing
13	0	Timestamp Request
14	0	Timestamp Reply
15	0	Information Request
16	0	Information Reply
17	0	Address Mask Request
18	0	Address Mask Reply
30	0	Traceroute

ist im Gegensatz zu TCP ein verbindungsloses Protokoll und benötigt daher keinen Drei-Wege-Verbindungsaufbau (Three-Way-Handshake). Sapphire kann sich also durch das Versenden eines einzelnen Paketes ausbreiten, ohne auf Antwortpakete warten zu müssen.

Diese beiden Gründe haben im Januar 2003 dazu geführt, dass die Ausbreitungsgeschwindigkeit des Wurms sich bereits nach kurzer Zeit nicht weiter erhöhen konnte, weil sie durch die im ganzen Internet zur Verfügung stehende Bandbreite begrenzt wurde. Durch eine Verdopplung der Ausbreitungsgeschwindigkeit in jeweils weniger als 9 Sekunden trat diese Sättigung bereits nach nur drei Minuten ein. Zu diesem Zeitpunkt wanderten etwa 55 Millionen Sapphire-Pakete pro Sekunde über das Internet. Die Folge war ein massiver Zusammenbruch des Internets. Zwar dauerte die aktivste Phase des Wurms nur wenige Stunden an, dennoch hatte der Wurm weltweit eine negative Auswirkung auf Internet-basierte Systeme aller Art.

Einige der Pakete, die an dieser Stelle zur Infrastruktur gezählt werden, können äußerlich auch die Form von Nutzdaten annehmen. Zur Verdeutlichung sollen die beiden Routing Protokolle Open Shortest Path First (OSPF) [133] und Routing Information Protocol (RIP) [82] herangezogen werden. Beide Protokolle sind für den Austausch dynamischer Routing Informationen innerhalb eines Autonomen Systems (AS) zuständig; es werden jedoch jeweils andere Verfahren eingesetzt. Auf dynamisches Routing kann vor allem in größeren Netzwerken nicht verzichtet werden, da es einen automatischen Mechanismus zur Fehlererkennung und -umgehung bietet[11]. Das Internet besteht wiederum aus vielen Autonomen Systemen, die über externe Routing Protokolle miteinander verknüpft sind. Erst durch eine mehrfach redundante Vernetzung aller Komponenten ist das Internet überhaupt in der Lage, die ständig auftretenden Störungen größtenteils vor den Nutzern zu verbergen. Im Folgenden werden beispielhaft die vier vorgestellten Bedrohungsarten Informationsverlust, Informationsvortäuschung, Informationsverfälschung und das Bekanntwerden von Informationen auf die beiden dynamischen Routingprotokolle angewendet.

Beispiel: RIP

Beim RIP Protokoll [82] und seinem Nachfolger RIP-2 [112] teilen – vereinfacht gesagt – die Router alle Informationen, die sie über das gesamte Netzwerk (das Autonome System) haben, mit ihren direkten Nachbarn. Dies geschieht auf direkte Anfrage eines benachbarten Routers und zusätzlich in regelmäßigen Abständen an alle benachbarten Router. Durch die Angabe von Metriken für jedes einzelne Ziel kann sich jeder Router sein eigenes Bild vom Netzwerk in seiner Routing-Tabelle aufbauen. Die Metriken geben dabei an, über wie viele andere Router ein entsprechendes Ziel zu erreichen ist. An dieser

[11] Selbstverständlich bieten dynamische Routing Protokolle keine Fehlerbehebung; defekte Hardware muss immer noch manuell ersetzt werden.

Stelle soll nicht weiter auf die ausgeklügelten Algorithmen eingegangen werden, die zirkuläre Routen und andere Widrigkeiten verhindern sollen. Wichtig ist lediglich, dass durch einen einzelnen Router das gesamte Autonome System beeinflusst werden kann. Im Speziellen existieren bei RIP die Bedrohungen der Informationsvortäuschung und der Informationsverfälschung. In beiden Fällen verbreiten sich die falschen Daten sukzessiv an alle anderen Router das Autonomen Systems.

Informationsvortäuschung. In diesem Szenario sendet ein Router Informationen über die vorgetäuschte Erreichbarkeit zusätzlicher Netzwerkbereiche. Auf diesem Weg kann der Router Pakete an sich ziehen, die anderenfalls mangels Route an ein vorkonfiguriertes Gateway zur Übergabe an ein anderes Autonomes System geleitet worden wären. Ein Angreifer kann zur Ausnutzung dieser Bedrohung entweder einen eigenen Router im Autonomen System platzieren, welcher die zusätzlichen Informationen verbreitet. In diesem Fall handelt es sich um die Vortäuschung von Daten bei der Übermittlung. Ein Angreifer kann aber auch einen vorhandenen Router manipulieren, so dass dieser Informationen über die zusätzlichen Netzwerke verbreitet. In diesem Fall werden statische Daten vorgetäuscht.

Informationsverfälschung. Im Fall der Informationsverfälschung sendet ein Router Informationen mit einer geringeren Metrik für eine Erreichbarkeit von bereits vorhandenen Netzwerkbereichen. Wiederum werden unberechtigt Pakete angezogen, allerdings nur auf Grund der geringeren Metrik. Unabhängig davon, ob der Angreifer einen eigenen Router im Autonomen System platziert oder ob er einen vorhandenen Router manipuliert, findet eine Informationsverfälschung der statisch auf den benachbarten Routern gespeicherten Daten statt.

Beispiel: OSPF

Das Open Shortest Path First (OSPF) [133] Routing Protokoll verfolgt wie RIP das Ziel der Informationsverbreitung von Routen im Netzwerk. Anders als bei RIP werden von einem OSPF-Router aber nicht Informationen über das gesamte Netzwerk verbreitet, sondern nur Angaben über die jeweiligen Nachbarn des Gerätes. Damit dennoch jeder Router sich sein eigenes Bild vom Netzwerk machen kann, werden diese Informationen über die Nachbarn an alle anderen Router des Autonomen Systems gesendet. Wiederum seien an dieser Stelle nicht die Algorithmen von Interesse, die zirkuläre Routen und andere Probleme verhindern sollen. Der Fokus richtet sich vielmehr wieder auf die Tatsache, dass ein einzelner Router das gesamte Autonome System beeinflussen kann. Bei OSPF lassen sich insbesondere die Bedrohungen des Informationsverlustes und des Bekanntwerdens von Informationen verdeutlichen.

Bekanntwerden von Informationen. Da bei OSPF-Routern die Informationen über die Nachbarn eines jeden Routers über das ganze Autonome

System hinweg verteilt werden müssen, wird es dadurch auch jedem einzelnen Rechner im Netzwerk ermöglicht, sich dasselbe Bild über die Netzwerkstruktur zu machen. Somit werden Informationen über das Netzwerk-Design öffentlich zugänglich. Dieser Umstand wird dadurch erschwert, dass OSPF keinerlei Verschlüsselungsmechanismen aufweist.

Informationsverlust. Die Gefahr des Informationsverlustes ist bei OSPF größer als bei RIP, da ja die Informationen aus den Routing-Tabellen über die direkten Nachbarn hinaus an das gesamte Netzwerk verteilt werden müssen. Gelingt es einem Angreifer, die Nachrichten eines oder mehrerer Router zu unterdrücken, so verlieren unter ungünstigen Bedingungen Teile des Autonomen Systems die Kenntnis über einen tatsächlich vorhandenen Netzwerkbereich, der dann von dort aus nicht mehr erreichbar scheint. Dieser Angriff basiert auf dem Informationsverlust von gesendeten Daten.

9.3.3 Ungerichtete Angriffe

Hinter den ungerichteten Angriffen verbergen sich im Wesentlichen die automatisierten Angriffe, bei denen oftmals ganz andere Motive vorherrschen. Als vielzitiertes Beispiel sollen hier die so genannten „Script-Kiddies“ herangezogen werden. Gemeint sind Personen – oft jüngeren Alters – die nicht zwangsweise über besondere Fähigkeiten und Talente im Bereich der Netzwerktechnik oder Informationstechnik im Allgemeinen verfügen müssen. Die große Effektivität ihrer Angriffe resultiert aus der Tatsache, dass die vorhandenen automatisierten Angriffswerkzeuge immer einfacher zu bedienen sind. Bereits durch wenige Mausklicks versuchen die Script-Kiddies manchmal sogar mit Erfolg Berühmtheit zu erlangen. Beinahe tragisch mag da der Fall des Autors der beiden sehr berühmten Würmer „Netsky“ und „Sasser“ sein. Der ursprüngliche Beweggrund zur Programmierung der beiden Würmer mit allen ihren Untervarianten lag nicht im Anrichten von Schaden. Ganz im Gegenteil war Netsky darauf programmiert, zwei andere schädliche Würmer „Mydoom“ und „Bagle“ von infizierten Rechnern zu entfernen [204]. Allerdings haben charakterliche Unreife gepaart mit Schwächen in den Programmierkenntnissen beim Autor von Sasser zu einem unerwünschten Nebeneffekt geführt. Die ungeplante explosionsartige Verbreitung der verschiedenen Varianten von Sasser haben schließlich weltweit für hohe Schäden gesorgt, da der Wurm – einmal in Umlauf gebracht – nicht mehr zu stoppen war.

Das Beispiel Sasser hat gezeigt, dass es durchaus verschiedene Beweggründe für einen ungerichteten Angriff geben mag. Entscheidend für den Betreiber eines Netzwerkes sind jedoch lediglich die möglichen Schäden, welche der Angriff anrichten kann sowie die Identifikation und Implementierung von Methoden zum Schutz vor dem Angriff.

10

Auswirkungen auf das Netzwerkmanagement

Die vielen bestehenden Bedrohungen für Netzwerke und die ständigen Angriffe gegen diese machen die Planung und Einrichtung wirksamer Schutzmechanismen unumgänglich. Die Auswirkungen treffen nicht nur ein Netzwerk als Ganzes, sondern vermehrt auch das Netzwerkmanagement. Während auf der einen Seite die Netzwerküberwachung als perfekte Informationsquelle für potentielle Angreifer dienen kann, stellt die Netzwerkkonfiguration das ideale Instrument zur Durchführung von Angriffen dar. Aus diesem Grund präsentiert sich das Netzwerkmanagement als ein lohnendes und wertvolles Ziel für gerichtete Angriffe. Aber auch ungerichtete Angriffe können durch die Übernahme eines Netzwerkmanagementsystems einen effektiven Multiplikator für ihre Aktionen gewinnen. Aus diesem Grund sollte dem Netzwerkmanagement besonderes Interesse zugeteilt werden. Im Folgenden sollen aus den in Kapitel 9 beschriebenen Bedrohungen und Angriffen geeignete Maßnahmen zur Verbesserung der Sicherheit im Netzwerk und insbesondere im Netzwerkmanagement abgeleitet werden.

10.1 Der perfekte Schutz?

Um die Antwort auf die Frage vorwegzunehmen: Den *perfekten* Schutz gibt es nicht. Zwar lassen sich prinzipiell gegen alle Bedrohungen und Angriffe geeignete Gegenmaßnahmen ergreifen, eine absolute Sicherheit lässt sich dennoch nicht erreichen. Es lassen sich allerdings verschiedene Maßnahmen und Vorbereitungen treffen, welche die Wahrscheinlichkeit des Erfolgs eines Angriffs deutlich verringern.

10.1.1 Ein spielerischer Vergleich

In vielerlei Hinsicht ähnelt das Absichern eines Netzwerkes und des Netzwerkmanagements einem Schachspiel: Ein Angriffszug, welcher den König des Gegenübers bedroht, lässt sich gut mit einem Angriff auf das Netzwerk oder

auf Teile des Netzwerks vergleichen. Abbildung 10.1 zeigt die Ausgangssituation in einer Schachpartie mit Schwarz am Zug. An dieser Stelle ist es nur von untergeordneter Bedeutung, ob die gezeigte Stellung besonders sinnvoll erscheinen mag und wer diese Partie vermutlich gewinnen wird. Es ist lediglich wichtig, dass der Spieler mit den schwarzen Figuren in dieser fiktiven Stellung Weiß auf mehreren Arten Schach bieten kann. Eine mögliche Variante wäre der Zug `Lb5+`, den wiederum Weiß auf drei grundlegend verschiedene Arten kontern kann (siehe Abbildung 10.2):

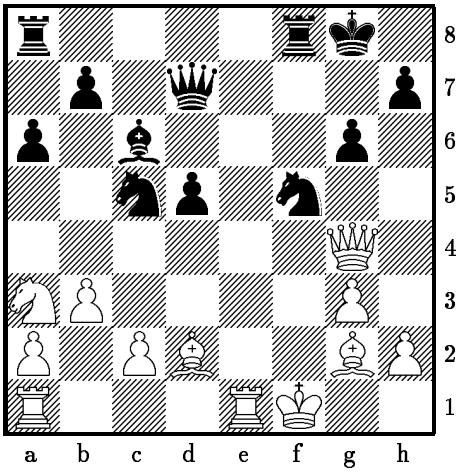

Abb. 10.1. Ausgangssituation in einer fiktiven Schachpartie. Schwarz ist am Zug und hat nun mehrere Zugmöglichkeiten, um Weiß Schach zu bieten.

1. Der Spieler kann die angreifende Spielfigur schlagen. Dazu muss eine seiner eigenen Figuren – zur Not auch der König selbst – in Reichweite der angreifenden Figur stehen.
2. Der Spieler kann eine seiner anderen Figuren zwischen die angreifende Spielfigur und den bedrohten König ziehen. Auch hier muss sich eine passende Figur in Reichweite befinden.
3. Der Spieler kann den König aus der Bedrohung ziehen. Dies setzt voraus, dass noch mindestens ein freies Feld direkt neben dem König existiert, welches der Angreifer nicht bedroht.

Zieht man nun die Parallelen zum Angriff auf ein Netzwerk, so lassen sich schnell Analogien finden. Im ersten Fall schlägt der Spieler, dessen König bedroht wurde, einfach die angreifende Spielfigur. Diese angreifende Figur könnte beispielsweise ein Virus sein, der von einem Virenscanner erkannt und entfernt wird. Der Angriff kann auch von einem schädlichen Paket aus dem

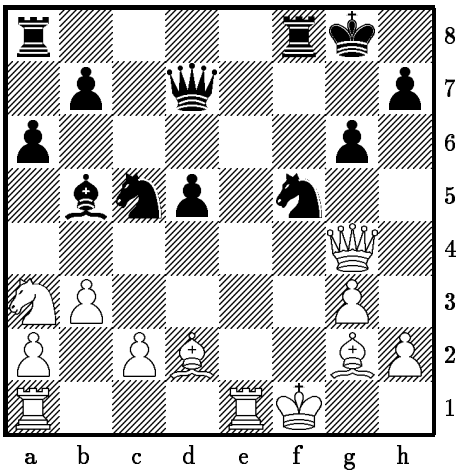

Abb. 10.2. Schwarz hat Weiß mit seinem Zug `Lb5+` Schach geboten und erzwingt damit eine Reaktion. Weiß kann nun entweder 1. den schwarzen Läufer mit seinem Springer schlagen (`S×b5`), 2. eine Figur zwischen den schwarzen Läufer und seinen König ziehen – beispielsweise durch `c2-c4`, oder 3. den König mit `Kg1` oder `Kf2` aus der Bedrohung ziehen.

Internet erfolgen, das von der Firewall erkannt und gelöscht wird. Im zweiten Fall verlagert der angegriffene Spieler den Angriff von seinem König auf eine andere Spielfigur. Das kann man am ehesten mit der Einrichtung eines Application Gateways, also einer Firewall auf der Anwendungsschicht des OSI Referenzmodells. Ein Application Gateway interpretiert sämtliche Pakete und filtert ungültige Anweisungen aus. Die Bedrohung für Netzwerkkomponenten hinter dem Application Gateway ist dadurch deutlich gesenkt, jedoch ist das Application Gateway selbst allen Angriffen ausgesetzt. Im dritten und letzten Fall zieht der angegriffene einfach seine Spielfigur aus der Bedrohung. Vergleichen kann man diesen Schritt mit dem Deaktivieren gefährdeter Dienste oder Teilfunktionen, die eine Schwachstelle aufweisen. Auch das Beheben eines Sicherheitsproblems durch Einspielen von Sicherheits-Updates manövriert das angegriffene System zumindest vorübergehend – bis zur Entdeckung neuer Sicherheitslöcher – aus der Bedrohung.

In vielen Fällen kann sich ein angegriffener Schachspieler nur dann erfolgreich verteidigen, wenn er geeignete Maßnahmen gegen die vielen Bedrohungen ergreift. Beispielsweise kann der Spieler seine Figuren derart in Stellung bringen, dass sie mögliche angreifende Spielfiguren schlagen oder zumindest deren Angriffe blocken können. Auch beim Schutz von Netzwerken können Angriffe besonders dann erfolgreich abgewehrt werden, wenn zuvor verschiedenste Sicherheitsmechanismen installiert wurden. Hierzu zählen beispielsweise die erwähnten Virenscanner, Firewalls oder Application Gateways.

Beim Schachspiel werden häufig Spielzüge getätigt, die einen Angriff auf den gegnerischen König vorbereiten sollen. Das kann beispielsweise so aussehen, dass der Angreifer mehrere Spielfiguren in Stellung bringt, um dann einen ganze Serie von Angriffen zu starten. Im Netzwerk lässt sich diese mit der Installation eines so genannten „*root-kits*“ durch einen Angreifer vergleichen. Ein *root-kit* beinhaltet in vielen Fällen eine Hintertür für den Angreifer, durch die er jederzeit einen Zugriff auf das System erhält. Oft enthalten *root-kits* auch weitere Werkzeuge, mit denen der Angreifer das kompromittierte System den eigenen Wünschen entsprechend konfigurieren kann. Hat der Angreifer seine Planungen und Vorbereitungen sorgfältig durchgeführt, so kann er im Anschluss seine Angriffsreihe durchführen und den Gegner permanent unter Zugzwang setzen.

Eine weitere besondere Angriffsform beim Schachspiel ist das Abzugsschach. Hier blockiert eine der eigenen Figuren einen Angriff auf den gegnerischen König. Zieht man nun diese Figur zu Seite, so entsteht als Nebeneffekt ein „Schach“ und damit eine echte Bedrohung für den gegnerischen König. Die gezogene Spielfigur kann gleichzeitig vorzugsweise dazu verwendet werden, andere Bedrohungen gegen andere gegnerische Spielfiguren auszusprechen oder diese gar zu schlagen. In der Ausgangsstellung aus Abbildung 10.1 stellt der schwarze Springer auf `f5` eine solche blockierende Figur dar. Schwarz kann durch Bewegen des Springers den dahinter auf `f8` stehenden Turm ins Spiel bringen, der direkt den weißen König bedroht. Schwarz kann also mit `Sh6+`[1] den weißen König indirekt ins Schach manövrieren und gleichzeitig die weiße Dame mit seinem Springer bedrohen (siehe Abbildung 10.3).

Diese Form des Angriffs ähnelt einem Ablenkungsmanöver. Auf den Schutz eines Netzwerkes bezogen kann dies eine parallele Initiierung zweier oder mehrerer Angriffe auf verschiedene Ziele entsprechen. In diesem Fall versucht der Angreifer, seine wahren Ziele zu verbergen, in dem er von seinem eigentlichen Vorhaben durch zeitgleiche Angriffe auf andere Teile des Netzwerkes ablenkt.

Eine spezielle Form des Abzugsschachs ist das Doppelschach, bei dem die gezogene Spielfigur die Blockade eines eigenen Angriffs aufhebt und gleichzeitig selbst einen Angriff auf den gegnerischen König ausübt. Nimmt man wieder Abbildung 10.1 als Ausgangsposition, so hätte Schwarz durch Bewegen seines Springers nicht nur den Turm als Angreifer ins Spiel bringen können, sondern mit den Zügen `Se3++` oder `S×g3++` hätte er auch gleichzeitig eine zweite Bedrohung für den weißen König aufbauen können (siehe Abbildung 10.4).

In diesem Fall bleibt dem angegriffenen Spieler kein anderer Ausweg als seinen König aus der Gefahrenzone zu ziehen – sofern er noch ein Feld ohne gegnerische Bedrohung findet. Auch beim Schutz von Netzwerken kann es zu einer solchen Situation kommen. Startet ein Angreifer mehrere Angriffe

[1] Ebensowenig wie der Zug `Lb5+` würde der Zug `Sh6+` als der beste Zug in dieser Situation gelten. Beim Schachspiel finden sich oft mehrere mögliche Züge, die zu einer ähnlichen Spielsituation führen würden. Im hier vorgestellten Beispiel würde Schwarz vermutlich `... S×g3+, Kg1 Dxg4` spielen.

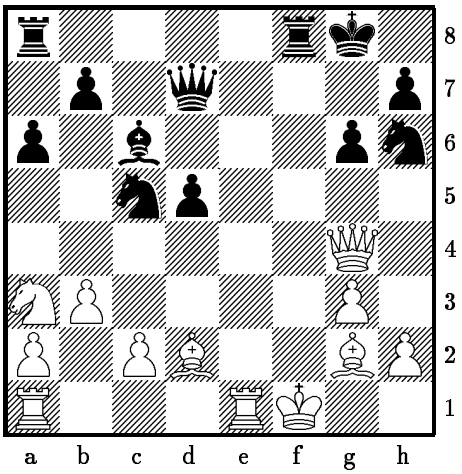

Abb. 10.3. Schwarz hat mit seinem Zug `Sh6+` ein Abzugsschach gespielt. Der Turm auf `f8` greift nun den gegnerischen König an, während der Springer die weiße Dame auf `g4` bedroht.

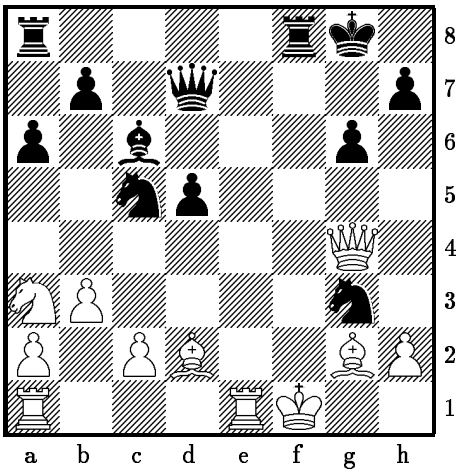

Abb. 10.4. Schwarz hat mit seinem Zug `S×g3++` ein Doppelschach gespielt. Sowohl der Turm auf `f8` als auch der gezogene Springer greifen nun den gegnerischen König an.

gleichzeitig auf dasselbe Ziel, so bleibt einem Netzwerkadministrator manchmal keine andere Wahl, als den bedrohten Dienst oder das bedrohte System vollständig aus dem Netzwerk zu entfernen.

Auch für die vielen Eröffnungsvarianten, zu denen Schach-Computer umfangreiche Bibliotheken besitzen, existiert eine gute Analogie: So, wie beim Schach für Eröffnungen gut ausgeklügelte Zugvarianten existieren, stehen einem Angreifer des Netzwerks bereits viele Programme und Werkzeuge für seine Zwecke zur Verfügung. Diese können bereits von „Script-Kiddies“ ohne große Vorkenntnisse bedient werden. Glücklicherweise beinhalten nicht nur die Eröffnungsvarianten beim Schach auch vorgegebene gute Gegenzüge, sondern es existieren auch eine Reihe von Werkzeugen, die mit wenig Aufwand den Schutz der Netzwerke deutlich steigern können.

Es gibt allerdings auch entscheidende Nachteile beim Vergleich mit dem Schachspiel. Während das Spiel irgendwann zwangsweise beendet ist, sei es durch ein Matt, ein Patt oder auch durch das Ablaufen der Zeit, so werden die Angriffe auf Netzwerke ununterbrochen fortgesetzt, ohne dass ein Ende in Sicht wäre. Außerdem bekämpft man beim Schutz eines Netzwerkes nicht nur einen einzigen Gegner, wie beim Schach, sondern es findet sich eine schier endlose Zahl an Angreifern im Internet, die unermüdlich und pausenlos, teilweise sogar automatisiert, ihre Angriffe auf verschiedene Netzwerke ausüben. Und es existiert noch ein entscheidender Unterschied: Datennetze sind um ein Vielfaches komplexer, als ein Schachspiel. Zwar ist das Königsspiel so komplex, dass es momentan keine Rechenmaschine auf der Welt gibt, der alle möglichen Züge einer Partie vorausberechnen könnte, jedoch beschränkt sich das Spiel auf „nur“ 64 Felder und „nur“ 16 Spielfiguren pro Spieler, die auch „nur“ eingeschränkte Züge durchführen können. Netzwerke mit Anschluss an das Internet haben nicht nur eine unbeschränkte Anzahl an „Spielern“, sondern es existieren auch unzählbar viele „Felder“, „Figuren“, „Zugmöglichkeiten“ und auch Angriffsziele. Aus diesem Grund können erfolgreiche Angriffe sogar vollständig im Verborgenen bleiben. Beim Schachspiel ist spätestens der Sieg durch Matt für beide Spieler offensichtlich. Im Netzwerk können aber Angreifer unbemerkt in ein System eindringen, ihre Aktionen durchführen und anschließend alle Spuren ihres Eindringens wieder beseitigen. In diesem Fall kennt nur der „Sieger“ seinen eigenen Sieg, nicht aber der „Verlierer“.

Eine ganz allgemeine Gemeinsamkeit zwischen Schachspiel und dem Schutz von Netzwerken vor Angriffen lässt sich schließlich dennoch festhalten: Man muss aktiv handeln, Passivität verliert. Wer sich durch seine(n) Gegenspieler ständig in die Defensive drängen lässt, der läuft auch Gefahr, die Kontrolle zu verlieren, und das ist sowohl beim Schach als auch als Netzwerkadministrator äußerst schlecht. Eine aggressive, aktive Handlungsweise kann zwar nichts garantieren, jedoch verringert sie die Wahrscheinlichkeit und die Gefahr, Opfer eines Angriffs und damit zum Verlierer zu werden.

10.1.2 Ernüchterndes Ergebnis

An dieser Stelle lässt sich resümieren, dass es den *perfekten* Schutz für Netzwerke ganz einfach nicht gibt. Das Problem des *perfekten* Schutzes liegt in der Tatsache, dass auf jeden gemachten Verteidigungszug der Angreifer einen neuen Angriffszug machen kann. Die Kunst besteht nun darin, dem Angreifer immer mindestens einen Schritt im Voraus zu sein. Um also ein Netzwerk und sein Management vor unbefugtem Zugang zu schützen, sollten alle Bereiche derart konfiguriert sein, dass sie die kleinstmögliche Angriffsfläche bieten. Das schließt vor allem das zeitnahe Einspielen von Sicherheits-Updates und das stetige Aktualisieren von Viren-Pattern der Virenscanner ein. Außerdem sollte man sich nicht auf eine einzelne Komponente zum Schutz des Netzwerkes verlassen. Besser ist eine Umsetzung des Zwiebel-Modells in der Netzwerksicherheit, bei der sich die Sicherheitsmechanismen in mehreren Schichten um den Kern des Netzwerkes hüllen – wie bei einer Zwiebel.

10.2 Abschwächung von Bedrohungen

Gegen alle Bedrohungen lassen sich prinzipiell Gegenmaßnahmen ergreifen, jedoch sollte an dieser Stelle noch einmal deutlich gemacht werden, dass es den *perfekten* Schutz nicht gibt. Alles, was man erreichen kann, ist eine möglichst effektive Abschwächung der Bedrohung und damit eine Minimierung der Verwundbarkeit und der Wahrscheinlichkeit eines erfolgreichen Angriffs.

An dieser Stelle werden nun die in Kapitel 9 vorgestellten Bedrohungen noch einmal aufgegriffen und auf das Netzwerkmanagement angewendet. Der dort verwendete Begriff ‚Informationen' bezieht sich demnach auf Daten des Netzwerkmanagements, also beispielsweise auf SNMP Pakete oder andere statische Informationen zu den überwachten Geräten, den Managementstationen und den verwendeten Kommunikationswegen. Zu allen Bedrohungen werden im Folgenden ein oder mehrere Beispiele gegeben, zu denen dann jeweils ein möglichst effektiver Schutz beschrieben werden soll.

10.2.1 Verlust von Informationen

Ein Informationsverlust kann bei der Datenhaltung oder beim Datentransport auftreten. Es sind deshalb unterschiedliche Vorkehrungen gegen die jeweiligen Bedrohungen zu treffen.

Der Sender hat eine zu übermittelnde Nachricht nicht gesendet

Wenn ein Sender eine zu übertragende Nachricht nicht übermittelt, so kann das verschiedene Hintergründe haben. Der Sender könnte beispielsweise den Auslöser für ein Sende-Ereignis nicht erkennen oder aber er ist derart konfiguriert, dass er trotz korrekter Identifizierung des auslösenden Ereignisses nicht die richtigen Schritte einleitet.

Beispiel 1

Die Bedrohung einer nicht gesendeten Nachricht aufgrund fehlerhafter Erkennung der auslösenden Ereignisse lässt sich gut auf ein Intrusion Detection System (IDS) anwenden. Ein IDS, welches darüber hinaus direkt auf erkannte Einbruchsversuche reagieren und entsprechende Reaktionen veranlassen kann, fällt in die Kategorie der Intrusion Detection/Response Systeme (IDRS). Wenn die zugrunde liegenden Pattern des IDRS nicht dem Muster der aktuell eingehenden Netzwerkpakete entsprechen, so kann das Einbruchserkennungssystem den Angriff nicht identifizieren. Folgerichtig wird auch keine entsprechende Meldung an das Netzwerkmanagement weitergegeben, das anderenfalls durch dynamische Umkonfiguration verschiedener Komponenten des Netzwerks den Angriff hätte abblocken können. In Abbildung 10.5 ist ein Beispiel für Patterns des IDS SNORT [201] angegeben, die Angriffe auf das SNMP Protokoll erkennen und entsprechenden Alarm schlagen.

Analog zu den IDS und den IDRS verhalten sich auch Virenscanner oder Spam-Filter. Diese benötigen ebenfalls aktuelle Pattern, um ihre Aufgabe möglichst erfolgreich meistern zu können. Daraus leitet sich auch direkt ein wirkungsvoller Mechanismus ab, um die hier beschriebene Bedrohung klein zu halten. Damit die auf Pattern basierenden Systeme so effektiv wie möglich arbeiten können, müssen sie ständig mit aktuellen Pattern versorgt werden. Ein automatisierter Update-Mechanismus kann zusätzlich den Schutz erhöhen, da so die Aktualisierungen zeitnah bei Erscheinen der Updates eingespielt werden können. Allerdings sollte nicht verschwiegen werden, dass ein automatischer Update-Mechanismus für Patterns auch Nachteile bergen kann, insbesondere dann, wenn die Quelle der Updates nicht unter dem eigenen Einflussbereich steht – was den Normalfall darstellt. Durch den automatischen Mechanismus können schließlich auch ungewollte, bösartige Daten in das Netzwerk eingebracht werden, was es eigentlich zu vermeiden gilt. Hier gilt es abzuwägen, welches der beiden Risiken das größere darstellt: die ständig neu auftauchenden Angriffe, die durch die aktuellen Pattern abgewehrt werden könnten, oder die Gefahr, sich über einen automatischen Update-Mechanismus einen bösartigen Code in das Netzwerk zu holen.

Beispiel 2

Die Bedrohung der nicht gesendeten Nachricht aufgrund falscher Konfiguration der einzuleitenden Schritte kann sehr einfach auf den SYSLOG und den SYSLOG-NG Mechanismus angewendet werden, wobei unterschiedliche Sender in Frage kommen können. Ein Beispiel für einen Sender wäre der Ursprung einer SYSLOG Nachricht in einem Anwendungsprogramm wie einem Mail-Server. Durch geschickte Manipulation der Konfigurationsdateien kann bei vielen dieser Programme die Menge an gesendeten Informationen herabgesetzt werden. Als Folge davon würden wichtige Informationen über diesen Dienst, der Teil des Netzwerkes ist, nicht mehr an den zugehörigen SYSLOG oder SYSLOG-NG

```
alert udp $EXTERNAL_NET any -> $HOME_NET 161 (
   msg:"SNMP missing community string attempt";
   content:"|04 00|";
   depth:15;
   offset:5;
   reference:bugtraq,2112;
   reference:cve,1999-0517;
   classtype:misc-attack;
   sid:1893;
   rev:4;
)

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (
   msg:"SNMP null community string attempt";
   content:"|04 01 00|";
   depth:15;
   offset:5;
   reference:bugtraq,2112;
   reference:bugtraq,8974;
   reference:cve,1999-0517;
   classtype:misc-attack;
   sid:1892;
   rev:6;
)

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (
   msg:"SNMP public access udp";
   content:"public";
   reference:bugtraq,2112;
   reference:bugtraq,4088;
   reference:bugtraq,4089;
   reference:cve,1999-0517;
   reference:cve,2002-0012;
   reference:cve,2002-0013;
   classtype:attempted-recon;
   sid:1411;
   rev:10;
)
```

Abb. 10.5. Beispiel für Pattern-Definitionen des IDS Systems SNORT. Die angegebenen Pattern zielen auf die Erkennung von Angriffen über das SNMP Protokoll.

Server gesendet. Dieser kann deshalb auch nicht die erforderlichen geeigneten Mittel initiieren, so dass sich ein Problem im Netzwerk ausbilden und eventuell auch weiter ausbreiten kann, ohne dass der Administrator davon Kenntnis erhält. Abbildung 10.6 veranschaulicht dieses Szenario am Beispiel der Konfiguration eines Secure Shell (SSH) Servers. Durch Einfügen von zwei zusätzlichen Parametern in die Konfigurationsdatei wird das Logging auf ein Minimum beschränkt, so dass fehlgeschlagene Anmeldeversuche nicht mehr protokolliert werden. So kann ein Angreifer verhindern, dass der Systemadministrator durch eine Häufung von SYSLOG Einträgen auf einen Einbruchsversuch beim SSH Server aufmerksam wird.

```
# Global sshd configuration
Protocol 2

# Authentication
PermitEmptyPasswords no
ForcedPasswdChange yes
LoginGraceTime 45
IdleTimeout 600

# Allowed users
PermitRootLogin no
AllowGroups research
AllowUsers peter paul

# No XWindows
X11Forwarding no
```

```
# Global sshd configuration
Protocol 2

# Authentication
PermitEmptyPasswords no
ForcedPasswdChange yes
LoginGraceTime 45
QuietMode yes
SilentDeny yes
IdleTimeout 600

# Allowed users
PermitRootLogin no
AllowGroups research
AllowUsers peter paul

# No XWindows
X11Forwarding no
```

Abb. 10.6. Konfigurationsdatei eines Secure Shell Servers mit Einstellungen zum Logging. Links: Ursprüngliche Konfiguration mit aktiviertem Logging (Ohne besondere Angabe ist das Logging aktiviert). Rechts: Nach dem Angriff finden sich zwei zusätzliche Einträge in der Konfigurationsdatei, welche den SSHD Server nur noch kritische Fehlermeldungen an den SYSLOG Server senden lassen.

Um die Bedrohung eines derartigen Angriffs möglichst klein zu halten, lassen sich mehrere Vorkehrungen treffen. Zuerst einmal ist es wichtig, die Konfigurationsdateien aller wichtigen Dienste mit möglichst restriktiven Dateiberechtigungen zu versehen. Nur Administratoren, denen die jeweiligen Dienste unterstehen, sollten auch einen Lesezugriff und einen Schreibzugriff auf die jeweiligen Konfigurationsdateien haben. Zusätzlich lässt sich mit Prüfsummen die Veränderung von Dateien leichter erkennen. Das Prinzip beruht auf einer Hashwert-Bildung der gesamten Datei. Dieser wird zunächst einmalig

ermittelt und separat gespeichert. Stimmen nun der aktuelle und der separat gespeicherte Hashwert nicht überein, so wurde die Datei seit der Hashwert-Bildung verändert. Verwendet man sichere Einweg-Hashfunktionen wie MD5, so lassen sich leicht alle veränderten Dateien im System aufspüren. Eine regelmäßige Überprüfung der Hashwerte aller wichtigen Konfigurationsdateien identifiziert somit unerwünschte Änderungen und verhindert die Manipulation der Sensoren.

Der Übertragungsweg einer Nachricht ist unterbrochen

Das korrekte Versenden einer Nachricht bedeutet nicht zwangsweise, dass die Nachricht auch den Empfänger erreicht. Dazu muss die Nachricht zunächst den Übertragungsweg erfolgreich passieren. Es finden sich viele Ursachen für den Verlust einer Nachricht auf ihrem Transportweg. Ein einfacher Grund könnte beispielsweise die physikalische Unterbrechung des Kommunikationsweges sein. Hierbei ist es irrelevant, ob die Übertragungsstrecke den Fehler aufweist oder eines der vermittelnden Geräte auf dem Weg zum Empfänger. Insbesondere beim verbindungslosen Protokoll UDP, bei dem keine Antworten für eingehende Pakete gesendet werden, besteht die Gefahr des unbemerkten Verlustes der Nachricht.

Beispiel 3

Fällt der Übertragungsweg für eine Nachricht vollständig aus, so erreichen auch andere Netzwerkpakete nicht mehr ihren Bestimmungsort. In diesem Fall wirkt sich das Problem im Normalfall an vielen Stellen gleichzeitig aus, so dass der Netzwerkadministrator von vielen verschiedenen Quellen über den Ausfall informiert wird[2]. Komplettausfälle genießen bei den Betreibern von Netzwerken höchste Priorität, so dass derartige Störungen meist nicht von langer Dauer sind. Viele Ausfälle gehen sogar beinahe unbemerkt an den Nutzern des Netzwerkes vorbei, da die Administration einem derart gravierenden Problem im Regelfall durch redundante Übertragungswege und Vermittlungskomponenten vorbeugt. Beim Ausfall eines Transportweges werden die Datenpakete deshalb automatisch über andere parallele Wege zum Ziel transportiert. Dies betrifft auch die Nachrichten an den Administrator, welche über das aufgetretene Problem informieren. Einen Komplettausfall kann ein Angreifer auf verschiedenen Wegen herbeiführen. Ein einfaches Beispiel besteht im Herunterfahren einer oder mehrerer Schnittstellen einer zentralen Vermittlungskomponente mittels entsprechender SNMP Befehle. Viele SNMP-fähige Geräte implementieren die Standard-Community Namen `public` für Lesezugriffe und `private` für Schreibzugriffe. Abbildung 10.7 zeigt, wie ein

[2] Manche Netzwerkadministratoren behaupten, die besten Ausfallsensoren seien die Benutzer, die bei Problemen unverzüglich zum Telefonhörer greifen ...

Angreifer in einem solchen Fall über wenige Befehle sämtliche Schnittstellen eines Netzwerkgerätes deaktivieren kann[3]

```
badboy$ snmpget -c public 172.17.2.1 interfaces.1
interfaces.ifNumber : INTEGER: 3
badboy$ snmpset -c private 172.17.2.1 ifAdminStatus.1 i 2
ifAdminStatus.1 : INTEGER: 2
badboy$ snmpset -c private 172.17.2.1 ifAdminStatus.2 i 2
ifAdminStatus.2 : INTEGER: 2
badboy$ snmpset -c private 172.17.2.1 ifAdminStatus.3 i 2
ifAdminStatus.3 : INTEGER: 2
```

Abb. 10.7. Herunterfahren der Schnittstellen einer beliebigen Netzwerkkomponente mittels SNMP. Der erste Befehl bringt die Anzahl der Schnittstellen des Gerätes in Erfahrung. Die nachfolgenden drei Befehle fahren die vorhandenen Schnittstellen herunter in den administrativen Zustand *down(2)*.

Gegen diese Bedrohung existiert ein wirksamer Schutz, der sich problemlos auf alle Geräte und Dienste im Netzwerk ausweiten lässt. Durch das Umkonfigurieren sämtlicher Standard-Passwörter in sichere Passwörter wird das Risiko eines derartigen Angriffs erheblich reduziert. Gerade beim SNMP Dienst wird hierauf oftmals verzichtet. Es darf aber nicht vergessen werden, dass der SNMP Dienst ein wichtiges und mächtiges Instrument zur Verwaltung von Netzwerken darstellt und dass die Community Namen aus diesem Grund besonders schützenswert sind.

Beispiel 4

Problematischer wird die Situation, wenn ein Übertragungsweg im eigenen Netzwerk selektiv unterbrochen ist. In einem solchen Fall wird nur die Übermittlung von ausgewählten Paketen unterdrückt. Rein äußerlich scheint der Transportweg intakt zu sein, da einige der Daten ihr Ziel ordnungsgemäß erreichen. Gelingt es einem Angreifer, ausgewählte Pakete des Netzwerkmanagements zu unterdrücken, so kann er bedeutend ungestörter und mit einem geringeren Risiko der Entdeckung seine Aktivitäten durchführen. Auch zur selektiven Unterbrechung eines Übertragungsweges bieten sich einem Angreifer wieder verschiedene Möglichkeiten. Insbesondere dann, wenn sich vermittelnde Komponenten der Vermittlungsschicht (Layer 3) des OSI Referenzmodells

[3] In der Praxis würde ein Angreifer sicherlich etwas intelligenter vorgehen können, indem er zunächst die IP Adressen der Schnittstellen in Erfahrung bringen würde. So kann er vermeiden, diejenige Schnittstelle, über welche das Netzwerkgerät die SNMP Befehle erhält, zu früh herunterzufahren und sich damit den eigenen Weg zur Komponente abzuschneiden.

auf dem Übertragungsweg befinden, ist eine Selektion der Datenpakete vergleichsweise einfach. Abbildung 10.8 knüpft an das vorangegangene Beispiel an und veranschaulicht die selektive Unterbrechung eines Kommunikationsweges an einer Router Konfiguration. Im gezeigten Ausschnitt wird durch einfaches Hinzufügen einer Zugangskontrollliste das Gerät derart umkonfiguriert, dass die Log-Meldungen des SSH Servers blockiert werden und den vorgesehenen SYSLOG Server nicht mehr erreichen.

```
version 12.3
no service pad
service tcp-keepalives-in
service tcp-keepalives-out
!
hostname backbone-1
!
  ...
interface FastEthernet0/0
 description Remote Login Zone
 ip address 10.1.4.6 255.0.0.0
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 no cdp enable
!
  ...
```

```
version 12.3
no service pad
service tcp-keepalives-in
service tcp-keepalives-out
!
hostname backbone-1
!
  ...
interface FastEthernet0/0
 description Remote Login Zone
 ip address 10.1.4.6 255.0.0.0
 ip access-group 137 in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 no cdp enable
!
  ...
access-list 137
      deny udp any any eq 514
  ...
```

Abb. 10.8. Ausschnitt aus der Konfiguration eines Cisco Routers. Links: Ursprüngliche Konfiguration einer Schnittstelle ohne Zugangskontrolllisten (alle Pakete werden weitergeleitet). Rechts: Nach dem Angriff wurde die Zugangskontrollliste 137 hinzugefügt, welche die SYSLOG Meldung ausfiltert.

Um diese Art von Bedrohungen möglichst effektiv zu bekämpfen, sind die aktiven Vermittlungskomponenten des Netzwerkes vor unbefugtem Zugang gut abzusichern. Der im obigen Beispiel gezeigte Cisco Router ist durch den Angreifer zu seinen Zwecken umkonfiguriert worden. In diesem konkreten Fall war der Zugriff auf den Router nicht ausreichend abgesichert und es wurden nicht alle zur Verfügung stehenden Sicherheitsmechanismen ausgeschöpft. Ne-

ben der Wahl eines sicheren Passwortes für den Zugang zum Router bietet auch der Hersteller bietet eigene Mechanismen zur Erhöhung der Sicherheit an. Mit dem Konfigurationsbefehl

```
service password-encryption
```

wird die Darstellung von Passwörtern im Klartext verhindert. Auf diese Weise kann ein Angreifer, der aus irgendeinem Grund in den Besitz der laufenden Konfiguration des Gerätes gelangt ist, dennoch keine Passwörter auslesen. Ein zweiter, zusätzlicher Schutz muss in diesem Fall die Verwendung von verschlüsselten Passwörtern im Gerät selber sein. Dazu wird beispielsweise das *enable* Passwort, mit dem ein Benutzer in den administrativen Konfigurationsmodus des Cisco Routers wechseln kann, über eine sichere Variante konfiguriert:

```
enable secret akJW21g7=3xU#5%!K
```

Beispiel 5

Noch anders sieht der Sachverhalt aus, wenn die auf dem Weg liegenden Netzwerkkomponenten nicht unter der Kontrolle des Administrators stehen. In einem solchen Fall müssen diese Komponenten genauso betrachtet werden, als wenn sie unter der Kontrolle eines potentiellen Angreifers stünden. Damit Nachrichten auf einem unbekannten und unsicheren Weg zuverlässig übermittelt werden und nicht unbemerkt verloren gehen können, müssen diese Nachrichten ein verbindungsorientiertes Protokoll verwenden. Wie an verschiedenen Stellen bereits beschrieben, ist das aber gerade beim Netzwerkmanagement nicht immer der Fall. Sowohl SNMP als auch SYSLOG setzen auf das kompakte aber auch verbindungslose Protokoll UDP. Eine Umstellung auf das verbindungsorientierte Protokoll TCP ist zwar bei beiden Mechanismen theoretisch möglich, unglücklicherweise unterstützen aber nur ausgewählte Netzwerkkomponenten und Netzwerkdienste das zuverlässigere Protokoll. Einen möglichen Ausweg aus der speziellen Situation der unzuverlässigen Nachrichtenauslieferung können Proxy-Geräte bieten. Beide aufgeführten Netzwerkmanagement-Mechanismen unterstützen in ihrer neuesten Version den Transport von Nachrichten über mehrere Zwischenstationen, die dann jeweils als Proxy fungieren. Legt man die Nachrichtenwege nun so aus, dass über die unzuverlässigen Transportwege ausschließlich TCP Pakete versendet werden, so kann der Verlust von Informationen zumindest in Grenzen gehalten werden. Abbildung 10.9 veranschaulicht den Einsatz von Proxy-Geräten zur Überbrückung unzuverlässiger Übertragungswege.

Eine weitere Möglichkeit zur Erhöhung der Zuverlässigkeit des Übertragungsweges über unkontrollierte Netzwerke besteht in der Einrichtung eines verschlüsselten Tunnels. Auch wenn die Daten nicht besonders schutzbedürftig sein sollten, erreicht man mit einem verschlüsselten Tunnel dennoch die zweifelsfreie Erkennung von Paket- und Informationsverlusten. Die meisten

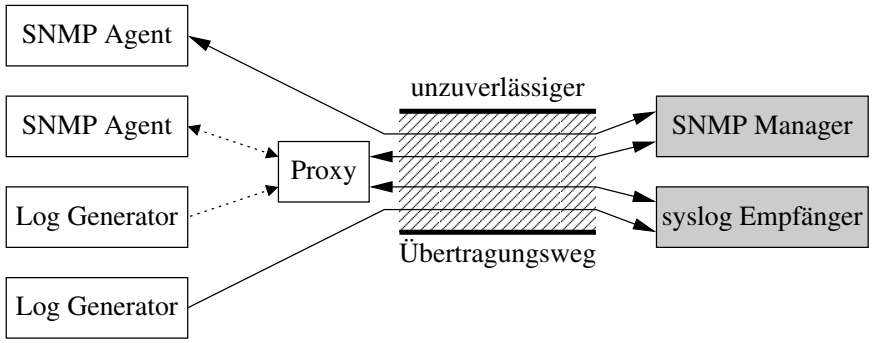

Abb. 10.9. Verwendung von Proxy-Geräten für SNMP und SYSLOG Nachrichten zur Überbrückung von unzuverlässigen Transportwegen. Einige der Netzwerkkomponenten und Netzwerkdienste können direkt TCP Pakete erzeugen (durchgezogene Linien) und über den unzuverlässigen Kanal senden. Alle anderen Geräte, die nur UDP Pakete erzeugen können (gestrichelte Linien), müssen den Proxy zum Nachrichtentransport verwenden.

Tunnelprotokolle erkennen einen Ausfall der Netzwerkverbindung und reagieren mit einem Abbau des Tunnels. Dies wiederum kann vom Netzwerkmanagement erkannt werden und automatisch notwendige Maßnahmen nach sich ziehen.

Der Empfänger ignoriert die Nachricht

Ähnlich dem SSH Server können auch der SYSLOG oder der SYSLOG-NG Server in ihrer Funktion als Nachrichtenempfänger ausfallen. Auch hier lassen sich wieder verschiedene mögliche Ursachen ausmachen. Beispielsweise könnte der Empfänger derart manipuliert sein, dass er zwar die Nachricht empfängt, aber anschließend keine Reaktion folgen lässt. Außerdem könnte der Empfänger deaktiviert sein, so dass er keine Nachrichten entgegennehmen kann.

Beispiel 6

Genau wie die meisten anderen Anwendungsprogramme besitzt auch ein Log-Server eine Konfigurationsdatei, die sein Verhalten steuert. Durch geschickte Manipulation der SYSLOG oder SYSLOG-NG Konfiguration kann der Log-Server dazu gebracht werden, ausgewählte Nachrichten zwar entgegenzunehmen, sie aber nicht weiter zu verwenden. Ein Beispiel mit ähnlichen Auswirkungen wie das in Abbildung 10.6 gezeigte Szenario ist in Abbildung 10.10 dargestellt. Durch geringfügige Abänderung der abgebildeten Konfigurationsdatei eines SYSLOG-NG Servers wurde dieser dazu veranlasst, die eingehenden Log-Meldungen des SSH Servers zu ignorieren. Die Änderung ist auf den ersten Blick kaum wahrnehmbar: In der Definitionszeile für den SYSLOG-NG Filter für den SSH Dienst wurde lediglich der vorhandene Programmname `sshd` durch

die Zeichen ssh ausgetauscht. Nach dieser kleinen Änderung werden nicht nur alle Nachrichten von anderen Programmen als dem SSH Server weggefiltert, sondern durch einen bewussten Schreibfehler werden zusätzlich auch die Meldungen des SSH Servers selber ausgeblendet[4].

```
# syslog-ng configuration
source {
  unix-stream("/dev/log");
  udp();
  internal();
};

filter ssh {
  not program("sshd");
};

destination sshdlog {
  file("/var/log/sshd.log"
       sync(0)
       log_fifo_size(10)
       create_dirs(yes)
       owner(root)
       group(system)
       perm(0600)
       dir_perm(0700));
};

log {
  source(src);
  filter(sshd);
  destination(verbose);
};
```

```
# syslog-ng configuration
source {
  unix-stream("/dev/log");
  udp();
  internal();
};

filter ssh {
  not program("ssh");
};

destination sshdlog {
  file("/var/log/sshd.log"
       sync(0)
       log_fifo_size(10)
       create_dirs(yes)
       owner(root)
       group(system)
       perm(0600)
       dir_perm(0700));
};

log {
  source(src);
  filter(sshd);
  destination(verbose);
};
```

Abb. 10.10. Konfigurationsdatei eines SYSLOG-NG Servers mit Einstellungen zur Behandlung von SSH Log-Nachrichten. Links: Ursprüngliche Konfiguration mit aktiviertem Logging. Rechts: Nach dem Angriff werden keine Logging Meldungen mehr in die angegebene Datei gespeichert, da die Nachrichten nicht mehr dem Filterkriterium ‚not program ("ssh");' genügen.

Zur Abschwächung dieser Bedrohung muss der Zugriff auf alle relevanten Systeme des Netzwerkmanagements eine maximale Restriktivität aufweisen. Zu

[4]Da es sich beim SSH Server um einen Unix Daemon handelt, lautet der übermittelte Programmname typischerweise ‚sshd' und nicht ‚ssh'.

diesem Zweck können verschiedene Schritte durchgeführt werden, die größtenteils auch auf andere kritische Netzwerkkomponenten übertragbar sind:

- Verwendung von sicheren Passwörtern für die Administrationszugänge
- Deaktivierung aller anderen Benutzerkonten, die auf einer Netzwerkmanagementstation nicht vorhanden sein sollten
- Erhöhung des Zugriffsschutzes auf die Konfigurationsdateien
- Verschlüsseln des Inhaltes der Konfigurationsdateien oder zumindest besonders kritischer Bereiche der Konfigurationen
- Gegebenenfalls Installation von Werkzeugen, die durch Prüfsummen eine unbemerkte Manipulation der Konfigurationsdateien aufspüren können.

Vor allem beim Schutz der Konfigurationsdateien sollte man unbedingt berücksichtigen, dass diese in vielen Fällen nicht nur an einer Stelle hinterlegt sind, sondern oftmals an mehreren unterschiedlichen Stellen zu finden sind. Betrachtet man das Beispiel eines Cisco Routers, so steht die Konfigurationsdatei selbstverständlich im Hauptspeicher in der laufenden Konfiguration. Sie steht aber eventuell auch noch in anderen Dateien des Speichers wie der Startkonfiguration, die bei der Neuinitialisierung des Gerätes ausgelesen und ausgeführt wird. In vielen Fällen werden die Router Konfigurationen außerdem extern auf einer Managementstation gepflegt und bei Bedarf auf die Systeme eingespielt. Die Ablage der Dateien kann dann sogar auf einem dritten System erfolgen, das vielleicht eine Datenbank mit einem Versionierungssystem enthält. Und schließlich befinden sich sämtliche Dateien der Managementstation und der Datenbank in einem oder mehreren Backup-Dateien auf verschiedenen Medien unterschiedlicher Server. Die aktuelle Router Konfiguration kann also an einer Vielzahl von Punkten im Netzwerk eingesehen und eventuell auch manipuliert werden. Gerade die hier erwähnten Cisco RouterindexCisco@Cisco!Router können sehr leicht unbemerkt manipuliert werden, indem lediglich ihre Startkonfiguration im Hauptspeicher verändert wird. In der laufenden Konfiguration lassen sich keine Änderungen feststellen, aber sobald das System neuinitialisiert wird, treten die Änderungen in Kraft. Aus diesem Grund ist es besonders wichtig, dass die Konfigurationsdateien an allen Stellen mit dem gleichen Maß an Sicherheit behandelt werden.

Beispiel 7

Eine Nachricht kann auf der Empfängerseite auch dann ignoriert werden, wenn der Empfänger schlicht und ergreifend deaktiviert ist. Sendet beispielsweise ein SNMP Agent fleißig SNMP Nachrichten über das verbindungslose Protokoll UDP an eine Netzwerkmanagementstation, so können sowohl Sender als auch Übertragungsweg vollständig intakt sein, aber bei ausgeschaltetem Trap Empfänger gehen die Meldungen dennoch verloren. Eine Möglichkeit zur Vermeidung dieses Problems ist die Selbstüberwachung der Netzwerkmanagement-Systeme. Was vielleicht auf den ersten Blick etwas seltsam anmutet, mag vor

allem in verteilten Managementsystemen durchaus sinnvoll sein. Ein Komplettausfall aller Managementkomponenten im ganzen Netzwerk wird mit sehr hoher Wahrscheinlichkeit eine derart gravierende Ursache haben, dass diese sich auch überdeutlich an anderen Stellen zeigen wird. Die Netzwerkadministration wird diesen Spezialfall wohl nur sehr schwer übersehen können.

Gespeicherte Daten gehen verloren

Unabhängig von der Übertragung von Informationen können auch gespeicherte Daten verloren gehen. Die möglichen Ursachen sind äußerst vielschichtig und reichen vom expliziten Löschen der Daten durch einen Angreifer bis hin zum Überlauf einer Datensenke.

Beispiel 8

Auch für die Bedrohungsart des Datenverlustes durch einen aktiven Angriff kann der SYSLOG Mechanismus ein gutes Beispiel liefern. Die im Netzwerk generierten Log-Meldungen werden entweder lokal gespeichert oder sie werden – eventuell sogar zusätzlich – über das Netzwerk zu einer Datensenke transportiert. Die gespeicherten Daten bilden eine Grundlage vieler verschiedener Managementaufgaben. Anhand der Log-Daten lassen sich beispielsweise Probleme im Netzwerk leichter identifizieren und lokalisieren. Aber auch Einbruchsversuche lassen sich in den Log-Daten erkennen und zurückverfolgen. Möchte ein Eindringling seine Spuren im System beseitigen oder seine Aktivitäten gänzlich verbergen, so muss er auch die in den Log-Daten vorhandenen Indizien seiner Aktionen auslöschen. Abbildung 10.11 zeigt einen gekürzten und vereinfachten Ausschnitt aus einer Log-Datei, in der neben einigen erfolglosen Versuchen auch die Spuren eines erfolgreichen Einbruchs zu erkennen sind. Ein intelligenter Angreifer würde die hervorgehobenen Einträge, welche seinen Angriff protokollieren, nach erfolgreicher Übernahme des Systems löschen, um so die Administratoren zu täuschen.

Zum Schutz vor einem solchen Angriff können – wie in einem der weiter oben beschriebenen Beispiele – prinzipiell mehrere Vorkehrungen getroffen werden. Wichtig ist vor allem, dass der Zugriff zum SYSLOG Server stark eingeschränkt ist und nur autorisierten Benutzern einen Zugang gewährt wird. Weiterhin müssen selbstverständlich alle Passwörter hinreichend sicher gewählt werden. Wichtig ist aber vor allem, dass der Zugriff auf die eigentlichen Log-Daten geschützt wird. Mechanismen wie die Hashwert-Bildung, die in anderen Situationen durchaus hilfreich sind, greifen bei der Überwachung von Log-Dateien nicht, da diese sich Einsatz-gemäß stetig ändern. Einzig eine hohe Rotation der Log-Dateien kann eine bedingte Einsetzbarkeit von Hashwerten bringen. Werden die Log-Daten häufig in neue Archive verschoben, so lassen sich zumindest die Archive überwachen. Die aktuelle Log-Datei ist für diese Methodik jedoch völlig ungeeignet. Daher ist vor allem auf eine möglichst restriktive Zugangsberechtigung der Log-Dateien zu achten. Besser noch ist

```
dhcpcd[614]: sending DHCP_REQUEST for 172.17.2.15 to 172.17.2.2
dhcpcd[614]: dhcpIPaddrLeaseTime=172800 in DHCP server response.
dhcpcd[614]: DHCP_ACK received from  (172.17.2.5)
cron[96]: (root) CMD ( rm -f /var/spool/cron/lastrun/cron.hourly)
-- MARK --
authpsa: IMAP connect from @ [80.181.71.227]
authpsa: checkmailpasswd: FAILED: admin - no such user
authpsa: IMAP connect from @ [80.181.71.227]
authpsa: checkmailpasswd: FAILED: test - no such user
authpsa: IMAP connect from @ [80.181.71.227]
authpsa: checkmailpasswd: FAILED: web - no such user
authpsa: IMAP connect from @ [80.181.71.227]
authpsa: checkmailpasswd: FAILED: www - no such user
proftpd[8282]: 24.9.156.54 ([24.9.156.54]) - FTP session opened.
proftpd[8282]: 24.9.156.54 ([24.9.156.54]) - FTP session closed.
-- MARK --
sshd[732]: Illegal user test from ::ffff:213.239.218.109
sshd[732]: Illegal user guest from ::ffff:213.239.218.109
sshd[732]: Illegal user admin from ::ffff:213.239.218.109
sshd[732]: Illegal user user from ::ffff:213.239.218.109
sshd[732]: error: PAM: Authentication failure
sshd[732]: error: PAM: Authentication failure
sshd[732]: error: PAM: Authentication failure
sshd[732]: error: PAM: Authentication failure
sshd[732]: error: PAM: Authentication failure
sshd[732]: error: PAM: Authentication failure
sshd[732]: Accepted keyboard-interactive/pam for www
```

Abb. 10.11. Ausschnitt aus einer Log-Datei eines Unix-Gerätes. Die Spuren eines erfolgreichen Einbruchs sind hervorgehoben (unterhalb des zweiten – *MARK* – Abschnitts). Zum Verschleiern und Vertuschen des Einbruchs muss der Angreifer noch diese Einträge aus den Log-Dateien löschen.

ein SYSLOG Server in einem besonders geschützten Netzwerk unterzubringen. Vorteilhaft kann sich auch ein Netzwerkmanagement nach dem Out-of-Band Management Verfahren auswirken (siehe Abschnitt 2.5). Die für den Betrieb und das Management des Netzwerkes wichtigen Komponenten sind in diesem Fall durch ein separates Netzwerk miteinander verbunden, bei dem höchste Ansprüche an die Sicherheit gestellt werden.

Beispiel 9

Neben dem aktiven Löschen können Daten auch indirekt durch Einflüsse ihrer Umgebung verloren gehen. Als Beispiel soll hier der Überlauf einer Datensenke dienen, was durchaus wieder am SYSLOG Mechanismus verdeutlicht werden kann. Voraussetzung ist, dass ein Angreifer in die Lage versetzt wird, entweder

direkt oder indirekt eine stark erhöhte Anzahl von Log-Meldungen zu generieren. Dies kann er beispielsweise durch einen eigenen Dienst erreichen, der zusätzliche Log-Meldungen an den SYSLOG oder SYSLOG-NG Server sendet. Er kann aber auch eine bestehende Quelle von Log-Meldungen derart angreifen, dass diese vermehrt gültige Log-Nachrichten versendet. Konkret könnte also der bereits mehrfach zitierte SSH Server mit Angriffen überschüttet werden, so dass dieser eine große Anzahl von Log-Meldungen erzeugt und versendet. Werden noch andere Dienste im Netzwerk auf die gleiche Weise angegriffen, so kann der Angreifer nicht nur versuchen, seine Spuren in der großen Anzahl von SYSLOG Meldungen zu verstecken, er kann auf diesem Weg eventuell auch den Überlauf des Speichers für die SYSLOG Meldungen provozieren. Durch die Verwendung eines Log-Rotationsmechanismus kann sich ein Netzwerkadministrator vor dem Überlauf des zugrunde liegenden Dateisystems schützen. Bei Erreichen einer bestimmten Größe werden die aktuellen Meldungen einfach in einer komprimierten Datei archiviert und in einem First-In-First-Out (FIFO) Verfahren abgelegt. Auf diese Weise ist die maximale Anzahl an Meldungsarchiven und der von ihnen belegte Speicherplatz begrenzt. Der klare Nachteil dieses Verfahrens liegt allerdings darin, dass beim Erzeugen einer neuen Archiv-Datei gleichzeitig die älteste Datei überschrieben wird und verloren geht. Kann ein Angreifer hinreichend viele SYSLOG Nachrichten in kürzester Zeit provozieren, so kann er ebenfalls die Spuren seines Einbrechens aus dem System entfernen – wenn auch mit einiger Zeitverzögerung.

Zum Schutz vor diesem Angriff können mehrere Ansätze parallel verfolgt werden. Zum einen sollte die zu speichernde Datenmenge möglichst minimiert werden. Einige SYSLOG und SYSLOG-NG Server sind beispielsweise in der Lage, eingehende Nachrichten zusammenzufassen. Wiederholen sich Meldungen mehrfach, so wird vom Server nur noch jeweils eine Meldung mit Angabe der Wiederholrate gespeichert. Gleichzeitig verringert natürlich die Anschaffung einer größeren Datensenke das Risiko eines Überlaufs. Allerdings sollte in solchen Fällen nicht vergessen werden, dass die gespeicherten Daten auch noch ausgewertet werden müssen. Man sollte also gleichzeitig entsprechende Werkzeuge zur Verfügung stellen, die mit den größeren Datenmengen umgehen können.

10.2.2 Bekanntwerden von Informationen

Genau genommen ist der Begriff Diebstahl bislang etwas unsauber und unscharf verwendet worden. Unter einem Diebstahl von Informationen versteht man in einigen Fällen die unberechtigte Inbesitznahme von Daten durch das Ausführen unbefugter Aktionen. Tatsächlich können Informationen auf diesem Wege auch veröffentlicht werden. Andererseits können Daten an Dritte gelangen, die keine oder zumindest keine unbefugten Aktionen durchgeführt haben. Dies gilt gleichermaßen für statisch gespeicherte Daten wie für übermittelte Informationen. Der Einfachheit halber soll an dieser Stelle weiterhin von Diebstahl gesprochen werden, wenn die Informationen an einen einge-

schränkten Teil der Öffentlichkeit gelangen. Bei einer Veröffentlichung von Informationen ist gleichzeitig die Gruppe der Informationsempfänger unbeschränkt und mit der Allgemeinheit gleichzusetzen.

Diebstahl von Informationen

Informationen aus dem Bereich Netzwerkmanagement liegen selten direkt im Mittelpunkt von Angriffen zu Informationsdiebstählen. Häufig bilden die anvisierten Daten nur ein Mittel und Werkzeug, um ganz andere und wertvollere Informationen erlangen zu können.

Beispiel 10

Informationen über die genaue Struktur eines Netzwerkes können äußerst hilfreich bei der Planung von Angriffen auf andere wertvolle Datenquellen sein. Hierzu zählen nicht nur allgemein die Kommunikationswege im Netzwerk, sondern vor allem auch die Transportwege des Netzwerkmanagements. Die Kenntnis der internen Netzwerkstruktur und der genauen Mechanismen des Netzwerkmanagements eröffnen einem Angreifer eventuell mögliche Schwachstellen und damit auch Angriffspunkte für einen ersten Einstieg. Ist der Anfang erst einmal gemacht, so kann der Angreifer dann Schritt für Schritt weitere Informationen sammeln und die anschließenden Aktivitäten planen und durchführen. Eine hilfreiche Informationsquelle zur Ermittlung der genauen Netzwerkstruktur sind beispielsweise die Address Resolution Protocol (ARP) [158] Speicher aller Netzwerkgeräte. Der ARP Speicher besteht aus einer Tabelle mit Zuweisungen von Media Access Control (MAC) Adressen und Netzwerkadressen. In den meisten Fällen handelt es sich dabei um IP Adressen. Durch das Auslesen des ARP Speichers kann ein Angreifer sukzessive Informationen über alle im Netzwerk bekannten Geräte erhalten. Unabhängig von ihrer IP Adresse kennt jedes Gerät im Netzwerk seine direkten Kommunikationsnachbarn. Damit Netzwerkkomponenten sich mittels höherer Protokolle verständigen können, müssen sie die MAC Adresse des auf dem Weg dorthin gelegenen Netzwerkknotens kennen, zu dem sie ihre Pakete senden. Dies ist entweder das Gerät selber – sofern es sich im selben Netzwerk befindet – oder das nächstgelegene Vermittlungsgerät. Da ARP im Hintergrund diese MAC Adressen und deren Zuordnung zu IP Adressen speichert, gibt es keine Garantie, dass jede benachbarte Netzwerkkomponente auch mit einem Eintrag in der ARP Tabelle vertreten ist. Einem Angreifer reicht oftmals bereits eine unvollständige Liste aus, um eventuelle neue Geräte zu identifizieren. Diese können anschließend angegriffen werden, um deren ARP Tabelle auszulesen. Mit hinreichend Geduld reicht manchmal sogar der ARP Speicher eines einzigen Gerätes aus, denn dieser ändert sich dynamisch mit den jeweiligen Kommunikationsbeziehungen zu anderen Geräten. So wird die ARP Tabelle einer Netzwerkkomponente ständig aktualisiert und auf dem Laufenden gehalten.

Abbildung 10.12 zeigt einen Ausschnitt der ARP Tabelle eines Netzwerkgerätes, aus dem sich eindeutig die Zuordnungen zwischen einigen der im Netzwerk vergebenen IP Adressen und den zugehörigen MAC Adressen ablesen lässt.

```
Address          HWtype  HWaddress           Flags  Iface
172.17.2.129     ether   08:00:20:B5:A1:BD   C      eth0
172.17.2.206             (incomplete)               eth0
172.17.2.95      ether   00:08:02:28:83:7D   C      eth0
172.17.2.57      ether   00:08:02:B0:69:70   C      eth0
172.17.2.199     ether   00:00:5A:9B:95:C3   C      eth0
172.17.2.145     ether   00:E0:C5:59:AF:68   C      eth0
172.17.2.1       ether   00:20:9C:10:37:75   C      eth0
172.17.2.221             (incomplete)               eth0
172.17.2.41      ether   00:08:02:B0:38:99   C      eth0
Entries: 9 Skipped: 0 Found: 9
```

Abb. 10.12. Ausschnitt aus der ARP Tabelle einer Netzwerkkomponente unter Zuhilfenahme des Unix Befehls ARP.

Gegen den hier beschriebenen Angriff kann man sich effektiv nur sehr schwer wehren, so lange man einen physischen Zugriff auf das Netzwerk nicht ausschließen kann. Hat ein Angreifer die Möglichkeit, ein Netzwerkgerät seiner Wahl im Zielnetzwerk zu etablieren, so hat er damit auch die Möglichkeit, vergleichsweise einfach an die im Netzwerk verbreiteten Pakete zu gelangen. Einen effektiven Schutz können deshalb nur Mechanismen auf niedriger Ebene bieten. Die Einbindung des 802.1X Standards [211] des Institute of Electrical and Electronics Engineering (IEEE) [89] ermöglicht beispielsweise eine Authentifizierung und Autorisierung von Ethernet Schnittstellen noch bevor höhere Protokolle wie IP oder TCP zum Tragen kommen (siehe Kapitel 7).

Beispiel 11

Auch die im Abschnitt 9.3.2 angeführten dynamischen Routingprotokolle RIP und OSPF unterliegen der Gefahr eines Informationsdiebstahls. Kann der Angreifer ein Gerät seiner Wahl im selben Autonomen System der Router platzieren, so kann er im Falle von RIP einen Nachbar-Router um Informationen über das gesamte Netzwerk befragen. Im Falle von OSPF kann er die Meldungen aller anderen Router sammeln, um so detaillierte Informationen über die Netzwerktopologie zu erhalten.

Zur Lösung dieses Problems kann das Routing im Autonomen System auf ein aktuelles sicheres Protokoll wie beispielsweise OSPF 2 [134] umgestellt werden, welches auch eine Verschlüsselung und Authentifizierung unterstützt. Ein Abhören der Routinginformationen ist somit für einen Angreifer deutlich schwieriger.

Veröffentlichung von Informationen

Wie bereits dargelegt soll unter der Veröffentlichung von Informationen die unberechtigte oder unbeabsichtigte Weitergabe von Daten an die gesamte Öffentlichkeit verstanden werden. Die Unterscheidung zwischen unberechtigt und unbeabsichtigt differenziert dabei noch einmal die beiden verschiedenen Szenarien, in denen entweder die Daten gestohlen und anschließend veröffentlicht werden oder aber die Daten auf Grund eines Fehlers an die Öffentlichkeit dringen. Das erste Beispiel unterscheidet sich nur unwesentlich vom Informationsdiebstahl und soll deshalb hier nicht mehr erläutert werden. Es sei an dieser Stelle lediglich noch einmal auf den vorhergehenden Abschnitt verwiesen. Das zweite Szenario findet sich jedoch bedeutend häufiger. Hier besteht zunächst einmal das Problem, die vor der Veröffentlichung zu schützenden Daten zu identifizieren. Wenn es sich um einen Informationsaustausch einer Entwicklungsabteilung handelt, bei der sensible Daten transportiert werden, dann ist dies noch ein schnell einleuchtendes Beispiel. Es finden sich jedoch auch Informationen, bei denen man sich auf den ersten Blick nicht bewusst ist, dass ein Schutz vor Veröffentlichung durchaus sinnvoll sein kann.

Beispiel 12

Ein konkretes Beispiel für veröffentlichte Informationen, die häufig unbeachtet bleiben, sind die Begrüßungsmeldungen, welche die Dienste im Netzwerk jedem Anfragenden zusenden. In vielen Fällen beinhaltet die Begrüßungsmeldung verschiedene Informationen zum System, die deutliche Rückschlüsse auf die verwendete Software und deren genaue Version sowie das verwendete Betriebssystem und dessen genaue Version zulassen. Manchmal beinhalten die Begrüßungsmeldungen auch noch weitere Informationen zur Organisation, zu welcher das Gerät gehört. Ein intelligenter Angreifer kann die Informationen der Begrüßungsmeldung zu seinem Vorteil nutzen, da die genaue Spezifizierung von Hardware und Software-Versionen damit direkt auf dazu bekannte Schwachstellen verweist. Mit der Kenntnis der speziellen Versionen kann ein Angreifer gezielt diese Schwachstellen ausfindig machen und muss nicht erst mühsam Angriffspunkte identifizieren. In Abbildung 10.13 sind einige Beispiele für Begrüßungsmeldungen mehrerer Netzwerkdienste aufgelistet. Einen direkten Zugriff auf die Begrüßungsmeldungen erhält man leicht mit Hilfe des TELNET Programms. Dazu verbindet man sich einfach mit dem jeweiligen Server unter Verwendung der korrekten Portnummer des gewünschten Netzwerkdienstes. Anhand der gezeigten Beispiele lässt sich nicht nur der unterschiedliche Detaillierungsgrad der Begrüßungsmeldungen erkennen, sondern auch die Tatsache, dass die Begrüßungsmeldungen noch vor der Authentifizierung an den Anfragenden gesendet werden. Ist ein solcher Netzwerkdienst über das Internet öffentlich erreichbar, so werden die gezeigten Informationen vom Netzwerkbetreiber buchstäblich veröffentlicht.

Der Schutz vor dieser Bedrohung ist denkbar einfach: Bereits die Umkonfiguration der Begrüßungsmeldungen, so dass diese keine Detailinformationen

```
badboy$ telnet sshserver 22
Trying 172.27.60.155...
Connected to sshserver.
Escape character is '^]'.
SSH-1.99-OpenSSH_3.8p1
```

```
badboy$ telnet mta 25
Trying 172.27.50.2...
Connected to mta.
Escape character is '^]'.
220 mta.domain.local ESMTP Postfix
```

```
badboy$ telnet mstore 110
Trying 172.27.50.5...
Connected to mstore.
Escape character is '^]'.
+OK Microsoft Exchange 2000 POP3 server version 6.0.6249.0 ready.
```

```
badboy$ telnet www 80
Trying 172.27.70.3...
Connected to www.
Escape character is '^]'.
```

Abb. 10.13. Beispiele für gängige Begrüßungsmeldungen verschiedener Netzwerkdienste, die teilweise detaillierte Angaben zu Hardware und Software-Versionen enthalten. Zur Sichtbarmachung der Meldungen wird das TELNET Programm verwendet.

sondern nur noch die wesentlichen Angaben enthalten, ist oftmals bereits ausreichend. Zusätzlich lässt sich überlegen, ob die Begrüßungsmeldungen um eine entsprechende Aussage ergänzt werden, die den unbefugten Zugang zu den Komponenten und den jeweiligen Diensten explizit verbieten.

Beispiel 13

Ein weiteres Beispiel für die unbeabsichtigte Veröffentlichung von Informationen, bei dem auch Auswirkungen auf das Netzwerkmanagement zu erkennen sind, ist eng mit der Vergabe von Standard-Passwörtern verbunden. Gemeint sind hier öffentlich zugängliche, SNMP-fähige Systeme, die mit Standard-Community Namen konfiguriert sind. Jeder Anfragende erhält von diesen Netzwerkkomponenten unter Verwendung der weithin bekannten Community Namen bereitwillig Auskunft über eine Vielzahl von Detailinformationen

zum System, die dann leicht zur Ermittlung von weiteren Informationen oder potentiellen Schwachstellen dienen können. Insbesondere das SNMP-Objekt *system.sysDescr* soll nach der Vorgabe detaillierte Informationen zur verwendeten Hardware, zum eingesetzten Betriebssystem und zur installierten Software machen.

Auch der Schutz vor dieser Bedrohung ist denkbar simpel. Es muss lediglich penibel auf eine Konfiguration sicherer Community Namen bei allen Geräten geachtet werden. Bei manchen Geräten müssen die Standard-Community Namen sogar explizit auskonfiguriert werden, da sie ansonsten zusätzlich zu den eigenen vergebenen Community Namen weiterhin gültig sind. Konkretes Beispiel hierfür ist die Cisco PIX Firewall, die im Auslieferungszustand bereits den Community Namen `public` für Lesezugriffe vorinstalliert hat [42]. Es handelt sich hierbei keineswegs um eine Ausnahme; ein ernüchternd großer Anteil aller SNMP-fähigen Netzwerkkomponenten ist bereits mit ähnlichen Standard-Community Namen vorkonfiguriert. In solchen Fällen kann man sich vor der Bedrohung der Veröffentlichung von Informationen vergleichsweise leicht schützen. Entscheidend ist lediglich, dass für sämtliche Komponenten neue, sichere Community Namen vergeben werden. Für Geräte, die nicht zwingend vom Netzwerkmanagement über das SNMP Protokoll administriert werden sollen, hilft auch das rigorose Deaktivieren des SNMP Agenten auf der Netzwerkkomponente. Ist allerdings ein Zugriff über das SNMP Protokoll notwendig, so müssen die Standard-Community Namen durch sichere Community Namen ersetzt werden. Dies gilt sowohl für die Community Namen, die einen Lesezugriff auf die implementierten MIBs erlauben, als auch für eventuelle Standard-Community Namen, die zusätzlich das Schreiben der MIBs gestatten. Cisco weist in seinem Handbuch für die PIX Firewall explizit darauf hin, dass aus Sicherheitsgründen unbedingt ein neuer Community Name vergeben werden sollte.

10.2.3 Verfälschung von Informationen

Eine Verfälschung von Informationen findet in vielen Fällen nur durch Einwirkung eines Angreifers statt. Die Zielsetzungen für eine Informationsverfälschung können vielschichtig sein, so dass auch das Netzwerkmanagement ins Fadenkreuz der Angreifer rücken kann.

Beispiel 14

Ein gutes Beispiel für die Verfälschung von gespeicherten Daten mit finanziellem Hintergrund stellen die Informationen dar, welche ein Service Provider zur Ermittlung von Abrechnungsdaten seiner Kunden generiert, sammelt und speichert. Eine konkrete Möglichkeit zur Erfassung der Accounting Daten findet sich im IP-Accounting Mechanismus, der vor allem auf Routern des Herstellers Cisco zu finden ist. Für jede einzelne Schnittstelle eines Cisco Routers lässt sich mittels des Befehls `ip accounting` der Abrechnungsmechanismus aktivieren. Das Grundprinzip vom IP-Accounting Mechanismus

basiert auf einem Ringspeicher, der sämtliche gerouteten Pakete aufnimmt. In regelmäßigen Abständen wird nun der Speicher ausgelesen und die bis dahin gesammelten Daten können extern weiterverarbeitet werden. Gleichzeitig wird der Zeiger auf den Anfang des Ringspeichers hinter die Position des letzten ausgelesenen Eintrags verschoben, so dass bei erneutem Auslesen keine Einträge doppelt verarbeitet werden. Das komplette Auslesen der IP-Accounting Daten im Ringspeicher des Routers lässt sich bequem über SNMP abwickeln. Abbildung 10.14 zeigt den Ausschnitt eines in der Programmiersprache Perl [154] implementierten Beispiel-Programms, welches diese Aufgabe erledigt. Entscheidend sind die beiden OIDs *1.3.6.1.4.1.9.2.4.9.1.4* und *1.3.6.1.4.1.9.2.4.11.0*, über welche jeweils der Inhalt und der Zeiger auf den Anfang des Ringspeichers adressiert werden können.

Das für jeden Ringspeicher typische Problem liegt im zugrunde liegenden FIFO-Prinzip. Bei einem Überlauf des Speichers können zwar weiterhin Abrechnungsdaten gesammelt und gespeichert werden, jedoch gehen gleichzeitig die ältesten Daten des Ringspeichers verloren. Damit dies nicht passieren kann, muss das Abrechnungsmanagement hinreichend häufig die Daten vom Router abrufen und den Ringspeicher somit wieder freigeben. Gelingt es einem Angreifer nun, beispielsweise durch einen DoS Angriff die Abfragen der Managementstation lange genug zu unterdrücken oder zu blockieren, so würde nach einiger Zeit der Ringspeicher des Routers überlaufen und es würden Abrechnungsdaten verloren gehen. Das gleiche würde passieren, wenn der Angreifer SNMP-Schreibzugriff auf den Zeiger für den Beginn des Ringpuffers hätte. Durch Verschieben des Zeigers nach „hinten“ könnten einige der bereits gemessenen Pakete ohne vorherige Abrechnung aus dem Ringspeicher gelöscht werden. Zwar könnte man beide Varianten auch als einen Informationsverlust auffassen, die Gemeinsamkeit liegt jedoch in der Verfälschung der Abrechnungsdaten auf der Managementstation.

Zur Entschärfung dieser speziellen Bedrohung können verschiedene Lösungsansätze verfolgt werden:

- Gegebenenfalls sollen nicht alle, sondern nur ausgewählte Pakete abgerechnet werden. Anstatt nun sämtliche Pakete zu messen, zu speichern und nach der Übertragung auf der Managementstation auszuwerten, können die irrelevanten Pakete bereits auf dem Router ausgefiltert und bezüglich des IP-Accounting verworfen werden. Der Ringspeicher würde so entsprechend langsamer gefüllt und muss daher auch seltener ausgelesen werden. Die Filterung kann auf dem Cisco Router durch das Anwenden einer Zugangskontrollliste auf den Abrechnungsmechanismus mittels des Befehls `ip accounting-list` erfolgen.
- Der Umsetzung des Abrechnungsmanagements in einem Out-of-Band Management kann hier ebenfalls vorteilhaft sein. Durch die Trennung der wichtigen Abrechnungsdaten von den Nutzdaten lassen sich die äußeren Einflüsse auf den IP-Accounting Mechanismus besser kontrollieren. Wich-

```
my $cisco = "1.3.6.1.4.1.9";      # Base OID for vendor Cisco
my $ck    = "$cisco.2.4.6.0";     # Lost bytes from ring buffer
my $ac    = "$cisco.2.4.9.1.4";   # New accounting data
my $cp    = "$cisco.2.4.11.0";    # Start of ring buffer

# Establish the session
my ($sess, $s_error) = Net::SNMP->session(
   -hostname=>$router_ip, -community=>$router_community );
die "Error while establishing session! ($s_error)\n"
   unless defined($sess);

# Check the router for lost bytes
if (my $result = $sess->get_request(-varbindlist => [$ck])) {
   if ($result->{$ck} > 0) {
      log_to_file(timestamp().": $result->{$ck}\n";
      debug_msg("Error: Lost accounting bytes on router.");
   }
} else {
  log("Error while checking for lost bytes! ($sess->error())");
}

# Get and set the checkpoint of the accounting ring buffer.
if ($result = $sess->get_request(-varbindlist => [$cp])) {
   if ($result = $sess->set_request(
      -varbindlist => [$cp, INTEGER32, $result->{$cp}])) {
      debug("Checkpoint resetted.");
   } else {
      log("Error while setting check point! ($sess->error())");
   }
}

# Retrieve new accounting data from router and store them
my @args = (-varbindlist => [$ac]);
while (defined($result = $sess->get_next_request(@args))) {
   $base = (keys(%{$sess->var_bind_list}))[0];
   last unless (Net::SNMP::oid_base_match($ac, $base));
   if ($base =~ m/(\d+\.\d+\.\d+\.\d+)\.(\d+\.\d+\.\d+\.\d+)$/) {
      store_traffic($1, $2, $result->{$base}); # (src, dst, bytes)
   } else {
      log("Error retrieving accounting data! ($result->{$base})");
   }
   @args = (-varbindlist => [$base]);
}
```

Abb. 10.14. Ausschnitt aus einem beispielhaften Perl-Programm zur Abwicklung des IP-Accounting auf einem Cisco Rechner.

tige Voraussetzung bleibt selbstverständlich ein hinreichend gut abgesichertes und abgeschirmtes separates Managementnetzwerk.

- Wenn der Ringspeicher für das IP-Accounting zu schnell überzulaufen droht, so kann durch den Befehl `ip accounting-threshold` die Größe des Speichers erhöht werden. Im Auslieferungszustand ist dieser Wert auf 512 Einträge beschränkt, der sich jedoch in Abhängigkeit von der verwendeten Hardware deutlich erhöhen lässt. Zu bedenken bleibt jedoch, dass eine extreme Erhöhung der Ringspeichergröße auch mit einem erheblichen Leistungsverlust des Gerätes verbunden ist.

Beispiel 15

Informationen können nicht nur im statischen Zustand manipuliert werden, sondern auch bei der Übertragung im Netzwerk. Berühmtes Beispiel dafür ist der „Man-in-the-Middle" Angriff, bei dem Nachrichten im Netzwerk über einen Knotenpunkt umgeleitet werden, welcher unter der Kontrolle eines Angreifers steht. Die dort eintreffenden Pakete können vom Angreifer beliebig verändert werden, bevor sie dann zu ihrem ursprünglichen Bestimmungsort geleitet werden. Dies können als ein einfaches Beispiel die Zustandsinformationen des Netzwerkmanagements sein, die über das Hypertext Transfer Protocol (HTTP) versendet werden.

Interessanterweise werden die „Man-in-the-Middle" Angriffe selbst oftmals ebenfalls durch eine andere Informationsverfälschung vorbereitet. Schließlich muss zunächst das Netzwerk dazu gebracht werden, die für den Angreifer interessanten Pakete zu dem von ihm kontrollierten Knotenpunkt umzuleiten. Durch ein ARP-Spoofing – also dem Fälschen der MAC Adresse in der Sicherungsschicht des OSI Referenzmodells – kann der Netzwerkknoten beispielsweise eine andere gültige Identität vortäuschen, so dass die anderen Komponenten des Netzwerks bereitwillig die Pakete zum Angreifer senden. Dieser Angriff ist deshalb so leicht durchzuführen, weil das ARP Protokoll über keinerlei Sicherheitsmechanismen verfügt. Jedes Gerät kann also beliebige ARP Pakete versenden, ohne dass die Identität des Gerätes und die Gültigkeit des Paketes überprüft werden könnte. Ein unter dem Einfluss eines Angreifers stehendes Gerät kann also jederzeit die Identität eines im Netzwerk vorhandenen autorisierten Systems übernehmen. Als Folge davon werden beispielsweise alle weiterzuleitenden Pakete zu diesem System gesendet. Alternativ kann der Angreifer auch die Routingprotokolle des Netzwerks angreifen und das unter seiner Kontrolle stehende Gerät dort als autorisiert propagieren. Die Auswirkungen des Angriffs sind aber mit einem „Man-in-the-Middle" Angriff vergleichbar.

Für einen effektiven Schutz vor einem „Man-in-the-Middle" Angriff müssen zwei Mechanismen parallel eingesetzt werden. Zum einen muss die Kommunikation verschlüsselt und mit einer Prüfsumme versehen werden, so dass Veränderungen der Nachricht sofort entdeckt werden können. Zum anderen müssen die an der Kommunikation beteiligten Gesprächspartner sich gegen-

seitig authentifizieren, so dass die Identität der Systeme für das jeweils andere System überprüfbar wird. Nur so kann einem „Man-in-the-Middle“ Angriff erfolgreich vorgebeugt werden. Betrachtet man das eingeführte einfache Beispiel der HTTP-Übertragung von Zustandsinformationen des Netzwerkmanagements, so kann die Gefahr des „Man-in-the-Middle“ Angriffs und der Informationsverfälschung durch Verwendung des Secure Hypertext Transfer Protocols (HTTPS) [172] deutlich verringert werden. Das HTTPS Protokoll verfügt sowohl über eine Verschlüsselung wie auch über eine Zertifikat-basierte Authentifizierung. Ein Administrator kann mittels HTTPS die Informationen der Netzwerkmanagementstation sicher und mit nur sehr geringer Gefahr der Verfälschung auf dem Transportweg abrufen und auf seinem Client darstellen lassen.

10.2.4 Vortäuschung von Informationen

Informationen können sowohl statisch als auch in Form von übermittelten Nachrichten vorgetäuscht werden. Diese Bedrohung verschont auch das Netzwerkmanagement nicht.

Beispiel 16

Ein einfaches Beispiel für die Vortäuschung von Informationen in Form von Netzwerkpaketen findet sich im SNMP Rahmenwerk. Das gesamte Netzwerkmanagement basiert in diesem Fall auf einfachen Konfigurationsanweisungen der Managementstationen und auf SNMP Nachrichten der verschiedenen Agenten. Aufgrund des im Normalfall zugrunde liegenden verbindungslosen Protokolls UDP können diese Pakete vergleichsweise einfach erzeugt und vorgetäuscht werden. Ist ein Angreifer in der Lage, SNMP Pakete vorzutäuschen, so hat er damit oftmals bereits die volle Kontrolle über das komplette Netzwerk erhalten. Je nach Ausprägung sind beinahe alle Managementaufgaben des Netzwerkes über SNMP durchführbar. Kann ein Angreifer eigene SNMP Pakete erzeugen, so kann er auch sämtliche Managementaufgaben des Netzwerkadministrators übernehmen. Besitzt ein Angreifer nicht die Möglichkeit, SNMP Schreiboperationen durchzuführen, sondern ist er lediglich in der Lage, SNMP Nachrichten zu versenden, so kann er zwar nicht das Netzwerk kontrollieren, aber er kann es in entscheidendem Maße stören. Es sollte jedem klar sein, dass in der Übernahme des Netzwerkmanagements durch einen Angreifer der schlimmstmögliche Angriff symbolisiert ist, da augenblicklich alle vorgestellten Bedrohungsarten Realität werden. Der entstehende Schaden eines derartigen Angriffs kann immens sein und sogar zum Ruin des Netzwerkbetreibers führen. Um so wichtiger ist es, sich vor diesem Angriff bestmöglich zu schützen.

Im Falle des Netzwerkmanagements via SNMP kann das Risiko um Größenordnungen gesenkt werden, wenn zur Konfiguration und Überwachung flächendeckend das sichere SNMPv3 zum Einsatz kommt. Die über das Netzwerk

verschickten Pakete können bei SNMPv3 sowohl verschlüsselt als auch authentifiziert werden. Von einem Angreifer erzeugte und als gültig vorgetäuschte SNMP Pakete können im SNMPv3 Rahmenwerk leicht als Fälschungen identifiziert und verworfen werden. Dies betrifft sowohl die SNMP Anfragen von Managementstationen als auch die SNMP Nachrichten der verschiedenen Agenten. Wichtig und ungeachtet des sicheren Protokolls sind die vergebenen Passwörter zur Authentifizierung entsprechend dem Schadenspotential besonders sicher zu verwahren. Mit ihrer Geheimhaltung steht und fällt schließlich das gesamte Sicherheitskonzept des Netzwerkmanagements.

Beispiel 17

Statische Daten lassen sich analog zum Informationsverlust nicht nur löschen, sondern auch erzeugen. Die Zielsetzung der Informationsvortäuschung kann dabei eine gänzlich andere sein. Im weitesten Sinne lassen sich „*root-kits*“ und Hintertüren auch als Informationsvortäuschung auffassen. Gemeint ist das Einrichten von zusätzlichen Programmen und Diensten auf einem System durch einen Angreifer. In der einfachsten Variante täuscht der Angreifer lediglich ein zusätzliches gültiges Benutzerkonto auf dem System vor, mit dessen Hilfe er sich jederzeit ordnungsgemäß mit dem System verbinden kann. Weiterführende *root-kits* enthalten komplette Werkzeuggruppen, über welche die verschiedensten Aufgaben auf dem System ausgeübt werden können. Kontakt kann der Angreifer zum Gerät beispielsweise auch über einen im *root-kit* enthaltenen zusätzlichen Dienst aufnehmen, der als gültiger Systemdienst vorgetäuscht wird.

Durch die Übernahme eines beliebigen Systems im Netzwerk kann ein Angreifer in vielen Fällen neue Angriffe durchführen, die ihm weitere Möglichkeiten eröffnen. Erreicht der Angreifer schließlich die Managementstationen, so ist der gleiche, schlimmste anzunehmende Angriffsfall erreicht: Der Angreifer hat die Kontrolle über das gesamte Netzwerk erhalten. An dieser Stelle soll noch einmal ausdrücklich betont werden, welcher immense Schaden durch den Verlust der Kontrolle über die Managementstationen entstehen kann. Alle in Kapitel 9 vorgestellten Bedrohungen sind in diesem Fall zur akuten Gefahr geworden.

Als Schutz vor dieser folgenschweren Bedrohung kann erneut eine Auslagerung des Netzwerkmanagements in ein separates Netzwerk nach dem Out-of-Band Verfahren eingerichtet werden. Auf diese Weise kann man sich als Netzwerkadministrator einen sicheren Bereich im Netzwerk installieren, der von den Angriffen im restlichen Netzwerk unberührt bleibt und zuverlässigen Zugang zum Managementsystem und die verwalteten Objekte im Netzwerk bietet. Einschränkend sollte jedoch noch einmal auf den Anfang dieses Kapitels verwiesen werden: Den *perfekten* Schutz gibt es nicht! Kann mit einem DoS Angriff ein über das Netzwerk verwaltetes System überlastet werden, so kann dadurch auch die Kommunikation zum Netzwerkmanagement nachhaltig beeinträchtigt oder sogar unterbrochen werden.

10.3 Honeypots und Honeynets

Nicht nur beim Schachspiel gilt die Weisheit: Angriff ist die beste Verteidigung. Auch beim Schutz von Netzwerken und dem Netzwerkmanagement können proaktive Maßnahmen durchgeführt werden, um das Risiko eines erfolgreichen Angriffs zu minimieren. Ein Beispiel dafür ist das Einrichten eines „Honeypots" [169]. Wie man einem Bären mit Honig eine Falle stellen kann, so kann man auch Angreifern mit einem scheinbar leicht zu kompromittierenden Rechner einen Hinterhalt legen. Federführend in dieser Art der Offensivverteidigung ist das Honeynet-Projekt [215], dessen Mission das Erforschen der Werkzeuge, Taktiken und Motive bei Computer- und Netzwerkangriffen sowie das Teilen des gewonnenen Wissens mit anderen ist. Das Grundprinzip von Honeypots und Honeynets basiert darauf, dass man einen Rechner oder ein ganzes Netzwerk an das Internet anschließt, auf dem keinerlei Produktivdaten liegen oder produktive Dienste laufen. Da aus diesem Grund rein theoretisch keine Kommunikation zu und vom Honeypot zu sehen sein dürfte, kann jede Verbindung mit diesem Gerät oder Netzwerk demnach als unerwünscht betrachtet werden. Gleichzeitig werden verschiedene Analyse- und Protokollierungsmechanismen installiert, die eine detailgetreue Verfolgung aller Aktionen auf dem Honeynet erlauben. Hierzu zählen vor allem Paket-Sniffer, welche die Netzwerkpakete auf niedriger Ebene protokollieren, oder auch ein Tastatur-Logger, der sämtliche Tastatureingaben erfasst.

Nach den Mitgliedern des Honeynet-Projekt und allen voran dem Sicherheitsexperten Lance Spitzner definiert sich ein Honeypot wie folgt:

> *Ein Honeypot ist ein Informationssystem, dessen Wert in der unbefugten oder rechtswidrigen Benutzung dieser Ressource liegt.*[5]

10.3.1 Installation eines Honeypots

Da ein Honeypot per definitionem keine Produktivdaten enthält und auch keine produktiven Aufgaben zu erfüllen hat, muss er lediglich den gegen ihn gerichteten Angriffen gewachsen sein. Aus diesem Grund bedarf ein Honeypot auch nicht zwingend einer aktuellen Hardware. Selbst aus veralteten Systemen können Honeypots errichtet werden, da der Missbrauch von Systemen in den meisten Fällen nur einen geringen Prozentanteil ihrer Ressourcen verbraucht – oder zumindest sollte dies so sein. Die genaue Beschaffenheit des Honeypots ist beinahe irrelevant. Der Schwerpunkt liegt in der Tatsache, dass es sich beim Honeypot um ein System handelt, das Schwachstellen besitzt oder zumindest dessen Applikationen Schwachstellen aufweisen. Daher ist auch die Wahl des Betriebssystems und der zu installierenden Dienste von nur geringer Bedeutung. Es sollte nur darauf geachtet werden, dass sich wenigstens eine Schwachstelle im Gesamtsystem befindet, über welche ein Angreifer weiter in

[5] „A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource."

das System vordringen kann. Möchte man gezielt einen konkreten anderen Rechner mit dem Honeypot schützen, so ist die Verwendung einer ähnlichen oder sogar der gleichen Architektur für den Honeypot von Vorteil. Alle erfolgreichen, gegen den Honeypot gerichteten Angriffe würde gleichermaßen das zu schützende System kompromittieren. Die Installation von Applikationen auf dem Honeypot schließt auch Sicherheitsmechanismen und Verschlüsselungsalgorithmen wie SSH oder HTTPS mit ein. Schließlich stellt der Honeypot immer einen Endpunkt der Verbindungen dar. Eine Verschlüsselung würde nur die Daten auf dem Weg zum Honeypot absichern, dort aber müssen sie spätestens entschlüsselt werden, so dass sie auch im Klartext vorliegen.

10.3.2 Überwachung der Aktivitäten

Ist ein Honeypot oder gar ein Honeynet erst einmal installiert, so rücken die Mechanismen zur Überwachung der Systeme in den Vordergrund. Der erste Schritt besteht in der Überwachung sämtlichen Datenverkehrs zu und vom Honeypot. Nach der Definition kann schließlich jede Kommunikation mit dem System als unautorisiert betrachtet werden. Dabei sollte man sich explizit nicht nur auf Verbindungen von außen beschränken, um vielleicht die eigenen Schritte der Administration des Systems nicht mitzuprotokollieren. Schließlich kann ein Honeypot auch indirekt oder sogar direkt von einem anderen System im eigenen Netzwerk aus angegriffen werden. Der Standort und Einstiegspunkt eines Angreifers ist daher auch als grundsätzlich unbekannt anzusetzen. Es sollte allerdings auch ein Protokoll über sämtliche offiziellen Administrationsschritte angelegt werden, damit sich diese im Nachhinein eindeutig als solche identifizieren lassen.

Die Überwachung des Netzwerkverkehrs in einem ersten Schritt entfaltet ihre volle Wirkung insbesondere bei Datenverkehr auf Basis der niederen Netzwerkprotokolle wie beispielsweise ICMP. Je mehr höhere Protokolle zum Einsatz kommen, desto größer ist auch die Gefahr der Verwendung eines verschlüsselten Kommunikationsweges. Einem Angreifer sind die vorhandenen Sicherheitsmechanismen oftmals sehr willkommen, da er durch deren Einsatz auch seine eigenen Spuren verschleiern und verbergen kann. Von einer Firewall können in solchen Fällen nur noch wenige Informationen protokolliert werden. Dies sind im Wesentlichen die unverschlüsselten Anteile der Netzwerkpakete wie die IP Adresse und Portnummer des Absenders oder die angesprochene Portnummer des Honeypots. Manchmal kann auch noch der Typ des Paketes etwas näher ermittelt werden, der Inhalt verschließt sich jedoch in den meisten Fällen einer Firewall. Erst im Zusammenspiel mit anderen Informationen wie Schlüsseln oder Passwörtern kann im Nachhinein ein Paket eventuell noch entschlüsselt werden.

Der zweite Schritt der Überwachung sollte daher bereits auf dem Honeypot selbst stattfinden. Ein geeignetes Mittel wären die systemeigenen Protokollierungsmechanismen wie der SYSLOG Mechanismus oder bei Microsoft Systemen die Ereignisprotokollierung. Die meisten Applikationen lassen sich

so konfigurieren, dass sie Alarme, Warnungen und andere Meldungen an den zentralen Protokollierungsmechanismus senden. Auf einem Honeypot kann die Protokollierung außerdem derart konfiguriert werden, dass möglichst viele Informationen gespeichert werden. Zur ungestörten Auswertung der Nachrichten sollte man sämtliche Meldungen an ein anderes System weiterleiten, zu dem der Angreifer vom Honeypot aus keinen Zugriff erhalten kann.

Weitere Schritte zur Überwachung können die Menge und den Detaillierungsgrad der ermittelten Informationen noch weiter erhöhen. Ein Beispiel wäre die Installation eines Key-Loggers oder aber auch die Ersetzung ausgewählter Systembefehle durch präparierte Versionen, die eine detailliertere Analyse der eingehenden Daten erlauben. Gleichzeitig erhöht sich aber auch die Gefahr der Entdeckung des Honeypots. Schließlich sind auch die Angreifer mit den Mechanismen eines Honeypots vertraut. Wer also zu viele Überwachungsfunktionen installiert, zu wenige „Produktivdaten" vorhält und den Honeypot nicht wie ein „normales" System mit Schwachstellen aussehen lässt, steigert somit die Gefahr der Enttarnung des Systems.

Als letzter Punkt ist die Überwachung des Honeypots auch deshalb notwendig, weil selbst trotz intensiver Betreuung ein Angreifer das System dennoch kompromittieren kann. Ist dies einmal der Fall, so kann der Angreifer den Honeypot dazu verwenden, um anderen Unbeteiligten Schaden zuzufügen, der dann auf die Betreiber des Honeypots zurückfällt. Um dies unter allen Umständen vermeiden zu können, ist daher auch eine intensive Überwachung und Auswertung der gesammelten Informationen zwingend notwendig.

10.3.3 Auswertung der Informationen

Ein Honeypot dient vor allem dem Zweck, die Techniken, Tools und Methoden der Angreifer zu studieren, um sich ihnen gegenüber einen Vorteil verschaffen zu können. Daher ist es zwingend notwendig, dass die Aktivitäten auf dem Honeypot permanent überwacht und auch ausgewertet werden. So lässt sich nicht nur ein Missbrauch des Systems zeitnah aufdecken, sondern auch die Mittel, welche die Angreifer einsetzen, können schneller ausgewertet werden, um so einen Schutz dagegen entwickeln zu können. Sind erst einmal neue Methoden entdeckt worden, so liegt es im Interesse des gesamten Internets, Informationen über diese Bedrohungen und Gefahren zu erhalten. Dabei sollte man sich im Klaren darüber sein, dass man immer eine extreme Gratwanderung unternehmen muss, um so vielen anderen wie möglich helfen zu können oder auch Hilfe von ihnen zu erhalten, ohne gleichzeitig die neuen Angriffsmethoden an andere Angreifer auszuliefern. In letzter Konsequenz können Honeynets also eine wertvolle Hilfe im Kampf gegen die vielen Bedrohungen im Internet sein. Der Betrieb eines Honeypots ist jedoch sehr arbeitsintensiv und zeitaufwändig. An dieser Stelle kann also nur an alle appelliert werden, bei dieser Arbeit mitzuhelfen.

11

Management Lösungen

Auf dem kommerziellen Markt und bei den nicht-kommerziellen Einrichtungen findet sich eine scheinbar endlose Anzahl verschiedenster Werkzeuge zum Management von Systemen und Netzwerken. Dieses Buch soll nicht als Einkaufsführer dienen und wird daher keine Empfehlungen aussprechen. Schließlich unterscheiden sich die verschiedenen Produkte – ob kommerziell oder nicht – teilweise sehr stark in ihrem Funktionsumfang. Daher ist es auch unmöglich, an dieser Stelle eine generelle Empfehlung für ein einzelnes Produkt zu geben. Für jeden konkreten Einsatzzweck haben jeweils andere Werkzeuge besondere Vorzüge und Nachteile. Außerdem existiert nicht für jeden Einsatzzweck eine einzige perfekte Lösung. Letztendlich kann die Management-Software auch aus Individuallösungen bestehen, die speziell auf das zu verwaltende System oder Netzwerk zugeschnitten wurden. Spätestens hier enden die Möglichkeiten in diesem Buch. An dieser Stelle kann daher nur versucht werden, einen eingeschränkten Ausschnitt der zur Verfügung stehenden Software-Produkte zu geben. Wer sich umfassender informieren möchte, der wird im Internet mit Hilfe einer Suchmaschine leicht die zum Zeitpunkt der Recherche aktuellen Produkte auffinden können. Zusätzlich finden sich für die verschiedenen Produkte zahlreiche Bewertungen von Nutzern der jeweiligen Software. Nicht zu vergessen sind auch die Informationsquellen für die verschiedenen nichtkommerziellen Produkte, die in vielen Fällen gleichzeitig quelloffen sind. Wer über die nötigen Ressourcen verfügt, kann bei der Planung und Implementierung seiner Managementsysteme auch auf diese Werkzeuge zurückgreifen. Man sollte dabei jedoch nicht vergessen, dass die Entwicklung von Individuallösungen nahezu immer mit einem sehr großen personellen und zeitlichen Aufwand verbunden ist. Bei Bedarf kann in diesem Fall auch auf Hilfe von außen gesetzt werden. Je nach den eigenen Möglichkeiten kann bei der Planung und Implementierung des eigenen Managementsystems auch auf externe Entwickler und Berater zurückgegriffen werden.

Wichtig bei aller Anstrengung ist, dass man das eigentliche Ziel nicht aus den Augen verliert. Auf der einen Seite versprechen kommerzielle Lösungen häufig eine schnelle Umsetzung der Anforderungen. Im Gegenzug dazu muss

das zur Bedienung der Software vorgesehene Personal zuvor ausreichend geschult werden. Außerdem können Lücken in der Abdeckung der gewünschten Funktionalitäten häufig nicht oder nur stark zeitverzögert und in Verbindung mit zusätzlichen Kosten geschlossen werden. Auf der anderen Seite bieten Individuallösungen eine gute Möglichkeit zur perfekten Abdeckung der gewünschten Funktionalitäten. Nachteilig dabei ist der dafür hohe Zeitaufwand und der Bedarf an gut geschultem Personal. Übernehmen die Entwickler anschließend auch die Bedienung und Betreuung der Management-Software, so fällt dafür allerdings auch der ansonsten notwendige Schulungsbedarf deutlich geringer aus.

11.1 SNMP Werkzeuge

Bei den SNMP Werkzeugen findet sich die größte Vielfalt an Produkten aus dem Bereich der Management-Software. Neben den vielen kommerziellen Werkzeugen finden sich vor allem auch nicht-kommerzielle und OpenSource Werkzeuge. Ursache hierfür ist die Tatsache, dass es sich bei SNMP um einen offenen Standard handelt, der jedem zur Verfügung steht. Damit ist auch die Grundvoraussetzung für eine aktive OpenSource Gemeinde gegeben. Erst durch diese Basis entsteht die Möglichkeit zur Entwicklung von Individuallösungen. Zu guter Letzt bieten viele Systemhersteller eigene Produkte zur Verwaltung ihrer Geräte an. Diese sind im Normalfall optimal auf die eigenen Geräte abgestimmt, erlauben aber auch die Überwachung und Konfiguration von Systemen anderer Hersteller.

11.1.1 Kommerzielle Werkzeuge

Bei der großen Vielfalt und Unübersichtlichkeit auf dem Sektor der kommerziellen SNMP Management-Software Produkte ist es unmöglich, einen vollständigen Überblick geben zu können. An dieser Stelle soll daher stellvertretend eine Auswahl an Produkten vorgestellt werden, ohne dabei eine Wertung vorzunehmen. Eine deutlich umfangreichere Liste findet sich im Internet – beispielsweise auf den Seiten von Pierrick Simier [200].

Luteus SARL: LoriotPro V3

Mit dem Werkzeug LoriotPro V3 stellt Luteus SARL[1] eine Management-Software zur Verfügung, mit der neben der obligatorischen Überwachung von Netzwerken auch eine Konfiguration der Netzwerkkomponenten möglich ist. Dazu wird auf die grundlegenden Protokolle SNMP, HTTP und ICMP zurückgegriffen. Die hervorstechenden Produktmerkmale können wie folgt summiert werden:

[1] `www.loriotpro.com`

Netzwerküberwachung. Mit dem Tool LoriotPro V3 können mit Hilfe des SNMP Protokolls nicht nur Netzwerkkomponenten wie Router und Switches überwacht werden, sondern auch die Server und Arbeitsstationen sowie die Dienste im Netzwerk. Zur Erweiterung der Überwachungsfunktionen können einzelne Systeme auch via ICMP auf Erreichbarkeit überprüft und ausgewählte Dienste auf ihren TCP Ports überwacht werden.

Netzwerkkonfiguration. Das SNMP Protokoll erlaubt auch eine Administration von ausgewählten OIDs, die von LoriotPro V3 ebenfalls unterstützt wird. Zusätzlich ermöglicht die Implementierung eigener Scripts die automatische Ausführung von komplexeren SNMP Anfragen, die bei Bedarf auch selbst spezifiziert werden können.

Flexible Visualisierung. Die Visualisierungskomponente des Werkzeugs von Luteus ermöglicht eine flexible Präsentation der gesammelten Informationen, die nach mehreren Kriterien sortiert werden können. Beispielsweise existiert eine topologische Übersicht, in welcher die Komponenten auf einer Weltkarte angeordnet nach ihren Lokationen sortiert angezeigt werden. Dieselbe Darstellung kann auch hierarchisch in einer baumartigen Struktur angezeigt werden. Zusätzlich lassen sich auf die verschiedenen Darstellungen benutzerdefinierte Filter anwenden, die entweder nur einen zuvor spezifizierten Ausschnitt von wichtigen Komponenten anzeigen oder aber sich auf die Darstellung von Informationen einer spezifizierbaren Kritikalität beschränken. Router besitzen in LoriotPro V3 eine besondere Bedeutung und können separat und unabhängig von allen anderen Informationsquellen visualisiert werden.

Automatisches Auffinden von Geräten. Mit Hilfe der beiden Protokolle SNMP und ICMP können die im Netzwerk vorhandenen Systeme auf Wunsch automatisch ermittelt und in die Datenbank aufgenommen werden.

Alarmbehandlung. Die in der werkzeugeigenen Datenbank abgelegten Alarme, die im Netzwerk produziert werden, können nach definierbaren Kriterien sortiert dargestellt werden, um beispielsweise Alarme besonders hoher Kritikalität leicht von anderen Alarmen trennen zu können.

Report-Generierung. Über die eingebaute Web-Schnittstelle können jederzeit Berichte in einem vorformatiertem Design erstellt werden, die Auskunft über ausgewählte Statistiken und andere Informationen zu den SNMP Daten liefern.

Fernüberwachung. Das Werkzeug LoriotPro V3 besteht zwar aus einem proprietären Microsoft Client, der jedoch auch eine eigene Web-Server Komponente aufweist. Über diese kann ein Administrator auch von jeder beliebigen Web-fähigen Station die Informationen des Tools abfragen.

Erweiterbarkeit. Durch den modularen Aufbau und die mitgelieferten Plug-Ins mit zugehörigem Quelltext ist es einem Administrator leicht möglich, eige-

ne Plug-Ins in der Programmiersprache C zu programmieren. Auf diese Weise ist das Werkzeug nahezu beliebig erweiterbar.

AdventNet, Inc.: ManageEngine OpManager 5

Das Werkzeug ManageEngine OpManager 5 des Unternehmens AdventNet Inc.[2] gehört wie viele andere Produkte ebenfalls zu den Überwachungs-Tools. Unterstützt wird sowohl das SNMP Protokoll zur Überwachung von Netzwerkkomponenten als auch eine Überwachung von Diensten auf deren jeweiligen TCP Ports. Die Darstellung der Ergebnisse erfolgt über die integrierte Web-Server Komponente. Die wichtigsten Merkmale des Produktes lassen sich wie folgt zusammenfassen:

Plattformunabhängigkeit. Zwar lässt sich ManageEngine OpManager 5 nicht auf jeder Plattform installieren, jedoch werden mit Microsoft und Linux zwei der wichtigsten Betriebssysteme unterstützt. Diese Tatsache macht das Tool von AdventNet zu einer sehr flexibel einsetzbaren Software.

Flexible Überwachung. Durch die Verwendung des offenen SNMP Standards und der ergänzenden Diensteüberwachung direkt über das TCP Protokoll ist eine sehr flexible Überwachung verschiedenster Komponenten im Netzwerk möglich. Unterschieden wird nach mehreren Gerätegruppen, für die jeweils andere Funktionalitäten zur Verfügung stehen. Die erste Gruppe besteht aus Routern, für die vor allem Statistiken zu den Netzwerkschnittstellen angezeigt werden können. Die Server in der zweiten Gruppe werden sowohl über das SNMP Protokoll überwacht, das beispielsweise Informationen zur Auslastung von Prozessor, Hauptspeicher oder Festplatten liefert, als auch über das TCP Protokoll, über welches die wichtigsten Dienste direkt auf ihren Ports angesprochen und überwacht werden. Eine dritte Gruppe besteht aus grundlegenderen Netzwerkkomponenten wie Switches oder Drucker. Letztere können beispielsweise auch auf Papierstaus oder das Zuneigegehen der Toner-Kartusche überwacht werden. ManageEngine OpManager 5 erlaubt eine automatische Identifizierung dieser Komponenten im Netzwerk, die auf dem SNMP und dem ICMP Protokoll aufbaut. Die vierte und letzte Gruppe besteht aus ausgewählten Applikationen, die etwas detaillierter überwacht werden, als die in der zweiten Gruppe erwähnten TCP Dienste. Diese vier ausgewählten Applikationen bestehen aus den Microsoft Produkten Exchange und SQL-Server sowie aus der Oracle Datenbank und dem Notes-Server von Lotus. Für jede dieser speziellen Applikationen werden zwischen 10 und 30 verschiedene Parameter via SNMP überwacht.

Flexible Benachrichtigung. Die über SNMP Nachrichten erhaltenen Alarme können automatisch verschiedene Ereignisse auslösen, zu denen neben der Ausführung beliebiger Programme auch die Benachrichtigung der Administratoren via E-Mail oder Short Message System (SMS) zählt.

[2] www.adventnet.com

Alarmverwaltung. Die gesammelten SNMP Nachrichten werden in der integrierten Datenbank abgelegt und können dort bearbeitet werden. Das System ist so ausgelegt, dass Administratoren die jeweiligen Alarme bestätigen müssen und auch mit Kommentaren versehen dürfen. Erfolgt die Bestätigung nicht innerhalb einer einstellbaren Zeitspanne, so wird automatisch eine Benachrichtigung der Administratoren durchgeführt.

Flexible Berichterstellung. In ManageEngine OpManager 5 können sowohl regelmäßig als auch nach Bedarf verschiedene Berichte über alle oder ausgewählte Statistiken generiert werden. Diese Berichte können außerdem in verschiedensten Formaten erzeugt werden. Neben einer Erstellung von Seiten im Hypertext Markup Language (HTML) Format und der Erzeugung des grundlegenden Exportformats über Listen aus Comma Separated Values (CSV) können die Berichte auch im Portable Document Format (PDF) erstellt werden.

Erweiterbarkeit. Viele der verwendeten Mechanismen lassen sich leicht den eigenen Vorstellungen entsprechend erweitern. Die Liste der über TCP Ports zu überwachenden Dienste lässt sich daher ebenso ergänzen wie die über SNMP überwachten Objekte.

FineConnection: Monitor one

Wie der Name bereits vermuten lässt, liegt die Hauptaufgabe des vom Unternehmen FineConnection[3] entwickelten Werkzeugs Monitor one in der Überwachung von Netzwerken. Es wird dabei mit Ausnahme des ICMP Protokolls zur Erkennung von Hosts lediglich das SNMP Protokoll in den Versionen SNMPv1 und SNMPv2 verwendet. Die Hauptmerkmale des Tools fassen sich wie folgt zusammen:

Flexible Visualisierung. Mit Monitor one lassen sich eigene Topologiekarten erstellen, um dem Anwender die Überwachung des Netzwerkes zu erleichtern. Dazu werden bereits eine Reihe von Icons mitgeliefert, die auf benutzerdefinierbaren Hintergrundbildern platziert werden können. Eine halbautomatische Erkennung der Komponenten im Netzwerk hilft gleichzeitig bei der Erstellung der Übersichtskarten.

Erreichbarkeitsüberprüfung. Die Erreichbarkeit von verschiedenen Systemen im Netzwerk kann entweder via ICMP Protokoll oder bei Problemen mit Firewalls alternativ auch via SNMP Statusabfragen erfolgen. Gleichzeitig kann das Zeitintervall vom Administrator konfiguriert werden, mit welchem die einzelnen Komponenten überprüft werden sollen.

Flexible Überwachung und Visualisierung von SNMP Werten. Jede einzelne OID lässt sich in Monitor one mit einem eigenen Sensor überwachen, für den auch gleichzeitig die Art der Visualisierung definiert wird.

[3] `www.fineconnection.com`

Es existieren neun verschiedene Typen dieser Sensoren, die auch „Shooter“ genannt werden. Mit dem „Table Shooter“ können ganze SNMP Tabellen ausgelesen und dargestellt werden. Der „Graph Shooter“ erlaubt das Anzeigen von nummerischen Werten in Echtzeitgraphen. Mit einem „Threshold Shooter“ können nummerische Werte ausgelesen und das Überschreiten oder Unterschreiten eines einstellbaren Schwellwertes überwacht und erkannt werden. Mit dem „History Shooter“ können Werte für eine Langzeitüberwachung gespeichert werden. MONITOR ONE bietet hier auch eine Schnittstelle zum Open-Source Round-Robin Datenbankwerkzeug RRDTOOL [145] an (siehe auch Seite 349). Mit einem „Set Shooter“ können ausgewählte OIDs nicht nur gelesen, sondern auch geschrieben werden. Der „Meter Shooter“ stellt einen nummerischen Wert in Form eines analogen Messinstruments dar. Die beiden „SnipMon Shooter“ erlauben die Anzeige von Graphen oder Meter-Anzeigen als kleine Symbole auf der Topologiekarte unterhalb des Icons für das Gerät. Der „Pie Shooter“ visualisiert gleich mehrere Werte in Form eines Tortendiagramms.

Universeller Nachrichtenempfänger. Neben den SNMP Nachrichten zur Erzeugung von Alarmen können auch SYSLOG Meldungen von MONITOR ONE entgegengenommen und gespeichert werden. Die vom integrierten SYSLOG Server erhaltenen Meldungen können ebenfalls analysiert und zur Erzeugung von Alarmen herangezogen werden.

TFTP Server. Zum externen Konfigurationsmanagement bauen viele Netzwerkkomponenten auf TFTP Server. Zu diesen können die Geräte ihre Konfigurationen exportieren und von diesen können sie auch Konfigurationen importieren. In MONITOR ONE ist ein TFTP Server integriert, der zu diesen Zwecken verwendet werden kann.

Flexible Benachrichtigung. Beim Eintreten von Alarmen kann MONITOR ONE auf viele verschiedene Arten eine Benachrichtigung der Administratoren durchführen. Neben der Erzeugung von benutzerdefinierbaren akustischen oder optischen Signalen können die Administratoren auch via E-Mail, SMS oder Pager informiert werden. Zusätzlich ist die Ausführung eines beliebigen Programms möglich.

Fernüberwachung. Neben dem integrierten Client findet sich in FineConnections Werkzeug auch ein Web-Server, welcher einen operativen Betrieb auch über die Web-Schnittstelle ermöglicht. Dies erlaubt es einem Administrator, das Tool von jedem beliebigen Web-Client zu bedienen.

Crannog Software: NETWATCH 1.5.0

Das Unternehmen Crannog Software[4] bietet verschiedene Produkte aus dem Bereich der Netzwerkverwaltung und Systemadministration an. Für das Netzwerkmanagement findet sich das Produkt NETWATCH VERSION 1.5.0. Mit

[4] www.crannog-software.com

diesem Produkt ist eine Web-basierte Überwachung von Systemen und Diensten möglich. Zur Überwachung einzelner Systeme werden die beiden älteren SNMP Versionen SNMPv1 und SNMPv2 eingesetzt; die Überwachung von benutzerdefinierbaren Diensten erfolgt direkt über deren TCP Port. Zusätzlich ist eine simple Überprüfung der Erreichbarkeit eines Systems über das ICMP Protokoll möglich. Die wichtigsten Produkteigenschaften fassen sich wie folgt zusammen:

Überwachung von Netzwerkgeräten. Die Überwachung der Netzwerkgeräte erfolgt in erster Linie über eines der beiden älteren SNMP Protokolle SNMPv1 und SNMPv2, setzt aber auch das ICMP Protokoll ein. Bei der Überwachung von Diensten wird im Wesentlichen eine direkte Überprüfung der Erreichbarkeit über den jeweiligen TCP Port vorgenommen. Die möglichen gesammelten Informationen beinhalten entweder einen Status, einen Alarm oder in eine Auslastung ein. Die Überprüfung der Dienste via TCP Port und die Überprüfung der Erreichbarkeit von Systemen über das ICMP Protokoll liefern prinzipiell nur einen Statuszustand zurück. Häufig beschränkt sich die Überwachung auf die binären Statuszustände „ist erreichbar“ und „ist nicht erreichbar“. Die Überwachung der Auslastung von Netzwerkschnittstellen ist wesentlich detaillierter und liefert mehr als nur einen binären Statuszustand. Ein Alarm bezieht sich immer auf eine Änderung im Netzwerk. Dies kann eine Änderung eines Statuszustandes sein, beispielsweise der Verlust der Erreichbarkeit eines Systems oder eines Dienstes. Ein Alarm kann aber auch bei Überschreiten der zuvor definierten Antwortzeit eines Systems oder eines Dienstes erfolgen. Schließlich kann auch das Überschreiten der angegebenen maximalen Auslastung einer Netzwerkschnittstelle einen Alarm auslösen.

Graphische Darstellung des Netzwerks. Mit Hilfe der Visualisierungskomponente kann in NetWatch 1.5.0 eine graphische Repräsentation des Netzwerkes erstellt und hinterlegt werden. Die Erstellung der Netzwerkübersichtsbilder erfolgt vom Administrator, der für jede Komponente eins von mehreren vorgegebenen Icons auf ein zuvor frei wählbares Hintergrundbild platziert. Anschließend können die dargestellten Systeme durch Linien miteinander verbunden werden, um so ihre Netzwerkverbindungen untereinander zu repräsentieren.

Flexible Benachrichtigung. Bei Auftreten eines Alarms können mehrere verschiedene Aktionen ausgeführt werden, die vom Administrator konfiguriert werden können. Jeder Alarm kann entweder zu einer Benachrichtigung via E-Mail oder SMS führen; der Alarm kann aber auch über ein Alarmfenster angezeigt werden. Zusätzlich wird jeder Alarm in der werkzeugeigenen Datenbank hinterlegt, die über die Web-Schnittstelle erreichbar ist.

Nachrichtenempfänger. NetWatch 1.5.0 ist sowohl in der Lage, SNMP Nachrichten zu empfangen und auszuwerten, als auch SYSLOG Meldungen entgegenzunehmen. Zu diesem Zweck muss lediglich auf den SNMP-fähigen Geräten die Managementstation mit der NetWatch 1.5.0 Software als Trap

Empfänger konfiguriert werden und auf den SYSLOG-fähigen Geräten die Managementstation als Log-Server konfiguriert werden. Einige der eingehenden SNMP Nachrichten können zusätzlich als Quelle für die Erzeugung von Alarmen dienen.

Sicherheit. Die graphische Oberfläche des NETWATCH 1.5.0 Werkzeugs ist über einen Web-Server erreichbar. Verwendet wird dabei das unsichere HTTP Protokoll, das eine unverschlüsselte Übertragung der Daten vornimmt. Allerdings kann in dem Tool von Crannog Software der Zugriff auf die HTTP-Seiten rollenabhängig derart eingeschränkt werden, so dass nur nach korrekter Angabe des Passwortes ein Zugriff möglich ist.

Woodstone bvba.: SERVERS ALIVE

Mit dem Werkzeug SERVERS ALIVE vom Unternehmen Woodstone bvba.[5] können Netzwerke und deren Komponenten über SNMP überwacht werden sowie einzelne Dienste direkt über ihren TCP oder UDP Port. Bezüglich des SNMP Protokolls können die Versionen SNMPv1 und SNMPv2 uneingeschränkt sowie von SNMPv3 das Authentifizierungsmodul verwendet werden. Die wichtigsten Merkmale zu SERVERS ALIVE werden im Folgenden zusammengefasst:

Flexible Überwachung. Im Vordergrund des Tools von Woodstone steht eine möglichst flexible und universelle Überwachung von Systemen und Diensten. Es ist unter anderem möglich, beliebige Dienste auf ihren TCP oder auch UDP Ports zu überwachen. Letzteres erlaubt auch die Überwachung von besonderen Diensten wie RADIUS oder auch Spiele-Server. Web-Server werden dabei speziell unterstützt und sogar der über eines der Protokolle HTTP oder HTTPS sowie das Real Time Streaming Protocol (RTSP) [191] übertragene Inhalt kann überwacht werden. Eine weitere Besonderheit von SERVERS ALIVE ist die Unterstützung des Novell IPX Protokolls, so dass ein PING nicht nur über das ICMP Protokoll versendet werden kann, sondern auch an nicht TCP/IP-fähige Novell Server. Die Unterstützung von Novell findet sich auch bei der SNMP Überwachung wieder. Außerdem ist eine Spezialisierung für Microsoft Server und Datenbanken von Microsoft und Oracle enthalten. Schließlich können vom Administrator noch eigene Prüfmodule entwickelt werden, die sich problemlos in das Werkzeug integrieren lassen.

Flexible Benachrichtigung. Erhaltene Alarme können von SERVERS ALIVE an verschiedene Ziele zur Benachrichtigung der Administratoren gesendet werden. Möglich sind vor allem E-Mails sowie SMS und Pager Nachrichten, die auch zeitabhängig an unterschiedliche Personen gesendet werden können. Darüber hinaus können aber auch verschiedenste Aktionen ausgeführt werden. Neben der Möglichkeit zur Ausführung eines beliebigen Programms sind bereits einige Aktionen vordefiniert. Dies sind unter anderem der Neustart eines

[5] `www.woodstone.nu`

Dienstes oder eines ganzen Systems sowie das Versenden einer SNMP Nachricht an eine andere Managementstation oder das Verschicken einer Nachricht über das HTTP Protokoll an eine Web-Applikation. Es kann sogar ein „Wake-on-LAN“ Paket an ein beliebiges anderes Gerät über das Netzwerk gesendet werden.

Universelle Ausgabe. Die von SERVERS ALIVE erstellten Informationen werden über einen integrierten Web-Server visualisiert, was eine Überwachung des Netzwerkes von einem beliebigen Ort aus möglich macht. Die unterstützten Formate sind dabei so universell angelegt, dass sogar PDAs mittels des Wireless Application Protocol (WAP) [223] mit Web-Seiten versorgt werden können. Die Ergebnisse können außerdem aktiv mittels des File Transfer Protocol (FTP) oder des Simple File Transfer Protocol (SFTP) [110] auf einen FTP Server geladen werden. Das sichere Tool SFTP kann dazu ebenfalls verwendet werden.

Flexible Archivierung. Die Messwerte von SERVERS ALIVE können optional auch an das OpenSource Round-Robin Datenbankwerkzeug RRDTOOL [145] weitergeleitet werden, das eine langfristige und platzsparende Speicherung der Daten ermöglicht. Eine nähere Beschreibung des Round-Robin Datenbankwerkzeugs findet sich auf Seite 349.

Castle Rock Computing: SNMPC 7

Das Produkt SNMPC VERSION 7 des Unternehmens Castle Rock Computing[6] ist ein sicheres und verteiltes Netzwerkmanagementsystem, das eine Visualisierung, Überwachung und ein pro-aktives Management einer ganzen Netzwerk Infrastruktur erlaubt. SNMPC 7 unterstützt sowohl die beiden älteren SNMP Versionen SNMPv1 und SNMPv2 als auch die Funktionalitäten Authentifizierung und Verschlüsselung der neueren Version SNMPv3. Neben der Einbindung von SNMP können mit dem SNMPC 7 Werkzeug von Castle Rock Computing auch beliebige Dienste über ihren jeweiligen TCP Port überwacht werden. Die wichtigsten Produktmerkmale lassen sich wie folgt zusammenfassen:

Sicherheit. SNMPC 7 unterstützt auch die sichere SNMPv3 Version des SNMP Protokolls. Verwendet werden können die beiden im Standard definierten Authentifizierungsmodule HMAC-SHA-96 und HMAC-MD5-96 sowie der einzige definierte Verschlüsselungsalgorithmus CBC-DES. Die benutzerabhängige Zugriffsbeschränkung erlaubt außerdem eine individuelle Anpassung der Benutzeroberfläche und der Sichten auf die einzelnen OIDs für jeden Benutzer einzeln zu steuern und zu konfigurieren.

Skalierbarkeit. Castle Rock Computing bietet sein Produkt SNMPC 7 in zwei verschiedene Versionen an. Die ‚Workgroup Edition‘ zielt auf kleinere und

[6] www.castlerock.com

mittlere Netzwerke; mit der ‚Enterprise Edition' lassen sich größere Netzwerke von mehreren Benutzern administrieren.

Konfigurierbare Benachrichtigung. SNMPc 7 erlaubt bei Erhalt einer SNMP Nachricht die Ausführung verschiedener Aktionen. Dazu zählen die beiden Varianten für eine Benachrichtigung über E-Mail und über Pager sowie das simple Öffnen eines Alarmfensters oder auch das Starten einer beliebigen Applikation.

Flexibler Zugriff. SNMPc 7 unterstützt einen Zugriff auf die eigenen Anzeigen sowohl über einen proprietären Microsoft Client als auch über die standardisierte Web-Schnittstelle mittels eines eigenen Clients in der Programmiersprache Java. Mit der optionalen ‚Remote Access' Erweiterung für die ‚Enterprise Edition' können auch beliebig viele Benutzer gleichzeitig auf das System zugreifen.

Pro-aktives Management. Die Möglichkeit zur Planung einer automatischen Erstellung von Web-basierten Berichten über die Auslastung des Netzwerkes und die Verfügbarkeit verschiedener Dienste hilft einem Administrator beim pro-aktiven Management und der vorausschauenden Planung für das Netzwerk.

Integrationsfähigkeit. Um die Integrationsfähigkeit in andere Produkte zu erhöhen, können die von SNMPc 7 erstellten Graphen und Statistiken in verschiedene andere Formate exportiert werden. Dazu zählt auch eine reine Textausgabe sowie die Unterstützung der offenen Datenbankschnittstelle Open Data Base Connectivity (ODBC).

Herstellerneutralität. Durch die Verwendung des offenen Standards SNMP und der Möglichkeit zur Überwachung von beliebigen Diensten über ihren jeweiligen TCP Port macht sich SNMPc 7 unabhängig von allen Herstellern. Über die SNMP Schnittstelle können beispielsweise alle verfügbaren Informationen von Geräten beliebiger Hersteller ausgelesen und bei entsprechender Unterstützung auch konfiguriert werden. Zusätzlich können sowohl via SNMP als auch via ICMP die im Netzwerk verfügbaren Geräte automatisch entdeckt werden.

Flexible Anpassung. Die Flexibilität der Anpassung des SNMPc 7 Werkzeugs geht bis in die Unterstützung zum Importieren eigener MIBs. Auf diese Weise können letztlich beliebige OIDs über das Tool überwacht und verwaltet werden. Auch die Überwachung der TCP Dienste erlaubt eine flexible Konfiguration der an die jeweils entsprechenden Ports zu sendenden Zeichenketten und der zur Auswertung der erhaltenen Antworten notwendigen Suchmuster.

Computer Associates International, Inc.: UNICENTER NETWORK AND SYSTEMS MANAGEMENT

Das Unternehmen Computer Associates International Inc.[7] bietet mit ihrem Werkzeug UNICENTER NETWORK AND SYSTEMS MANAGEMENT ein flexible und umfangreiche Lösung zur Überwachung und Konfiguration von Netzwerken beliebiger Größe an. Der Fokus des Tools liegt in der Schaffung einer Möglichkeit des Netzwerkmanagements, welche die beste Leistung für den geringsten Aufwand liefert. Unterstützt werden nativ nur die SNMP Versionen SNMPv1 und SNMPv2. Die wichtigsten Merkmale sind im Folgenden zusammengefasst:

Integration in andere Unternehmensbereiche. Für eine optimale Nutzung der Netzwerkinfrastruktur sind nicht nur die einzelnen Dienste maßgeblich entscheidend, sondern auch andere Aspekte des IT-Managements. Zu diesen gehören beispielsweise auch die Speicherverwaltung, die unternehmensweit Dateidienste mit integriertem Backup anbieten, oder auch die Sicherheit, die sich in allen Bereichen der Informationstechnologie (IT) wiederfindet. Aus diesem Grund ist UNICENTER NETWORK AND SYSTEMS MANAGEMENT stark modular und als eine Produktlinie ausgelegt.

Flexible Unterstützung von Management-Software anderer Hersteller. Um eine möglichst optimale Nutzung auch bereits vorhandener Software-Produkte aus dem Bereich des Netzwerkmanagements zu gewährleisten, sind in UNICENTER NETWORK AND SYSTEMS MANAGEMENT besondere Schnittstellen zur Einbindung anderer Managementsysteme implementiert. Auf diese Weise kann das Werkzeug auch parallel zu anderen Netzwerkmanagementsystemen eingesetzt werden.

Plattformunabhängigkeit. Um auch stark heterogene Netzwerke möglichst gut administrieren zu können, unterstützt UNICENTER NETWORK AND SYSTEMS MANAGEMENT eine große Bandbreite an verschiedener Hardware, verschiedenen Software-Herstellern und Technologien.

Abbildung des Netzwerkmanagements auf Geschäftsprozesse. Eine besondere Eigenschaft von UNICENTER NETWORK AND SYSTEMS MANAGEMENT ist die Möglichkeit, das Netzwerk nicht nur topologisch und logisch strukturiert abzubilden, sondern auch direkt mit den Geschäftsprozessen in Verbindung zu setzen. Auf diese Weise können parallel zur Erforschung und Analyse der Ursachen eines Fehlers vor allem die Auswirkungen sichtbar gemacht werden. Gleichzeitig wird den für das Netzwerkmanagement verantwortlichen Personen eine Hilfestellung zur idealen Priorisierung aufgetretener Fehler gegeben, so dass Probleme mit großen Auswirkungen auf die Geschäftsprozesse des Unternehmens vorrangig angegangen werden können.

[7] www.ca.com

Reaktive Netzwerkkonfiguration. UNICENTER NETWORK AND SYSTEMS MANAGEMENT erlaubt bereits im Vorfeld die Definition von Verhaltensregeln, die bei übermäßiger Auslastung oder Ausfall einzelner Komponenten anzuwenden sind. So lassen sich Lösungen für potentielle Probleme bereits im Vorfeld spezifizieren, damit im Fehlerfall nicht erst eine aufwändige Fehlerisolierung und Lösungssuche durchgeführt werden muss, sondern die voreingestellten Reaktionen automatisch initiiert werden können.

Umfangreiche Erweiterbarkeit. Die Vielzahl an bestehenden modularen Funktionalitätsgruppen zur Erweiterung von UNICENTER NETWORK AND SYSTEMS MANAGEMENT liefern häufig eine Unterstützung von weiterführenden Technologien oder von Protokollen anderer Hersteller. Als eine kleine Auswahl an Beispielen seien hier mehrere Module angeführt, welche das Managementsystem um jeweils unterschiedlichste Aspekte erweitern. UNICENTER ADVANCED NETWORK OPERATION ist vor allem spezialisiert auf die Unterstützung anderer Technologien und Protokolle wie Frame Relay, Asynchronous Transfer Mode (ATM) [6], IPX, IBMs proprietäres Systems Network Architecture (SNA) [46] oder das proprietäre DECnet [84] Protokoll von Digital Equipment Corporation. Die WIRELESS NETWORK MANAGEMENT OPTION erlaubt eine Überwachung sowohl von mobilen Geräten als auch von Access-Points, die über ein WLAN an das Netzwerk angeschlossen sind. Mit dem Zusatzmodul UNICENTER MAINFRAME NETWORK MANAGEMENT lassen sich vor allem klassische Mainframes verwalten wie etwa IBM Server unter dem „zero down-time" (z/OS) [234] Betriebssystem, das auch unter dem Namen OS/390 [148] bekannt geworden ist. Schließlich liefert das UNICENTER MANAGEMENT PORTAL eine flexible und sichere Oberfläche zur Verwaltung des gesamten Netzwerks. Als weiterführende Ergänzungen können auch Produkte anderer Hersteller in UNICENTER NETWORK AND SYSTEMS MANAGEMENT integriert werden. Beispielsweise erweitert die METACONSOLE von Netaphor Software Inc.[8] die Management-Software um eine Unterstützung des SNMPv3 Protokolls.

Ipswitch Inc.: WHATSUP GOLD 8

Das vom Unternehmen Ipswitch Inc.[9] entwickelte Tool WHATSUP GOLD 8 bietet eine Netzwerküberwachung mit Hilfe des SNMP Protokolls sowie eine Überwachung häufiger Netzwerkdienste über deren TCP Port an. Es werden lediglich die unsicheren SNMP Versionen SNMPv1 und SNMPv2 unterstützt. Die herausragendsten Produktmerkmale sind im Folgenden aufgelistet:

Automatische Erkennung von Systemen und Diensten. Eine der besonderen Funktionalitäten von WHATSUP GOLD 8 besteht in der automatischen Erkennung von Systemen und Diensten. Die im Netzwerk vorhandenen

[8] www.netaphor.com
[9] www.ipswitch.com

Systeme können auf eine von fünf möglichen Weisen ermittelt werden. Dazu zählt nicht nur eine allgemeine Methode über das ICMP Protokoll, sondern auch einige Microsoft proprietäre Methoden über die Netzwerkumgebung, die Registrierung oder die `hosts` Datei. Schließlich existiert auch noch die Möglichkeit der automatischen Erkennung von Systemen im Netzwerk über den SNMP Mechanismus, bei dem vorzugsweise Router als Informationsquelle dienen. Während der automatischen Analyse des Netzwerks mittels SNMP Protokoll kann zugleich eine automatische Erkennung von Diensten auf den jeweiligen Komponenten durchgeführt werden.

Universelle Visualisierung. Die durch die automatische Erkennung im Netzwerk aufgefundenen Komponenten können auch mit ihren Verbindungen untereinander dargestellt werden. Je nach verwendeter Erkennungsmethode kann dies entweder automatisch geschehen oder der Administrator kann selbst die Verbindungslinien einarbeiten. Zur Erleichterung ist in WHATSUP GOLD 8 auch eine *Traceroute* Funktionalität integriert, mit welcher die auf dem Kommunikationsweg zwischen zwei Systemen liegenden Netzwerkgeräte erkannt und in die Visualisierungskomponente integriert werden können.

Umfangreiche Überwachung. Eine der flexibel implementierten Lösungen ist die Überwachung der Erreichbarkeit von Systemen im Netzwerk. Das Tool von Ipswitch bietet dafür eine ganze Reihe von Möglichkeiten. Neben der einfachsten Methode mittels ICMP Protokoll ist außerdem eine Überwachung über das TCP oder UDP Protokoll möglich. Als Besonderheit werden für diese Aufgabe auch noch die beiden proprietären Protokolle Internetwork Packet Exchange (IPX) von Novell sowie das Network Basic Input/Output System (NetBIOS) von Microsoft unterstützt. Zusätzlich zur Erreichbarkeit von Systemen kann auch die Verfügbarkeit von TCP und UDP Diensten auf den jeweiligen Ports direkt überwacht werden. Als weitere Besonderheit lassen sich schließlich noch Microsoft-eigene Dienste überwachen.

Vielfältige Benachrichtigung. WHATSUP GOLD 8 ist in der Lage, aus den ermittelten Werten besondere Situationen zu erkennen und entsprechende Alarme zu generieren. Insbesondere die Möglichkeit zum Empfang von SNMP Nachrichten und SYSLOG Meldungen liefert eine gute Quelle für Alarme. Außerdem können die auf dem eigenen System erzeugten Einträge der Microsoft Systemprotokollierung ausgewertet werden. Im Falle eines Alarms können dann verschiedenste Ziele als Empfänger der zugehörigen Benachrichtigung spezifiziert werden. Neben der Weiterleitung der Informationen an die Administratoren via Pieper, Pager, SMS oder E-Mail können auch akustische Signale gegeben oder Alarmfenster geöffnet werden. Zusätzlich kann die Erzeugung einer eigenen SYSLOG Meldung oder aber auch der Neustart eines Dienstes oder eines ganzen Systems initiiert werden. Die Ausführung eines beliebigen Programms kann ebenfalls spezifiziert werden.

Fernüberwachung. WHATSUP GOLD 8 kann sowohl über die Textkonsole als auch über die Web-Schnittstelle administriert werden. Dies ermöglicht ei-

nem Administrator die flexible Verwaltung des Netzwerks von fast jedem Ort aus.

Hilfreiches Toolset. In das Hauptprogramm integriert sind eine ganze Reihe von kleineren, nützlichen Werkzeugen, mit denen weitere Informationen über das Netzwerk und die darin befindlichen Systeme eingeholt werden können. Neben den bekannteren Tools wie PING oder TRACEROUTE sind auch besondere Werkzeuge beispielsweise zum Durchsuchen von Verzeichnis-Servern über das Lightweight Directory Access Protocol (LDAP) oder zur Durchführung einer Zeitsynchronisation enthalten.

Flexible Archivierung und Berichterstattung. Mit WHATSUP GOLD 8 können alle erzeugten Alarme und andere wichtige Ereignisse in einer eigenen Datei archiviert werden. Über dieses Archiv ist es auch möglich, vordefinierte und selbst erstellte Berichte zu generieren. Dieser Vorgang lässt sich zudem noch automatisieren.

11.1.2 Herstellereigene Lösungen

Unter den herstellereigenen Lösungen sollen an dieser Stelle vor allem Werkzeuge von Systemherstellern sein, die mit der Management-Software eine Überwachung und Konfiguration der eigenen Hardware liefern wollen. In vielen Fällen sind diese Programme sehr umfangreich und nicht auf die eigene Hardware beschränkt. Es finden sich allerdings auch reine Software-Entwicklungen, deren Funktionalität vergleichbar komplex ist. Die Größe dieser Systeme ermöglicht fast immer ein Management des kompletten Netzwerks mit allen Systemen. Die hohe Funktionalität hat allerdings auch seinen Preis: Häufig ist die Anschaffung dieser universell und breitgefächert angelegten Tools mit einem gleichermaßen großzügigen Kaufpreis verbunden.

Netgear, Inc: PROSAFE NMS 100

Das Unternehmen Netgear Inc.[10] bietet unter dem Namen PROSAFE NETWORK MANAGEMENT SYSTEM NMS 100 ein Werkzeug zur Überwachung und Konfiguration von Netzwerken an, das besonders durch seine umfangreiche Unterstützung der bekannten Standards überzeugt. Neben der Möglichkeit zur Überwachung von TCP Diensten steht vor allem die Netzwerkverwaltung über das Protokoll SNMP im Vordergrund. Dabei sind sowohl die älteren Versionen SNMPv1 und SNMPv2 sowie die neuere Version SNMPv3 mit allen definierten Sicherheitsmechanismen implementiert. Zusätzlich ist auch eine RMON Überwachung möglich, die insbesondere von den hauseigenen Switches unterstützt wird. Die wichtigsten Funktionalitäten sind im Folgenden aufgelistet:

[10] `www.netgear.de`

Automatische Erkennung. ProSafe NMS 100 verwendet für die Erzeugung der dargestellten Netzwerkpläne einen automatischen Erkennungsmodus, der alle erkannten Geräte in eine hierarchische Struktur abbildet. Diese interne Datenstruktur erlaubt eine automatische Anordnung von Icons für die einzelnen Komponenten, welche der logischen Struktur des Netzwerks entspricht. Der automatische Mechanismus zur Anordnung der Icons kann auch zu Gunsten der Möglichkeit einer manuellen Konfiguration deaktiviert werden. Die automatische Erkennung von neuen Geräten bleibt dadurch aber unbeeinflusst.

Benutzerfreundliche Visualisierung. Die Anzeige der einzelnen Netzwerkkomponenten erfolgt in dem Tool von Netgear bei automatischer Erkennung zweistufig mit einer Ebene für vermittelnde Netzwerkkomponenten und einer Ebene mit mehreren Instanzen für die Endgeräte in den erkannten Netzwerken. Bei der manuellen Konfiguration wird die Anzeige um eine weitere Instanz der zweiten Ebene mit neu erkannten Geräten erweitert, so dass die vom Benutzer durchgeführten Individualisierungen nicht verloren gehen und dennoch der Vorteil einer automatischen Erkennung von neuen Geräten genutzt werden kann. Die Darstellung der mittels SNMP ermittelten nummerischen Werte kann in mehreren Varianten erfolgen, beispielsweise als einfacher Graph, als Balkendiagramm oder auch als Tortendiagramm.

Umfangreiche Überwachung. Von den in der zweistufigen Hierarchie dargestellten Geräten kann nicht nur die Erreichbarkeit via ICMP überwacht werden, sondern auch die Verfügbarkeit von vorkonfigurierten oder benutzerdefinierten TCP Diensten und SNMP Objekten. Außerdem können mit ProSafe NMS 100 die im RMONv1 spezifizierten Datenquellen abgerufen werden. Das Tool ist in dieser Hinsicht besonders auf Geräte aus dem eigenen Hause ausgerichtet, erlaubt aber auch die Abfrage anderer RMON-fähiger Komponenten.

Sicherheit. Bei der SNMP Überwachung kann auf alle drei wichtigen Versionen inklusive der Sicherheitsmechanismen von SNMPv3 zurückgegriffen werden. Dies betrifft die Authentifizierungsmodule HMAC-SHA-96 und HMAC-MD5-96 sowie den einzigen im SNMP Standard definierten Verschlüsselungsalgorithmus CBC-DES. Auf diese Weise ist auch ein Einsatz unter unsicheren Bedingungen möglich.

Flexible Benachrichtigung. Im Falle eines Alarms kann der Administrator sich von ProSafe NMS 100 entweder per Pager oder per E-Mail informieren lassen. Zu den Situationen, in denen ein Alarm erzeugt wird, zählen vor allem Ereignisse wie der Verlust der Erreichbarkeit eines Gerätes oder der Verlust der Verfügbarkeit eines Dienstes. Außerdem können für nummerische Werte, die über SNMP ermittelt werden, die Ereignisse des Überschreitens oder Unterschreitens eines spezifizierten Schwellwertes einen Alarm auslösen. Die Berechnung des Schwellwertes kann sogar automatisch durchgeführt werden.

In diesem Fall wird innerhalb einer Lernphase eine Baseline für den jeweiligen Wert ermittelt und daraus ein Schwellwert abgeleitet.

Hewlett-Packard: OpenView Network Node Manager

Ursprünglich zur Unterstützung der eigenen Hardware entwickelt stellt der OpenView Network Node Manager von Hewlett-Packard[11] in der aktuellen Version ein überaus flexibles und nahezu beliebig erweiterbares Netzwerkmanagementsystem dar. Vor allem die großzügige Skalierbarkeit auf jede Netzwerkgröße macht dieses Tool vor allem für Administratoren größerer Netzwerke besonders attraktiv. Im Folgenden soll eine Auswahl der wichtigsten Merkmale aufgelistet werden:

Automatische Erkennung. Der OpenView Network Node Manager kann selbst in großen, stark verschachtelten Netzwerkstrukturen noch eine automatische Erkennung der vorhandenen Komponenten und Dienste erfolgreich absolvieren. Gerade um mit großen Netzwerken umgehen zu können, werden die erkannten Geräte automatisch inventarisiert und für eine Überwachung vorkonfiguriert.

Benutzerfreundliche Darstellung. Durch eine hierarchische Vorgehensweise ist der OpenView Network Node Manager in der Lage, auch komplexere Netzwerke abzubilden und den Administratoren verständlich zu präsentieren. Die graphische Darstellung soll vor allem einen leichten Einstieg und einen schnellen Überblick ermöglichen.

Schnelle Fehlererkennung und Ursachenfindung. Eines der herausragendsten Merkmale von HPs Netzwerkmanagement-Software ist die Unterstützung der Administratoren bei der Erkennung von Fehlern und bei der Findung der zugrunde liegenden Ursachen. Zu diesem Zweck werden zunächst die eingehenden Alarme nach einem bereits gut vordefinierten Algorithmus sortiert und priorisiert. Zur Feinabstimmung lassen sich die zugehörigen Regeln noch weiter anpassen und ergänzen. Die als kritisch angesehenen Alarme werden dem Benutzer besonders signalisiert und können auf einen Klick erreicht werden. Zur Ermittlung der Ursache des Problems kann der OpenView Network Node Manager in begrenztem Maße auch kausale Beziehungen zwischen Alarmen identifizieren und somit je nach Situation die Fehler einem Verursacher zuordnen. Auf diese Weise kann ein Administrator schneller die Ursache eines Problems erkennen und entsprechende Aktionen durchführen.

Flexible Berichterstattung und Benachrichtigung. Der OpenView Network Node Manager sammelt sämtliche Alarme in einer frei wählbaren Datenbank und kann daraus vielfältige vordefinierte oder benutzerspezifizierte Berichte erstellen. Auf diese Weise kann nicht nur sehr einfach eine vollständige Inventarisierung des Netzwerkes vorgenommen werden, sondern

[11] `www.hp.com`

es können auch verschiedene Berichte zur Verfügbarkeit und Auslastung der einzelnen Komponenten und logischen Teile des Netzwerks erstellt werden. Eine automatische Erstellung von Baselines für nummerische, über SNMP ermittelte Werte erlaubt auch eine einfache Justierung der für eine Erzeugung entsprechender Alarme notwendigen Schwellwerte. Jeder Alarm kann zusätzlich zu einer Benachrichtigung der zuständigen Administratoren führen.

Modularisierung und Erweiterbarkeit. Der strikt modulare Aufbau von OpenView Network Node Manager erlaubt eine weitgehend flexible Erweiterung des Managementsystems. Zwar wird den Administratoren keine offene Schnittstelle zur einfachen Implementierung eigener Module geliefert, in der OpenView Produktfamilie existiert jedoch bereits eine große Anzahl nützlicher Ergänzungen. Diese fokussieren größtenteils auf neuartige oder proprietäre Protokolle, die in vielen Netzwerken nicht zum Einsatz kommen. Beispiele sind das Protokoll IPv6, das Multiprotocol Label Switching (MPLS) [186] Protokoll, Multicasting oder Voice over IP (VoIP) Protokolle. Ein spezielles Modul zur Überwachung von Frame Relay [5] Verbindungen hilft bei der Identifizierung von Problemen der Internetanbindung über externe ISPs. Wenn die Frame Relay MIB [23] von den Zugangsroutern unterstützt wird, kann die Ursache des Problems sogar der eigenen Seite oder dem Verantwortungsbereich des Providers zugewiesen werden. Ein weiteres Modul dient der Überwachung von redundant ausgelegten Routern, welche das Cisco-eigene Protokoll Hot Standby Router Protocol (HSRP) [209] verwenden.

IBM: Tivoli NetView

Das Werkzeug Tivoli NetView aus dem Hause IBM[12] kann zwar insbesondere mit den eigenen Geräten umgehen, es hat sich mittlerweile jedoch zu einer äußerst flexiblen Management-Software entwickelt, die eine Unterstützung für viele auf dem Markt befindliche Netzwerkkomponenten und Server-Systeme liefert. Einen seiner Schwerpunkte hat das Tool bei der Unterstützung einer verteilten Administration, die besonders in großen und komplexeren Netzwerken vorteilhaft eingesetzt werden kann. Im Folgenden finden sich die wichtigsten Merkmale des Tools:

Klare Definition wichtiger Sicherheitsvorgaben. Häufig finden sich die zu verwaltenden Geräte des Netzwerkes in Bereichen, in denen nur ein eingeschränkter Zugang besteht, da eine Firewall den entsprechenden Netzwerkverkehr blockiert. In diesen Fällen ist auch die Überwachung und Konfiguration der entsprechenden Netzwerkkomponenten nur schwer möglich. Tivoli NetView beschreibt aus diesem Grund detailliert, welche Firewall-Regeln zum korrekten Funktionieren der Management-Software einzurichten sind.

Besondere Unterstützung von Basis-Netzwerkkomponenten. Tivoli NetView unterstützt vor allem eine ausführliche Überwachung der grundle-

[12] www.ibm.com

genden Netzwerkkomponenten wie Switches oder Bridges. Durch die umfangreiche Implementierung des MPLS Protokolls können diese Geräte besonders intensiv überwacht werden. Selbst Angaben zum Spanning Tree [213] können auf diese Weise ermittelt werden. So erhalten die Administratoren immer einen ausführlichen Blick auf die Kernkomponenten des Netzwerks, welche den Backbone bilden. Als Besonderheit integriert sich das Tool von IBM optimal in die Management-Software des Herstellers Cisco, so dass eine ideale Unterstützung von Cisco Routern und Switches gegeben ist.

Regelbasierte Überwachung. Durch die Bildung von Gruppen können in TIVOLI NETVIEW mehrere Geräte zusammengefasst werden, die auf dieselbe Art und Weise überwacht werden sollen. Zwar ist es immer noch möglich, jedes Gerät für sich zu konfigurieren, aber durch die Einordnung der Systeme in Gruppen können parallel dazu die Überwachungsaufgaben definiert werden. Mögliche vorgegebene Gruppierungen fassen beispielsweise alle Geräte in einer Gruppe zusammen, die vom gleichen Hersteller oder Typ sind, sich am gleichen Standort befinden oder aber gleiche Dienste anbieten. Diese Einstellungen betreffen dann auch automatisch neu erkannte Geräte.

Flexible Benachrichtigung. Eingehende Alarme können in TIVOLI NETVIEW automatisch verschiedene Reaktionen auslösen. Unter anderem ist eine Benachrichtigung der Administratoren via Pager oder E-Mail möglich. Zusätzlich kann ein Alarm auch die Ausführung eines beliebigen Programms anstoßen, so dass letztlich beliebige Reaktionen konfiguriert werden können. Bei einer Benachrichtigung kann außerdem in eingeschränktem Maße unterschieden werden, ob es sich um eine Hauptursache oder einen Folgefehler handelt. Auf diese Weise können sekundäre oder unwichtige Alarme ausgefiltert und den Administratoren nur die vorrangigen Alarme übermittelt werden.

Intelligente Fehlerisolierung. TIVOLI NETVIEW ist in der Lage, die durch einen Ausfall einer vermittelnden Komponente hervorgerufene Nichterreichbarkeit einzelner Systeme und ganzer Netzwerke zu erkennen und als Folgefehler zu interpretieren. Auf diese Weise kann im Fehlerfall die Menge an Alarmen auf ein Minimum reduziert werden. Gleichzeitig erleichtert das Ausfiltern der sekundären Alarme eine Identifikation der Hauptursache des Fehlers, so dass entstandene Probleme schneller behoben werden können.

Fernüberwachung. Der Zugriff auf IBMs Tool über die Web-Schnittstelle ermöglicht eine Verwaltung des Netzwerkes mit seinen Komponenten von fast jedem Punkt der Welt aus. Neben der reinen Überwachung können über diese Schnittstelle auch kleinere Werkzeuge wie PING oder TRACEROUTE ausgeführt werden.

Verteiltes Netzwerkmanagement. Durch die Unterteilung des Netzwerkes in unterschiedlich überwachte und konfigurierte Regionen ist mit TIVOLI NETVIEW recht einfach eine verteilte Verwaltung des Netzwerkes möglich. Beispielsweise können lokale Administratoren die lokalen Geräte an den ein-

zelnen Standorten verwalten, während gleichzeitig eine andere Gruppe die globalen Kommunikationswege administriert.

Integrationsfähigkeit. TIVOLI NETVIEW bietet nicht nur eine Integrationsfähigkeit für das CISCO WORKS 2000 Tool aus dem Hause Cisco, sondern vor allem für ergänzende eigene Produkte. Beispielsweise bietet TIVOLI INVENTORY eine Inventarisierung der Netzwerkkomponenten und TIVOLI DECISION SUPPORT NETWORK GUIDES hilft bei der genaueren Analyse aller gesammelter Informationen.

Sicherheit. TIVOLI NETVIEW unterstützt auch die sichere Version 3 des SNMP Protokolls. Verwendet werden können alle im Standard definierten Authentifizierungsmodule und Verschlüsselungsalgorithmen. Zur Zeit sind dies bei der Authentifizierung HMAC-SHA-96 und HMAC-MD5-96 sowie bei der Verschlüsselung der CBC-DES Algorithmus.

11.1.3 OpenSource Tools

Im Bereich des Netzwerkmanagements findet sich eine große Anzahl von kommerziellen Produkten auf dem Markt. Diese Vielfalt ist auf Grund des hohen Bedarfs aber vor allem auch wegen der darunterliegenden offenen Standards entstanden. Parallel dazu ist im Bereich des Netzwerkmanagements eine gleichermaßen aktive OpenSource Gemeinde entstanden, welche eine Entwicklung der verschiedensten Tools aus den Bereichen Netzwerküberwachung und Netzwerkverwaltung vorantreibt. Obwohl eine OpenSource Software nicht zwangsweise kostenlos sein muss, können die meisten der hier vorgestellten und der anderen, an den unterschiedlichsten Stellen des Internets zu findenden Tools frei bezogen und eingesetzt werden. Dies macht den Einsatz von OpenSource Tools gerade für Administratoren kleinerer Netzwerke, die nicht über einen entsprechend großes Budget verfügen, überaus attraktiv. Man sollte allerdings nicht vergessen, dass beständige Weiterentwicklung und Verbesserung zu den Grundprinzipien von OpenSource ist; und eine stabile Zwischenversion wird nicht immer explizit erstellt.

ARGUS

Die Hauptaufgabe von ARGUS[13] liegt eindeutig in der Überwachung von Objekten. Der selbstgewählte Wahlspruch *In God we trust, everything else we monitor.*[14] verdeutlicht gleichzeitig die Ausrichtung auf eine möglichst flexible und universelle Überwachung. ARGUS ist in der Programmiersprache Perl entwickelt, so dass auch eine einfache Erweiterung des Tools möglich ist. Die wichtigsten Besonderheiten werden im Folgenden zusammengefasst:

[13] `argus.tcp4me.com`

[14] Die Übersetzung dieses Wahlspruchs lautet: „Auf Gott vertrauen wir, alles andere überwachen wir."

Universelle Überwachung. ARGUS kann beliebige TCP und UDP Dienste überwachen, von denen bereits einige vorkonfiguriert sind. Zu diesen zählen beispielsweise die Dienste aus den Bereichen SMTP, FTP, NNTP, HTTP, SSH oder DNS. Außerdem ist eine Überwachung der Ergebnisse von beliebigen Programmen möglich. Dies schließt auch die Überwachung von Datenbanken mit ein, die über frei konfigurierbare Structured Query Language (SQL) [210] Abfragen überwacht werden können. Schließlich können auch beliebige SNMP Objekte überwacht werden, wobei eine Unterstützung sowohl der unsicheren Versionen SNMPv1 und SNMPv2 sowie der sicheren Version SNMPv3 inklusive Authentifizierung und Verschlüsselung besteht. Eine Implementierung für das IPv6 Protokoll rundet das Potential von ARGUS ab.

Flexible Benachrichtigung. Wie bei vielen anderen Produkten aus dem Bereich der Netzwerküberwachung wird auch bei ARGUS eine Benachrichtigung der Administratoren bei Erkennen eines Alarms unterstützt. Mögliche Kommunikationswege sind dabei E-Mails oder Pager Nachrichten. Die umfangreiche Verwaltung von Benutzern ermöglicht außerdem eine feingranulare Zuordnung von Alarmen und Alarmtypen zu einzelnen Benutzern und Benutzergruppen.

Intuitive Darstellung. In ARGUS werden die Ergebnisse – wie in vielen anderen Programmen auch – intuitiv verständlich präsentiert. Zwar ist die Konfiguration des Tools nicht ganz so simpel, wie das Navigieren durch die via Web-Schnittstelle präsentierten Ergebnisse; die überaus große Flexibilität bei der Konfiguration führt aber indirekt auch gleichzeitig zu einer besonders fein anpassbaren Oberfläche. Alle Design-Elemente sind außerdem vom Administrator frei konfigurierbar, so dass ein individuelles Aussehen leicht einstellbar ist.

BIG SISTER

Das bei SourceForge verwaltete OpenSource Tool BIG SISTER[15] basiert auf dem ebenfalls quelloffenen Tool BIG BROTHER und stellt ein universelles Netzwerküberwachungs-Tool dar. Das Werkzeug ist in der Programmiersprache Perl implementiert und erlaubt nicht nur deshalb eine ideale Integration auch anderer OpenSource Tools wie MRTG oder RRDTOOL. Die wichtigsten Eigenschaften von BIG SISTER werden im Folgenden zusammengefasst:

Strikte Aufgabenteilung. BIG SISTER verwendet eine Client-Server Umgebung, in der ein einzelner Server für die Archivierung, Verarbeitung und Präsentation der Messwerte verantwortlich ist. Auf den überwachenden Geräten muss außerdem die Client-Software UXMON installiert werden, die für die Sammlung der Daten vom lokalen Host und die anschließende Weiterleitung zum BBD Server verantwortlich ist. Auf dem Server werden die Daten

[15] `bigsister.sourceforge.net`

gespeichert und ausgewertet, wobei erkannte Alarme auch zur Ausführung beliebiger Programme führen können. Das Teilprogramm BSMON ist schließlich für die Darstellung der Ergebnisse der Überwachung verantwortlich.

Universelle Überwachung von Objekten. Da BIG SISTER aus einer Client-Server Architektur besteht, bei der jeder zu überwachende Host mit einer eigenen Client-Software zu versehen ist, können prinzipiell auch beliebige Objekte überwacht werden. Ein auf dem Zielsystem installierter Sensor hat die besten Möglichkeiten, um an alle interessanten Werte zu gelangen. Eine ganze Reihe von Objekten wird bereits nativ vom Werkzeug unterstützt, zu denen neben einer ICMP Unterstützung vor allem auch verschiedene TCP Dienste zählen. Als Beispiel sollen hier die Dienste aus den Bereichen HTTP, NTP oder LDAP dienen. Es können aber auch andere frei definierbare TCP Dienste überwacht werden. Außerdem können verschiedenste Objekte über SNMP abgefragt werden wie beispielsweise die Prozessorauslastung, die Hauptspeicherauslastung, die Festplattenbelegung, die Anzahl der laufenden Prozesse oder auch der Status der Unterbrechungsfreien Stromversorgung (USV). Schließlich ist auch eine Überwachung beliebiger SNMP Objekte wie beispielsweise Netzwerkschnittstellen möglich. Mit der Möglichkeit, die Ausgabe eines beliebigen Programms zu überwachen und an den BBD Server weiterzuleiten, kann der Client universell jede beliebige Datenquelle auslesen.

Sicherheit. Zwar unterstützt BIG SISTER keine der Sicherheitsmechanismen von SNMP, die erst ab der Version SNMPv3 zur Verfügung stehen, jedoch kann auf eine andere Art eine Sicherheit in das verteilte System gebracht werden. Die strikte Trennung zwischen Server und Clients hat zur Implementierung eines eigenen Kommunikationsweges geführt. Wie bereits beim Vorgänger BIG BROTHER wird dazu der TCP Port 1984[16] verwendet. Optional kann die gesamte Kommunikation von BIG SISTER durch einen SSH Tunnel gesendet werden, wodurch vor allem eine Verschlüsselung der übertragenen Daten erfolgt. Außerdem können die beteiligten Systeme auf eine sichere Weise konfiguriert werden, die einen Aufbau eines SSH Tunnels nur von ausgewählten vertrauenswürdigen Hosts erlaubt. Ein weiterer Aspekt betrifft die Konfiguration des Servers. Diesem kann explizit mitgeteilt werden, welche Hosts sich zu welchem Zweck verbinden dürfen.

Flexible Benachrichtigung. Alarme können in BIG SISTER frei über ein beliebiges Suchmuster spezifiziert werden. Jeder dieser Alarme kann zu einer Benachrichtigung der Administratoren führen. Dazu kann entweder eine E-Mail oder auch eine Pager Nachricht an die jeweils zuständigen Administratoren gesendet werden.

[16] Nicht nur der Name des Programms BIG BROTHER liefert eine deutliche Anspielung an den berühmten Roman „1984“ von George Orwell [147], sondern auch der verwendete TCP Port unterstreicht die Herkunft des Namens.

Fernüberwachung. Die Anzeige der Ergebnisse von BIG SISTER erfolgen über die Web-Schnittstelle des auf dem Server installierten Web-Servers. Die graphische Oberfläche ist dabei weitgehend konfigurierbar. Neben einer beliebig hierarchisch spezifizierbaren Netzwerktopologie kann vor allem auch das äußere Erscheinungsbild frei angepasst werden. Die Verwendung des HTTP Mechanismus erlaubt außerdem eine Überwachung des Netzwerks von beliebigen Geräten aus.

CRICKET

Das OpenSource Tool CRICKET ist stark an die von Tobias Oetiker entwickelten Werkzeuge MRTG und RRDTOOL angelehnt. Tatsächlich wurden die Autoren des bei SourceForge verwalteten Tools stark vom „Multi Router Traffic Grapher" inspiriert. Daraus ist auch eine engere Zusammenarbeit zwischen den beiden Entwicklern entstanden. Nicht zuletzt wegen der erweiterten Anforderungen, welche die Autoren von CRICKET an MRTG hatten, ist letztlich das „Round-Robin Database Tool" ins Leben gerufen worden. Die von RRDTOOL gelösten Probleme wurden größtenteils auch von den CRICKET Entwicklern erkannt. Die Besonderheiten des Überwachungswerkzeugs und Visualisierungs-Tools werden im Folgenden zusammengefasst:

Performanz- und Speicherplatz-optimierte Datenhaltung. Die Speicherung der Daten erfolgt in CRICKET über das RRDTOOL (siehe Seite 349), das somit zugleich ein fester Bestandteil des Tools ist. Die Entwicklung der Round-Robin Datenbank wird zwar unabhängig von Tobias Oetiker durchgeführt, jedoch arbeiten die Entwickler der beiden Teams zusammen. Zu den Vorteilen und den Nachteilen bezüglich der Datenhaltung von CRICKET gelten daher die für RRDTOOL gemachten Aussagen.

Skalierbarkeit. Vor allem aus Gründen der Performanz ist MRTG nur bedingt skalierbar gewesen. Entsprechend wurde bei der Entwicklung von CRICKET auf eine bessere Performanz und damit auch bessere Skalierbarkeit geachtet. Die Performanzprobleme werden durch den Einsatz der RRDTOOL Datenbank gelöst; für eine bessere Skalierbarkeit wurde eine Konfigurationsmöglichkeit in einer baumartigen Struktur implementiert. Durch diese logische Strukturierung können vor allem Vererbungsmechanismen in die Funktionalitäten von MRTG integriert werden. Im hierarchischen Konfigurationsmodell können außerdem Mechanismen wie eine Zeichenkettenersetzung verwendet werden, die ein Arbeiten vergleichbar mit einer Unix-Konsole erlaubt. Um auch eine größere Anzahl an parallel überwachten Netzwerkkomponenten zu unterstützen, müssen mehrere CRICKET Server gleichzeitig gestartet werden, die jeweils einen Teil der Geräte überwachen. Während diese Aufgabe in MRTG noch manuell von den Administratoren durchgeführt werden musste, erlaubt der hierarchische Konfigurationsbaum auch eine einfache Unterteilung und Delegation der Aufgaben an parallele Instanzen von CRICKET.

JFFNMS

Bereits mit seiner Namensgebung macht das bei SourceForge[17] verwaltete OpenSource Tool ‚Just For Fun Network Management System' (JFFNMS)[18] auf sich aufmerksam. Das ursprüngliche Hauptziel von JFFNMS bestand im Sammeln und in der Darstellung von verschiedenen Informationen rund um Schnittstellen. Die Erweiterungen der letzten Jahre haben aber den Begriff „Schnittstelle" veralten lassen, da sich nun beinahe beliebige Objekte dahinter verbergen können, die einen Status oder einen Wert annehmen können. JFFNMS ist in der Programmiersprache PHP [157] entwickelt und hat bereits einen stabilen Zustand erreicht, der auch den Einsatz in einer Produktivumgebung erlaubt. Die wichtigsten Eigenschaften der Software sind im Folgenden zusammengefasst:

Benutzerabhängige Netzwerkverwaltung. Das Netzwerkmanagementsystem JFFNMS unterscheidet verschiedene Benutzer, denen unterschiedliche Berechtigungen zugewiesen werden können. Die Hauptunterscheidung findet zwischen den beiden Benutzergruppen „Kunden" und „Administratoren" statt. Während jedes überwachte Objekt einem Kunden zugewiesen werden muss, können Administratoren nicht Besitzer von Objekten sein. Kunden besitzen lediglich die Berechtigung, sich Informationen über die ihnen zugewiesenen Objekte anzeigen zu lassen. Administratoren können JFFNMS-eigene Einstellungen vornehmen und die Überwachung der für sie freigegebenen Objekte konfigurieren.

Flexible Generierung von Alarmen. Grundsätzlich existieren in JFFNMS drei verschiedene Quellen für Ereignisse. Die erste Art von Ereignis entstammt der Überwachung der konfigurierten Objekte. Hier kann beispielsweise das Überschreiten oder das Unterschreiten eines Schwellwertes sowie die Änderung eines Statuszustands ein Ereignis auslösen. Als zweite Quelle können SNMP Nachrichten dienen und die dritte Quelle bedient sich bei den eingehenden SYSLOG Meldungen. Neben der Unterstützung von SYSLOG-NG, mit dem sich bei Eintreffen einer Meldung beliebige Aktionen auslösen lassen, liefert das Netzwerkmanagementsystem auch einen eigenen SYSLOG Server mit dem Namen MSYSLOG. Dieser ermöglicht ebenfalls eine separate Behandlung aller eingehenden SYSLOG Meldungen. Alle Ereignisse besitzen eine Kritikalität und können prinzipiell dazu konfiguriert werden, einen Alarm eines beliebigen Schweregrades zu generieren.

Flexible Benachrichtigung. Jedes beliebige Ereignis kann in JFFNMS einen Alarm auslösen. Mit diesem kann dann eine Benachrichtigung der Administratoren via E-Mail oder Pager verbunden sein. Auch die Ausführung eines beliebigen Programms ist möglich, so dass letztlich volle Flexibilität bei der Reaktion auf Alarme besteht.

[17] `www.sourceforge.net`

[18] `jffnms.sourceforge.net`

Unterstützung vielfältiger Objekte. Die überwachten Objekte in JFFNMS entspringen der ursprünglichen Ausrichtung auf Netzwerkschnittstellen. Aus diesem Grund können die Objekte einen Statuszustand wie beispielsweise „aktiviert“ oder „inaktiviert“ besitzen oder einen Wert enthalten wie beispielsweise die Auslastung. Im Laufe der Zeit wurde der Fokus von Schnittstellen auch auf beliebige andere Objekte ausgeweitet, die über einen Status oder einen Wert verfügen. Es existiert bereits eine Liste von vorkonfigurierten Objekten, die mit JFFNMS abgefragt werden können. Beispielhaft seien hier mehrere dieser Objekte aufgeführt. Eines der generischen Objekte ist das TCP Objekt, mit dem sich beliebige TCP Dienste überwachen lassen. Mit Hilfe des ebenfalls quelloffenen Tools NMAP [142] wird eine Verbindung zum angegebenen Port hergestellt und die erhaltene Antwort anschließend mit einem Suchmuster verglichen. Vom APACHE Web-Server [71] lassen sich verschiedene Informationen abfragen wie beispielsweise die Anzahl aller eingegangenen Anfragen, das gesamte zurückgesendete Datenvolumen oder die Anzahl der untätigen und der aktiven APACHE Prozesse. Vom Microsoft Internet Information Server (IIS) [90] können im Wesentlichen die Anzahl der gesendeten und empfangenen Bytes ermittelt werden. Es existieren auch Objekte, die auf einzelne Betriebssysteme zugeschnitten sind, wie beispielsweise das WINDOWS Objekt und das Solaris Objekt. Die Funktionalität beschränkt sich aber auf das Auslesen und Anzeigen von Informationen wie Prozessorauslastung oder Speicherbelegung. Als letztes Beispiel soll das Reachability Objekt dienen, mit dessen Hilfe sich die Erreichbarkeit einer Netzwerkkomponente ermitteln lässt. Als Basis wird hier das ICMP ToolPING verwendet.

Verteiltes Netzwerkmanagement. JFFNMS lässt sich in einer Master/Satellite Umgebung betreiben, durch die letztlich ein verteiltes Management möglich ist. Jeder Server kann als Master fungieren, der das Abfragen des eigenen Systems von außen nicht zulässt. Gleichzeitig können andere JFFNMS Systeme als Satelliten betrieben werden, die anderen Master Servern einen Zugriff auf die von ihnen gesammelten Daten ermöglichen. So können die Aufgaben im Netzwerk verteilt werden und gleichzeitig ein zentrales System geschaffen werden, welches den Überblick über das gesamte Netzwerk behält.

MRTG

Das von Tobias Oetiker entwickelte OpenSource Tool „Multi Router Traffic Grapher“ (MRTG) ist mit dem klaren Hintergrund der SNMP Netzwerküberwachung entwickelt worden. Zur Ermittlung der jeweiligen Werte kommt daher das SNMP Protokoll in den Versionen SNMPv1 und SNMPv2 zum Einsatz. MRTG ist in der populären Programmiersprache Perl entwickelt und besitzt seit Jahren einen stabilen Status, der einen Einsatz in einer Produk-

tivumgebung erlaubt. Zu finden ist das Werkzeug auf den Web-Seiten seines Entwicklers[19]. Die wichtigsten Merkmale sind im Folgenden beschrieben:

Übersichtliche graphische Darstellung. Das Hauptziel von MRTG besteht in der Generierung von Graphen zu den via SNMP ermittelten Messwerten. Die in regelmäßigen Abfragen mit einstellbarer Wiederkehr ermittelten Daten werden nicht in Tabellenform visualisiert, sondern in einen Graphen überführt. Auf diese Weise lässt sich die Flut an Informationen übersichtlich und verständlich präsentieren. Prinzipiell lassen sich alle Werte über einen beliebigen Zeitraum speichern, jedoch würde die Darstellung aller Werte gleichzeitig die Übersicht wieder verringern. Daher lassen sich auch verschiedene Teilausschnitte der Graphen anzeigen, welche die gesammelten Daten beispielsweise der letzten Stunde, des letzten Tages oder der letzten Woche beinhalten. Die Visualisierung erfolgt mittels der Erstellung eine Bildes im Portable Network Graphics (PNG) [76], welches sich ideal über das Internet darstellen lässt.

Verteilte Überwachung. Durch die Transformation der Informationen in das Internet-fähige Bildformat PNG lassen sich die Daten von beliebigen Orten aus abrufen. Dies ermöglicht auch eine ideale verteilte Überwachung, da beliebig viele MRTG Server im Netzwerk platziert werden können, deren Informationen dann von einer zentralen Stelle aus abgerufen und in eine einzelne Web-Applikation integriert werden können. Es muss lediglich auf jedem System, welches Daten ermittelt und für andere zur Verfügung stellt, ein Web-Server wie der APACHE installiert sein.

Universelle Überwachung. Durch den Einsatz des SNMP Protokolls ist eine universelle Überwachung verschiedenster Größen möglich. Alle Standard-MIBs, herstellerspezifischen MIBs oder auch benutzerdefinierten MIBs können prinzipiell unterstützt werden, da zur Überwachung lediglich eine OID anzugeben ist.

Perfekte Integration mit RRDTOOL. Das OpenSource Round-Robin Datenbankwerkzeug RRDTOOL, das ebenfalls von Tobias Oetiker stammt, lässt sich perfekt in MRTG integrieren. Auf diese Weise kann die Größe des von den gesammelten Werten belegten Speicherplatzes drastisch gesenkt werden. RRDTOOL verwendet dazu einen Algorithmus, der ältere Daten mit jeweils sinkender Auflösung speichert. Die dargestellten Diagramme verlieren dadurch nicht an Genauigkeit, da die Vielzahl an Informationen für einen langen Zeitraum nur mit geringer Genauigkeit dargestellt werden können.

Flexible Individualisierung. Die Konfiguration von MRTG ist äußerst flexibel, da sich beinahe alle möglichen Attribute für jeden Graphen einzeln spezifizieren lassen. Zu den Einstellungsmöglichkeiten gehört beispielsweise die Beschriftung der Graphen, der dargestellte Wertebereich, die Richtung der Graphenbildung oder das Abfrageinterval der zugehörigen SNMP Da-

[19] `poeple.ee.ethz.ch/~oetiker/webtools/mrtg/pub/`

tenquelle. Diese umfangreichen Individualisierungsmöglichkeiten erschweren gleichzeitig die Konfiguration von MRTG, da alle Angaben in einer einzigen Konfigurationsdateien hinterlegt werden. Zur Unterstützung des Administrators werden zwei hilfreiche Tools mitgeliefert. Das Tool CFGMAKER hilft bei der Erstellung der Konfigurationsdatei, indem lediglich einfache Kommandozeilenparameter für die jeweils automatisch durchgeführte Konfiguration aller Komponenten angegeben werden müssen. Mit dem zweiten Tool INDEXMAKER können gleichzeitig automatisch Web-Seiten erstellt werden, in welche die erzeugten Graphen eingebunden werden. So kann man mit wenigen Befehlen eine komplexe und automatische Konfiguration von MRTG durchführen, ohne die Flexibilität der Individualisierung insgesamt zu verlieren.

OpenNMS

Das bei SourceForge verwaltete OpenSource Tool OpenNMS konzentriert sich bei seinen Überwachungsaufgaben in erster Linie auf die über TCP erreichbaren Dienste. Eine Unterstützung zur Überwachung von SNMPv1 und SNMPv2 Objekten wurde aber ebenfalls integriert. Die Besonderheiten von OpenNMS werden im Folgenden zusammengefasst:

Abbildung einfachster Objekte auf ein simples Modell. Innerhalb von OpenNMS werden alle überwachten Objekte auf drei Basisobjekte abgebildet. Die Ausrichtung auf TCP Dienste sorgt für eine Abbildung aller einzelnen Objekte auf ein Element des Typs „Dienst“. Verschiedene Dienste können über dieselbe IP Adresse erreicht werden, so dass ein weiteres Objekt in einer höheren Ebene erforderlich ist. Diese Elemente sind vom Typ „Schnittstelle“ und durch ihre eindeutige IP Adresse definiert. In der dritten und letzten Ebene, die sich noch oberhalb der Schnittstellen befindet, liegen schließlich Elemente vom Typ „Knoten“. Unter den Knoten sind einzelne Netzwerkkomponenten zu verstehen, die auch mehrere Schnittstellen mit unterschiedlichen IP Adressen aufweisen können.

Automatische Entdeckung von Systemen. Eine Besonderheit von OpenNMS liegt in der automatischen Erkennung von Netzwerkknoten. Die Erkennung wird über das ICMP Protokoll abgehandelt und funktioniert in vielen Fällen zuverlässig. Einmal erkannten Geräten können dann weitere Dienste zugewiesen werden, die vom Werkzeug überwacht werden sollen. Alle Dienste, die auf Geräten zur Verfügung stehen, welche nicht über ICMP erreichbar sind, können manuell von den Administratoren hinzugefügt werden, damit sie ebenfalls überwacht werden können.

Offene Schnittstellen. Sowohl die graphische Oberfläche als auch die Konfigurationsdateien unterstützen offene Schnittstellen. Die über die Web-Oberfläche dargestellten Ergebnisse der Überwachung können von beliebigen Punkten im Netzwerk aus angesprochen werden. Gleichzeitig ist die Konfiguration des Tools in Form von Extensible Markup Language (XML) [77]

Dateien implementiert. Damit sind alle wichtigen Schnittstellen über offene Standards implementiert.

RRDTOOL

Ebenso wie der ‚Multi Router Traffic Grapher' (MRTG) ist auch das ‚Round-Robin Database Tool' (RRDTOOL) eine Entwicklung von Tobias Oetiker, die mit einem durchaus ausgereiften Status den Einsatz in einer Produktivumgebung erlaubt. Im engeren Sinne handelt es sich nicht um ein Werkzeug zur Überwachung oder Konfiguration von Netzwerken. Vielmehr handelt es sich um eine Gruppe von Hilfswerkzeugen, mit deren Hilfe sich zeitabhängige Messwerte effizient abspeichern und auch wieder darstellen lassen. Wie auch MRTG ist das Werkzeug auf den Web-Seiten des Entwicklers[20] zu finden. Die Besonderheiten werden im Folgenden zusammengefasst:

Unterstützung beliebiger Datenquellen. Da RRDTOOL nicht für die Datenerfassung zuständig ist, unterstützt es zumindest verschiedenste Möglichkeiten zur Entgegennahme der Daten. Eine grundsätzliche Variante besteht im Aufruf des Haupt-Tools RRDTOOL und direkter Übergabe der Messwerte über die Kommandozeile. Auf diese Weise lassen sich auf jeden Fall alle Daten an die Round-Robin Datenbank übergeben. Eine deutlich performantere Variante arbeitet über Unix Pipes. So lässt sich der Standard-Ausgabekanal (STDOUT) eines beliebigen Programms unmittelbar mit dem Standard-Eingabekanal (STDIN) von RRDTOOL verbinden und gleichzeitig ein perfekter Kommunikationsweg herstellen. Die dritte Variante arbeitet über das Netzwerk. Unter Unix Systemen kann das Werkzeug an einen beliebigen Port gebunden werden, über den anschließend dieselben Befehle übermittelt werden können. Somit lässt sich die Datenbank auch aus der Ferne mit den Messwerten füllen.

Begrenzter Speicherplatzverbrauch. RRDTOOL speichert die empfangenen Daten in Round-Robin Archiven, die man auch als Ringspeicher bezeichnen kann. Jeder dieser Ringspeicher muss vor Beginn der Archivierung initialisiert werden. Dazu kann jeder zuvor spezifizierten Datenquelle ein oder mehrere Archive zugewiesen werden. Da Datenquellen immer mit ihrem Messintervall gespeichert werden, kann die Größe aller Archive anschließend präzise vorgegeben werden. Dazu werden lediglich die Anzahl der zu speichernden Werte und eine optionale Zusammenfassung definiert. Werden nun im Betrieb Daten an das RRDTOOL gesendet, so werden diese je nach Konfiguration zunächst bearbeitet und anschließend derart in der Datenbank gespeichert, dass ältere Daten überschrieben werden[21]. Als Konsequenz aus diesem Vorgehen

[20] `people.ee.ethz.ch/~oetiker/webtools/rrdtool/`.

[21] Zu Beginn der Speicherung existieren selbstverständlich noch keine älteren Daten, die überschrieben werden könnten, da die Datenbank mit Null-Werten initialisiert wird

kann die Größe jedes einzelnen Archivs und damit auch der gesamten Datenbank bereits vor Beginn der Messungen exakt definiert werden. Unabhängig davon, wie viele Messwerte danach in die Datenbank gespeichert werden, bleibt der verbrauchte Speicherplatz konstant.

Fehlertolerante Datenspeicherung. Das Grundprinzip zur Darstellung des Verhaltens einer Datenquelle basiert auf der regelmäßigen Ermittlung von Messwerten. In der Praxis ist es allerdings schwierig, die Messwerte exakt zum richtigen Zeitpunkt zu ermitteln. Soll beispielsweise eine Größe im Minutentakt überwacht werden, so kann selbst bei einem Echtzeitbetriebssystem der genaue Zeitpunkt der Datenerfassung nicht gesteuert werden. In diesem Fall sind die einzelnen Daten mit einem Messfehler beim Zeitstempel versehen. Um dieses Problem grundsätzlich zu umgehen, können beim RRDTOOL die Daten zu jedem beliebigen Zeitpunkt aktualisiert werden. Wichtig ist nur, dass zu jedem Messwert auch die exakte Zeitangabe der Datenerfassung gespeichert wird. Zur Darstellung der Graphen ist das Tool anschließend in der Lage, einen Wert für den ursprünglich vorgesehenen Zeitpunkt der Datenmessung zu interpolieren.

Konsolidierung großer Datenmengen. Wird wie im obigen Beispiel eine Datenquelle im Minutentakt überwacht und deren aktueller Wert in die Datenbank geschrieben, dann sind nur mit erhöhtem Speicherplatzbedarf die Werte für einen längeren Zeitraum zu speichern. Häufig ist die feingranulare Speicherung der Messwerte aus länger vergangenen Zeiten jedoch nicht mehr von großer Bedeutung. Oftmals reicht es einem Administrator aus, wenn er die Messwerte aus der Vergangenheit in einer kompakteren Weise noch abrufen kann, also beispielsweise die Zusammenfassung der Werte aus Zeiträumen von jeweils fünf Minuten oder auch Stunden. Dadurch würden sich auch die Berechnungen zur Darstellung der Messwerte über einen längeren Zeitraum erheblich verkürzen lassen. Aus diesem Grund ist das RRDTOOL mit einer Funktion zur Datenkonsolidierung ausgestattet. Bereits zu Beginn beim Anlegen der Datenbank können Zeitpunkte definiert werden, zu denen eine Konsolidierung der Daten stattfindet. Als Beispiel dient wieder der im Minutentakt gemessene Wert, der für eine genaue Darstellung der letzten 24 Stunden herangezogen werden kann. Soll gleichzeitig ein Graph zur zeitlichen Entwicklung desselben Messwerts über die letzte Woche dargestellt werden, so würde es für eine ausreichende Genauigkeit bereits ausreichen, wenn man die Messwerte alle 10 Minuten ermittelt hätte. Deshalb kann die Datenbank auch so konfiguriert werden, dass alle Messwerte, die älter als 24 Stunden sind, zu einzelnen Werten für jeweils einen Zeitraum von zehn Minuten zusammengefasst werden. Für jeden bereits vergangenen Tag muss deshalb nur noch ein Zehntel des Speicherplatzes zur Verfügung gestellt werden. Der Vorgang der Konsolidierung kann mehrfach zu unterschiedlichen Zeiten erzwungen werden. Sollen beispielsweise auch Graphen für den letzten Monat und das letzte Jahr angezeigt werden, so sind für eine ausreichend genaue Darstellung noch ein-

mal erheblich weniger Daten notwendig. Eine zusätzliche Konsolidierung kann also noch einmal nach sieben Tagen oder nach 30 Tagen erfolgen.

Flexible Darstellung. Bei der Generierung der Graphen sind vielfältige Einstellungsmöglichkeiten zur Individualisierung der Diagramme möglich. Neben einfachen Änderungen wie Angeben einer Beschriftung der Achsen oder des gesamten Graphen sowie ein Skalieren der Achsen inklusive der Logarithmisierung sind noch viele andere Konfigurationen möglich. Zu diesem zählt beispielsweise das Ändern der Farbe zur Darstellung der Messwertkurve oder aber auch die Definition, ob die Messwerte als Linie oder als gefüllte Fläche dargestellt werden sollen. Außerdem ist es möglich, mehrere Messwertkurven im selben Graphen abzubilden sowie zusätzliche Linien und Kurven durch Angabe einer entsprechenden Gleichung in den Graphen einzublenden. Alle Messwerte können außerdem durch Angabe einer beliebigen Formel in Umgekehrt Polnischer Notation (RPN)[22] [80] vor der Ausgabe auf unbegrenzte Art und Weise angepasst werden, um auch komplexere Zusammenhänge zwischen den Messwerten und einer zu zeigenden Größe berücksichtigen zu können. RRDTOOL übernimmt somit weit mehr als nur die Aufgaben einer Datenbank zur intelligenten und effizienten Datenspeicherung, sondern es implementiert auch eine äußerst flexible Funktionalität zur Visualisierung der gespeicherten Daten.

Erkennung von ungewöhnlichem Verhalten. Ähnlich aufwändige mathematische Funktionen innerhalb von RRDTOOL wie die Interpolation und die Konsolidierung von Daten werden auch bei der Erkennung einer ungewöhnlichen Verhaltensweise eines Systems eingesetzt. Durch die in der Vergangenheit gemessenen Werte ist das Werkzeug in der Lage, eine Vorhersage über den nächsten Messwert in der Zukunft zu liefern. Liegt der tatsächliche Wert innerhalb einer kurzen Zeitspanne mehrmals außerhalb des berechneten Wertebereiches, so interpretiert das Tool dies als ein ungewöhnliches Verhalten und protokolliert es entsprechend. Auf diese Weise ist es auch möglich, eine automatische Benachrichtigung der Administratoren einzurichten.

SCOTTY

Bei SCOTTY handelt es sich um eine Erweiterung der Tool Command Language (TCL[23]) um die Fähigkeit zur Durchführung von Aufgaben des Netzwerkmanagements. Im engeren Sinne ist SCOTTY nicht als ein Netzwerkmanagementwerkzeug, sondern vielmehr als ein Paar von zwei Entwicklungs-Tools aus dem Bereich des Netzwerkmanagements zu verstehen. Das erste Tool ist die Tcl Network Management (TNM) Erweiterung, die eine einfache Schnittstelle

[22] „Reverse Polish Notation"

[23] Die Programmiersprache TCL und deren graphische Schnittstelle TK – genannt TCL/TK – sind wie SCOTTY selbst OpenSource Software und werden bei SourceForge verwaltet.

zu den einzelnen Diensten und Protokollen im Netzwerk bildet. Das zweite Tool ist der TK Interactive Network Editor (TKINED), der auf dem Werkzeug Tk für die graphische Oberfläche zu TCL basiert. SCOTTY hat einen stabilen Status erreicht und kann daher auch in einer Produktivumgebung zur Netzwerküberwachung verwendet werden. Die besonderen Merkmale werden im Folgenden zusammengefasst:

Plattformunabhängigkeit. Durch die Tatsache, dass es sich bei SCOTTY um ein OpenSource Tool handelt, das zugleich in der OpenSource Programmiersprache TCL/TK entwickelt wird, macht es zu einem plattformunabhängigen Werkzeug, das auf vielen Betriebssystemen installiert werden kann. Da SCOTTY eher als Erweiterung dieser Programmiersprache zu sehen ist, wird es daher auch von ebenso vielen Plattformen unterstützt.

Einfache Schnittstelle zu verschiedensten Netzwerktechnologien. Das Ziel von TNM besteht darin, für die verschiedensten Dienste und Protokolle eine einheitliche und vereinfachte Schnittstelle zu liefern. Zu den unterstützten Technologien zählen beispielsweise die Protokolle ICMP, TCP und UDP, sowie die unterschiedlichen Dienste aus den Bereichen DNS, HTTP, SNMP, SYSLOG oder dem Network Time Protocol (NTP). Zusätzlich können SNMP MIBs und lokale Datenbanken ausgelesen sowie die Ergebnisse von anderen Tcl Programmen ausgewertet werden. Alle diese verschiedenen Objekte können in TNM über eine Schnittstelle angesprochen werden, die einfachste Befehle über eine Textkonsolen-ähnliche Verbindung entgegennimmt. Diese Voraussetzungen erlauben eine einfache Überwachung der verschiedensten Netzwerkobjekte.

Individuelle Visualisierung. Das zweite Werkzeug TKINED dient zur Erstellung der graphischen Oberfläche, in der sich SCOTTY dem Administrator präsentieren soll. Da eine automatische Erkennung von Komponenten im Netzwerk nicht unterstützt wird, muss der Administrator die Objekte einzeln zur Oberfläche hinzufügen. Im gleichen Schritt wird auch spezifiziert, welche Überwachung für das jeweilige Objekt durchgeführt werden soll. Zur Auswahl stehen alle durch TNM unterstützten Objekte wie beispielsweise eine Überprüfung der Erreichbarkeit mittels ICMP, eine Überwachung von SNMP Objekten oder eine Analyse von eingehenden SYSLOG Meldungen. Dem Administrator steht es dabei offen, jedes Objekt in der graphischen Oberfläche individuell zu formatieren. Es steht bereits eine Auswahl an verschiedenen Icons zur Verfügung, die aber jederzeit durch Eigenkreationen ergänzt werden kann. Auch die Farben der Icons oder das Hintergrundbild können individualisiert werden. Die Zusammenfassung von mehreren Objekten zu einzelnen Gruppen erlaubt überdies eine Strukturierung der Oberfläche. Die direkt bei den Icons für die einzelnen Objekte angezeigten Alarme werden bei der Gruppierung auf das Icon für die Gruppe übertragen, so dass keine wichtigen Informationen verloren gehen.

11.1.4 Individuallösungen

Da das Netzwerkmanagement im Wesentlichen auf offenen Standards aufbaut und da gleichzeitig eine große Anzahl von teilweise gut ausgereifter OpenSource Software entstanden ist, sind damit auch ideale Grundvoraussetzungen für die Entwicklung von Individuallösungen geschaffen worden. Jeder Administrator muss sich vor Aufbau eines Netzwerkmanagementsystems Gedanken über die Realisierung der gewünschten Funktionalitäten machen. Dabei sollten bereits in dieser Phase die Anforderungen an die Sicherheit des Netzwerkmanagements spezifiziert werden. Diese haben im weiteren Verlauf weitreichende Auswirkungen auf die Wahl der einzusetzenden Software-Produkte. Sowohl bei den kommerziellen als auch bei den quelloffenen Tools finden sich Produkte mit unterschiedlicher Implementierung der möglichen Sicherheitsfunktionalitäten des Netzwerkmanagements. Ziel an dieser Stelle kann es nur sein, die Wichtigkeit der Sicherheit des Netzwerkmanagements noch einmal zu betonen. Sollte es einem Angreifer gelingen, die Kontrolle über das Managementsystem zu erlangen, so erhält er damit gleichzeitig die Kontrolle über das zentrale „Nervensystem“ des Unternehmens oder der Institution. Bei der Wahl der verwendeten Produkte ist kein eindeutiger Trend in Bezug auf die Sicherheit zu erkennen. Weder kommerzielle noch quelloffene Systeme stechen durch besondere Berücksichtigung oder Missachtung moderner Sicherheitsmechanismen hervor. Hier gilt es, die eigenen Bedürfnisse objektiv einzuordnen und individuell zu entscheiden.

Die in diesem Buch gemachten Aussagen über Netzwerkmanagementsysteme können weder ein vollständiges Bild über die jeweiligen Werkzeuge noch über das Marktsegment der Netzwerkmanagement Tools im Allgemeinen abliefern. Vor dem Einsatz eines speziellen Werkzeugs sollten unbedingt weitere Informationen von den Entwicklern und anderen Anwendern des Produktes eingeholt werden. Bei kommerziellen Produkten ist allerdings darauf zu achten, dass die Beschreibungen oftmals die positiven Aspekte hervorheben, während die Nachteile häufig vernachlässigt werden. Bei quelloffenen Tools sieht das vielfach anders aus. Hier werden häufig die offenen Probleme beim Namen genannt und oft sogar eigene Sektionen eingerichtet, die sich ausschließlich den Mängeln widmen. Zur gleichen Zeit sind die Vorzüge der OpenSource Werkzeuge nicht auf eine Maximierung des Verkaufs ausgelegt, da sie in den meisten Fällen sogar kostenlos zu beziehen sind. Wurde die Beschreibung des Tools jedoch von einem Unternehmen erstellt, dass mit der Beratung oder Dienstleistung rund um das jeweilige Werkzeug einen Gewinn machen will, so ist auch hier mit Vorsicht zu verfahren. Das Risiko einer Fehlinvestition ist jedoch bei kostenfreien Tools grundsätzlich am Geringsten.

Wer all diesen Schwierigkeiten und Problemen aus dem Wege gehen möchte, der kann sich zur Entwicklung einer vollständig eigenen Lösung entschließen. Die Integration von OpenSource Tools zur Abdeckung einzelner Aufgaben ist damit nicht grundsätzlich ausgeschlossen. Die klaren Vorteile einer Individuallösung liegen bei der perfekten Anpassung an die eigenen Anfor-

derungen. Eine Individuallösung kann immer zielgerichtet zur Lösung der eigenen Probleme weiterentwickelt werden. Ein klarer Nachteil ist jedoch der hohe Zeitaufwand bei der Entwicklung. Je nachdem, von welcher Basis aus die eigene Entwicklung gestartet wird, sind unterschiedlich umfangreiche Arbeiten bis zum Einsatz der Lösung erforderlich. Auch an den Kenntnisstand des Personals werden bei der Entwicklung einer Individuallösung ganz andere Anforderungen gestellt. Erfordern kommerzielle Produkte eher Fähigkeiten in der Bedienung der Software, so sind bei Individuallösungen auch erweiterte Kenntnisse bei den verwendeten Programmiersprachen unverzichtbar. All diese Faktoren sind bei der Wahl eines Netzwerkmanagementsystems zu bewerten und zu berücksichtigen.

Als ein Beispiel für eine Entwicklungsumgebung zur Erstellung von individuellen Netzwerkmanagementlösungen soll an dieser Stelle die quelloffene Werkzeugfamilie NET-SNMP [138] dienen. Diese umfasst neben einem vollständigen und beliebig erweiterbaren SNMP Agenten auch eine ganze Reihe von Entwicklungshilfen und Modulen in der Programmiersprache Perl, mit denen sich vergleichsweise einfach komplett eigene Netzwerkmanagement Applikationen entwickeln lassen. Die Beschränkung auf NET-SNMP ist willkürlich gewählt und es finden sich im Internet noch zahlreiche andere Tools, welche die Entwicklung einer Individuallösung unterstützen.

NET-SNMP

Das NET-SNMP Projekt ist aus dem UCD-SNMP Projekt der ‚University of California, Davis'[24] (UCD) hervorgegangen und basiert auf den Programmiersprachen C [105] und Perl. Das Tool hat durchaus einen stabilen Status erreicht, der einen Einsatz in einer Produktivumgebung erlaubt, und wird auf einer eigenen Web-Seite[25] verwaltet. Eine außergewöhnliche Besonderheit des freien Tools ist die Unterstützung beinahe aller in den SNMPv1, SNMPv2 und SNMPv3 Standards definierten Mechanismen. Sogar das Protokoll IPv6 ist in der Werkzeugfamilie implementiert. Einzigartig ist die Unterstützung der experimentellen Authentifizierungsmethode auf Basis von Kerberos Version 5 (Kerberos5) [139]. Die Werkzeugfamilie ist mittlerweile sehr umfangreich und setzt sich aus fünf Hauptkategorien zusammen, die im Folgenden näher beschrieben werden:

Erweiterbarer SNMP Agent. Der bei NET-SNMP enthaltene SNMP Agent kann durch die Offenlegung des Quelltextes leicht um eigene, in der Programmiersprache C entwickelte MIBs erweitert werden. Die Werkzeugfamilie unterstützt den Entwickler außerdem bei der Erstellung eigener MIB Module. Das enthaltene Kommandozeilenwerkzeug MIB2C erstellt aus einer beliebigen MIB eine Vorlage für eine entsprechende C Implementierung. Es sind „lediglich" die einzelnen Funktionen zu implementieren, die im Agenten

[24] www.ucdavis.edu

[25] www.net-snmp.org

den OIDs ihre Inhalte zuweisen. Zusätzlich bietet der erweiterbare SNMP Agent mit den PASS und PASS_PERSIST Mechanismen einfache Möglichkeiten zur Einbindung beliebiger Scripte, welche benutzerdefinierte MIBs implementieren. Auf diese Weise kann der SNMP Agent um eigene MIBs, die in einer nahezu beliebigen Programmiersprache implementiert sein können, erweitert werden.

SNMP Nachrichten Empfänger. Mit dem SNMP Nachrichten-Empfänger SNMPTRAPD können Alarme und Nachrichten des Protokolls SNMPv1 empfangen und archiviert werden. Der NET-SNMP Nachrichtenempfänger kann bei Eintreffen einer Meldung verschiedene Aktionen auslösen wie beispielsweise das Weiterleiten der Nachrichten an Systemprotokolle auf Basis von SYSLOG oder dem Microsoft Ereignisprotokoll sowie eine Archivierung in einer lokalen Datei. Die eingehenden SNMP Nachrichten können außerdem die Ausführung eines beliebigen Programms initiieren. Schließlich kann der Nachrichtenempfänger auch als Proxy konfiguriert werden und die erhaltenen Meldungen an einen anderen SNMP Nachrichtenempfänger weiterleiten.

Kommandozeilenwerkzeuge. Zu den wichtigsten Komponenten von NET-SNMP zählen vermutlich die universellen Kommandozeilenwerkzeuge, mit denen sich fast jede Aufgabe aus dem Bereich SNMP erledigen lässt. Zu den bekanntesten und hilfreichsten zählen die Befehle zum Auslesen oder Setzen von einzelnen OIDs oder ganzen OID-Unterbäumen. Mit SNMPGET kann direkt der Inhalt der spezifizierten OID ausgelesen werden. Unterstützt werden alle drei SNMP Versionen. Vor allem bei Tabellen eignet sich der SNMPGETNEXT Befehl, mit dessen Hilfe der Wert einer nachfolgenden OID ausgelesen werden kann. Der Befehl SNMPBULKGET ermöglicht das Auslesen von mehreren OIDs zur gleichen Zeit. Mit dem SNMPWALK Befehl kann das Auslesen eines vollständigen MIB-Unterbaumes gestartet werden, und mit dem Befehl SNMPBULKWALK können die dabei übermittelten Werte noch zu größeren Datenpaketen zusammengefasst werden. Auch der SNMPTABLE Befehl besitzt klare Vorzüge: mit ihm lassen sich ganze SNMP Tabellen auslesen und anschließend in Tabellenform visualisieren. Schließlich existiert mit SNMPSET noch ein Befehl zum Schreiben von OIDs. Neben diesen Befehlen beinhaltet die NET-SNMP Werkzeugfamilie noch eine ganze Reihe anderer hilfreicher Kommandos. Beispielsweise können mit den beiden Befehlen SNMPTRAP und SNMPINFORM SNMP Nachrichten der Versionen SNMPv1 oder SNMPv2 versendet werden oder mit dem MIB2C Tool aus einer MIB eine Vorlage zur Erstellung einer Erweiterung des Agenten in der Programmiersprache C erstellt werden.

Graphischer MIB Betrachter. Der enthaltene graphische MIB Betrachter TKMIB ist auf Basis des Toolkit (Tk) entwickelt worden. Mit diesem Werkzeug lassen sich beliebige MIB Spezifikationen visualisieren.

Bibliothek zur Entwicklung eigener C oder Perl SNMP Applikationen. Zur Entwicklung von Individuallösungen stehen schließlich die C und

Perl Bibliotheken zur Verfügung. Mehr als 40 verschiedene Application Program Interface (API) Definitionen stehen dem Entwickler zur Verfügung, mit denen sich sogar ein eigener Agent entwickeln ließe. Häufig werden diese aber nur zur Implementierung von benutzerdefinierten MIBs eingesetzt.

11.2 IPMI Werkzeuge

Eine ganze Reihe von Herstellern implementiert die IPMI Spezifikation entweder in Form von Hardware oder in Form von Software. Eine ausführliche Liste aller aktuellen Implementierer der IPMI Spezifikation findet sich in Anhang B. Unter den Werkzeugen zur Überwachung von IPMI-fähigen Systemen hebt sich ein kommerzieller Software-Hersteller hervor, da er für viele andere Produkte die Basis-Software liefert. Gemeint ist das Unternehmen OSA Technologies [149], welches den IPMI Standard weitestgehend unterstützt. Der wichtigste Lizenznehmer und Integrator der OSA Technologie ist vermutlich Avocent International Ltd., die mit ihrer DSI5100 Hardware und dem dazu passenden DSVIEW 3 Software-Produkt eine umfangreiche Lösung für das Systemmanagement via IPMI anbieten. Daneben sind einige der Managementfunktionalitäten von IPMI in größere Managementsysteme integriert. Einige der weiter oben bei den herstellereigenen SNMP Werkzeugen aufgeführten Produkte setzen auch Teile der IPMI Spezifikation um. Schließlich findet sich auch eine OpenSource Gemeinde, die sich für eine Implementierung der IPMI Spezifikationen in den Linux Kernel oder auch in andere Applikationen einsetzt.

11.2.1 Werkzeuge der IPMI Entwickler

Die vier Entwickler der IPMI Spezifikation – Intel, Hewlett-Packard, NEC und Dell – liefern selbstverständlich eigene Werkzeuge zur Überwachung und Verwaltung von Geräten über den IPMI Standard. Zwar fällt diese Art von Werkzeug nicht direkt in die Kategorie des Netzwerkmanagements, jedoch ist auch das Systemmanagement eng mit der Netzwerkverwaltung verbunden, wenn es sich um die Überwachung von Schlüsselkomponenten im Netzwerk handelt. Ohne eine Bewertung vornehmen zu wollen, sollen an dieser Stelle die Werkzeuge von zwei der vier Hersteller näher beschrieben werden.

Intel: SERVER MANAGEMENT 5

Als einer der Hauptentwickler der IPMI Spezifikation hat selbstverständlich auch Intel[26] eine eigene Management-Software zur Überwachung und Konfiguration von IPMI-fähigen Geräten. Das Tool SERVER MANAGEMENT 5 ist speziell für Intel Plattformen entwickelt und zielt daher auf die IA32 und

[26] www.intel.com

IA64 Server Architektur. Es wird ein breites Spektrum an Funktionalitäten aus der IPMI Spezifikation unterstützt. Daher ist das Tool auch in fünf verschiedene einzelne Applikationen aufgeteilt, die jeweils unterschiedliche Aufgaben erfüllen. Ziel bei der Entwicklung war es, sowohl eine Architektur mit einer einheitlichen Benutzeroberfläche für alle Teilwerkzeuge zu schaffen und gleichzeitig die Tools so unabhängig wie möglich zu entwerfen, so dass sie sich problemlos in andere Managementsysteme integrieren lassen. Die wichtigsten Funktionalitäten der einzelnen Tools sind im Folgenden aufgeführt:

Vereinheitlichte Administration. Mit dem SERVER MANAGEMENT 5 Werkzeug können verschiedenste Managementaufgaben über eine einheitliche Schnittstelle durchgeführt werden. Die meisten Tools sind dabei in einem Rahmenwerk zusammengefasst, das auf einem Microsoft Betriebssystem aufbaut. Für jedes Werkzeug wurde versucht, die graphische Oberfläche möglichst gleich zu gestalten. Der Zugriff auf die zu verwaltenden Geräte kann dabei auf drei verschiedene Möglichkeiten erfolgen. Neben der Administration über eine direkte serielle Verbindung können auch eine Überwachung und Konfiguration der Systeme aus größerer Entfernung durchgeführt werden. Dazu unterstützt die Management-Software Verbindungen über ein Modem oder über die LAN Schnittstelle.

PLATFORM INSTRUMENTATION CONTROL. Das Tool PLATFORM INSTRUMENTATION CONTROL liefert dem Systemadministrator eine graphische Oberfläche zur Überwachung der einzelnen Sensoren in einem Intel IPMI Gerät. Das Tool ist für eine Microsoft Architektur entwickelt und liefert eine an dessen Betriebssysteme erinnernde Oberfläche. Zu den Hauptaufgaben des Tools zählt die Visualisierung der Zustände aller IPMI Sensoren des Systems. Zusätzlich können mehrere Schwellwerte zu den einzelnen Sensorgeräten konfiguriert werden, die jeweils den Übergang zwischen den verschiedenen Statuszuständen markieren. Bei Änderung des Systemzustandes können darauf Alarme und Aktionen generiert werden, die sich über das Tool ebenfalls konfigurieren lassen. Der Zugriff erfolgt ausschließlich über die LAN Schnittstelle. PLATFORM INSTRUMENTATION CONTROL ist leicht in mehrere Managementsysteme integrierbar wie beispielsweise in HPs OPENVIEW oder CAs UNICENTER.

DIRECT PLATFORM CONTROL. Das DIRECT PLATFORM CONTROL Werkzeug ist weniger auf die Überwachung als auf die Konfiguration der IPMI Geräte ausgelegt. Mit dem Tool können die verwalteten Systeme unabhängig von ihrem Einschaltzustand über eine Textkonsole administriert werden. Wie das PLATFORM INSTRUMENTATION CONTROL Tool unterstützt auch das DIRECT PLATFORM CONTROL Tool eine Verbindung über die LAN Schnittstelle. Zusätzlich kann die Kommunikation aber auch über eine direkte serielle Verbindung oder über ein Modem erfolgen. Die IPMI Spezifikation erlaubt es sogar, ein ausgeschaltetes Gerät über den ständig mit Strom versorgten BMC zu administrieren. Zur Verfügung stehen neben Befehlen zum Einschalten, Aus-

schalten und Neuinitialisieren des Systems vor allem auch Befehle, mit denen verschiedenste Informationen aus dem BMC ausgelesen werden können. Dies beinhaltet die Einträge aus den System Event Log (SEL) und den Sensor Data Record (SDR) Datenbanken genauso wie allgemeine Informationen über die angeschlossenen FRUs und Sensoren. Außerdem kann die Adressdatenbank mit den Zielen für die Versendung von Alarmen administriert werden.

Client System Setup Utility. Auch das Client System Setup Utility ist ein auf Microsoft basierendes Tool, welches der Fernadministration von IPMI-fähigen Systemen dient. Zusätzlich zu den Funktionen, die schon das Direct Platform Control Werkzeug liefert, können vor allem Einstellungen für den Bootvorgang des Systems oder aber auch Passwörter konfiguriert werden. Auch eine Funktion zur Aktualisierung des BIOS oder der BMC Firmware wird von dem graphischen Werkzeug unterstützt. Schließlich können mit dem Tool auch die Konfigurationseinstellungen des BMC auf der Managementstation gesichert oder von dort wiederhergestellt werden.

LAN Alert Viewer. Obwohl der LAN Alert Viewer in der offenen Programmiersprache Java [101] implementiert ist, ist er dennoch fest mit der Microsoft Architektur verbunden. Seine Aufgabe besteht im Wesentlichen in der Entgegennahme von IPMI Alarmen, welche die Management-Software via SNMP Nachrichten vom BMC erhält, und der Visualisierung dieser Alarme. In der werkzeugeigenen Datenbank können die eingegangenen Alarme betrachtet, bestätigt oder gelöscht werden. Mögliche Alarmquellen sind das Überschreiten oder das Unterschreiten eines Schwellwertes, die Erkennung eines Einbruchs in das Gehäuse, ein Fehler in der Spannungsversorgung, eine Fehlermeldung des BIOS oder auch der Ausfall eines Kühlelementes.

Kommandozeile via Serial Over LAN. Mit der Serial Over LAN (SOL) Funktionalität kann die Kommandozeile des Betriebssystems des entfernten Gerätes an eine Managementstation weitergeleitet werden. Zwar sind die einzelnen Software-Komponenten des Intel Server Management 5 auf Microsoft Systeme beschränkt, die überwachten und verwalteten Geräte können jedoch nahezu jedes beliebige Betriebssystem verwenden. Vor allem die vollständige Unabhängigkeit der IPMI Spezifikation vom Betriebssystem der jeweiligen Geräte erlaubt eine derartige Flexibilität. Serial Over LAN kann allerdings nur dann eine Kommandozeile weiterleiten, wenn das installierte Betriebssystem auch über eine administrative Textkonsole verfügt.

Hewlett-Packard: Integrated Lights-Out

Hewlett-Packard[27] zählt ebenfalls zu den vier Entwicklern der IPMI Spezifikation. Mit dem Integrated Lights-Out wird dabei eine enge Verknüpfung zwischen Hardware und Software realisiert. Große Teile der Funktionalität befinden sich direkt in Hardware-Komponenten der HP ProLiant Server-

[27] www.hp.com

Systeme. Zusätzlich zur Firmware des IPMI Controllers können mit der optionalen Software INTEGRATED LIGHTS-OUT ADVANCED diese Funktionalitäten noch einmal erweitert werden. Die Hauptmerkmale des Managementsystems sind im Folgenden aufgeführt:

Betriebssystemunabhängige Konsole. In der INTEGRATED LIGHTS-OUT IPMI Firmware ist bereits die Implementierung einer Textkonsole enthalten, die vollständig unabhängig vom Betriebssystem arbeitet. Über diese Konsole können vorrangig Informationen zu Statuszuständen des Systems abgerufen werden. Die optionale Software INTEGRATED LIGHTS-OUT ADVANCED liefert zusätzlich noch eine graphische Oberfläche mit Mausunterstützung, mit deren Hilfe sich im Wesentlichen dieselben Funktionen ausführen lassen.

Steuerung und Überwachung der Stromzufuhr. Mit INTEGRATED LIGHTS-OUT können nicht nur die Systeme nach Belieben eingeschaltet oder wieder ausgeschaltet werden, auch ein Leistungsmanagement lässt sich bequem aus der Ferne konfigurieren. Somit können zu Zeiten geringerer Auslastung der Stromverbrauch und die notwendige Kühlleistung gesenkt werden. Ein separater Sensor überwacht gleichzeitig die Stromzufuhr und kann über Unregelmäßigkeiten informieren.

In-Band Management über das Netzwerk. Die Kommunikation mit dem INTEGRATED LIGHTS-OUT BMC kann flexibel über das Netzwerk erfolgen. Die eingebaute Netzwerkschnittstelle des Systems kann dazu entweder mitbenutzt werden oder aber exklusiv für IPMI-Verkehr reserviert werden. Neben der Möglichkeit zur Konfiguration einer statischen IP Adresse kann auch auf einen im Netzwerk befindlichen DHCP Server zurückgegriffen werden. Gleichzeitig sorgt die Onboard-Software dafür, dass die überwachte Komponente in den entsprechenden DNS Nameservern und den Windows Internet Naming Service (WINS) Servern eingetragen wird.

Flexible Zugangsmöglichkeiten. Das IPMI Managementwerkzeug INTEGRATED LIGHTS-OUT unterstützt verschiedene Kommunikationswege, über welche das System angesprochen werden kann. Neben der Möglichkeit zum Abrufen von Informationen über eine Web-Schnittstelle kann der Zugang auch über das Netzwerk, direkt über die Serielle Schnittstelle oder per Einwahl über ein Modem an der Seriellen Schnittstelle erfolgen. Zusätzlich kann über IPMI auch die serielle Verbindung zum Betriebssystem über das Netzwerk weitergeleitet werden.

Security. INTEGRATED LIGHTS-OUT verfügt über eine ausgeprägte Unterstützung verschiedenster Sicherheitsmechanismen. Dazu zählt unter anderem auch die Unterscheidung von bis zu zwölf verschiedenen Benutzern, deren Berechtigungen unabhängig voneinander konfiguriert werden können. Zur Vereinfachung der Administration können die Benutzer mittels INTEGRATED LIGHTS-OUT ADVANCED auch extern in entsprechenden Verzeichnis-Servern verwaltet werden. Dadurch ist auch eine Zertifikat-basierte Authentifizierung

der Benutzer möglich. Vor allem aber zeichnet sich das Werkzeug aus dem Hause HP durch die Unterstützung von mehreren Sicherheitsmechanismen bei der Übertragung von Informationen über das Netzwerk aus. Die über die Web-Schnittstelle präsentierten Informationen werden verschlüsselt über das HTTPS Protokoll übertragen und die Verbindung zur Textkonsole kann mittels SSH erfolgen. Außerdem implementiert das Werkzeug auch eine VPN-Lösung, mit der sich die über das Netzwerk übertragenen Daten ebenfalls verschlüsseln lassen.

11.2.2 Kommerzielle Werkzeuge

Wie schon bei den SNMP Werkzeugen findet sich auch beim Systemmanagement über IPMI eine große Anzahl von Herstellern auf dem Markt, die mit einer Lösung aufwarten können. In Anhang B findet sich eine überaus umfangreiche Liste aller Hersteller, die zum Zeitpunkt der Erstellung dieses Buches den IPMI Standard in irgendeiner Weise unterstützen. Viele von ihnen liefern auch eine eigene Software-Lösung zur Administration von Systemen über die IPMI Schnittstelle an. Es ist unmöglich, an dieser Stelle alle Tools vorzustellen, daher sollen wertungsfrei nur einige zufällig ausgewählte Werkzeuge näher beschrieben werden.

AMI: Unified Management Server

Als bekannter BIOS Hersteller implementiert das Unternehmen American Megatrends Inc. (AMI[28]) vor allem die Hardware-seitige Unterstützung der IPMI Spezifikation. Mit dem Produkt Unified Management Server wird allerdings auch eine Software-Lösung zur optimalen Ausnutzung der Hardware-Möglichkeiten bereitgestellt. Die Unterstützung fokussiert sich zunächst auf eigene Produkte, ist aber auch auf anderen Geräten anwendbar. Es wird eine große Bandbreite der Funktionalitäten von IPMI unterstützt. Ergänzt wurde im Wesentlichen nur eine graphische Oberfläche mit einer Web-Schnittstelle. Die herausragendsten Besonderheiten werden im Folgenden aufgelistet:

Plattformunabhängigkeit. Durch die Verwendung einer Web-Oberfläche und Implementierung der Funktionalitäten mittels der Java 2 Platform, Enterprise Edition (J2EE) [102] entkoppelt sich der Unified Management Server vollständig vom Betriebssystem. Die Management-Software kann somit auf beliebigen Systemen installiert werden.

Fernadministration. Die Implementierung des Unified Management Server auf Basis von Web-Technologien macht das Managementsystem nicht nur unabhängig vom Betriebssystem, sondern erlaubt auch die Administration der Hardware-Komponenten von einem beliebigen Ort. Dabei können nicht

[28] www.ami.com

nur die Systemstatuszustände der überwachten Komponenten angezeigt, sondern auch wichtige administrative Aufgaben durchgeführt werden. Zu diesen Aufgaben zählt beispielsweise das Starten, Stoppen oder Neuinitialisieren von Systemen. Die Unterstützung des Serial Over LAN (SOL) Mechanismus erlaubt es außerdem, die Textkonsole des verwalteten Systems an jeden beliebigen Ort weiterzuleiten und eine Administration der Komponente mit vorhandenen Mitteln durchzuführen.

Fehlertoleranz. Durch die IPMI Spezifikation bedingt kann die Administration der einzelnen Systeme auch noch bei Eintreten eines Fehlers durchgeführt werden. Selbst bei Ausfall des Hauptprozessors oder bei Absturz des Betriebssystems der verwalteten Komponente kann über IPMI das System noch angesprochen werden und der genaue Fehler identifiziert werden. Bei Einsatz des UNIFIED MANAGEMENT SERVER in einer Out-of-Band Managementumgebung gilt die Fehlertoleranz sogar dann noch, wenn das Hauptnetzwerk auf dem Weg zur Komponente ausfällt. Durch das parallele Managementnetzwerk kann das System weiterhin angesprochen und administriert werden. Auf diese Weise können auch Netzwerkkomponenten mit dem Ziel der Fehlerbeseitigung verwaltet werden, die für eine Störung im Netzwerk verantwortlich sind.

Komfortable Benachrichtigung. Ein IPMI-fähiges Gerät ist in der Lage, auf verschiedenste Art einen Fehler zu melden. Im Normalfall werden bei den durch die einzelnen Sensoren erkannten Fehlern entsprechende Nachrichten in die IPMI System Event Log (SEL) Datenbank geschrieben. Der Platform Event Filtering (PEF) Mechanismus kann zusätzlich derart konfiguriert werden, dass er bei Erkennen eines kritischen Ereignisses nicht nur eine Aktion wie das Ausschalten des jeweiligen Systems durchführt, sondern dass er einen Alarm an ein beliebiges Zielsystem sendet. Durch die Unterstützung des SNMP Standards ist der UNIFIED MANAGEMENT SERVER in der Lage, als Ziel für die Alarme zu dienen sowie die SNMP Nachrichten des BMC zu empfangen und auszuwerten. Die Management-Software erlaubt es anschließend auch, eine Benachrichtigung der Administratoren via E-Mail durchzuführen.

Automatische Erkennung. Der UNIFIED MANAGEMENT SERVER unterstützt eine automatische Erkennung von Systemen im Netzwerk über IPMI, SNMP oder das Common Information Model (CIM). Auf diese Weise lassen sich die Komponenten bequem zur Fernwartung hinzufügen.

Sicherheit. Der über eine Web-Schnittstelle implementierte Zugriff auf das Managementwerkzeug lässt sich benutzerabhängig einschränken. Durch die Unterscheidung der einzelnen Benutzer ist auch eine Individualisierung der graphischen Oberfläche möglich. Zusätzlich werden die Sicherheitsmechanismen von IMPI und RMCP+ voll unterstützt. Auf diese Weise ist auch eine erweiterte Authentifizierung und Verschlüsselung der übertragenen Daten möglich.

Amphus: ManageSite

Das Unternehmen Amphus Inc.[29] gehört wie auch AMI zu den offiziellen IPMI Implementierern. Im Gegensatz zum Hardware-Hersteller zielt Amphus aber explizit auf die Unterstützung von Hardware-Komponenten beliebiger Hersteller. Das Produkt ManageSite ermöglicht den Administratoren eine umfassende Überwachung und Konfiguration der einzelnen Komponenten selbst in großen und heterogenen Netzwerken. Neben einer breiten Unterstützung der IPMI Spezifikation findet sich im Tool vor allem eine Web-Schnittstelle zur Administration. Alle wichtigen Funktionalitäten werden im Folgenden aufgeführt:

Betriebssystemunabhängige Funktionalität. Der Einsatz der IPMI Spezifikation erlaubt nicht nur eine Überwachung von Komponenten, die unterschiedlichste Betriebssysteme aufweisen. Auch eine vollständige Administration von „Bare-Metal“[30] Systemen ist mit IPMI problemlos möglich. ManageSite kann daher jede beliebige IPMI-fähige Hardware administrieren.

Kritikalitätsabhängige Alarmbehandlung. Zur Entlastung der Administratoren ist ManageSite mit einer intelligenten Alarmfilterung ausgestattet. Die Kritikalität der eingehenden Alarme kann vom System automatisch erkannt werden und abhängig von den eingestellten Regeln werden nur die jeweils zuständigen Administratoren benachrichtigt. Auf diese Weise können die Administratoren gezielt auf die wichtigeren Probleme aufmerksam gemacht werden.

Zeitabhängig Ausführung von Aufgaben. ManageSite enthält als wichtigen Bestandteil einen Aufgabenplaner, über den sich beliebige komplexe Managementaufgaben zeitabhängig konfigurieren und ausführen lassen. Rechenintensive Administrationsaufgaben können so leicht zu unkritischen Zeitpunkten gestartet werden. Das Managementsystem unterstützt außerdem eine nahezu endlose Anzahl von verwalteten Geräten, so dass sich auch komponentenübergreifende Aufgaben, die eine große Anzahl von Systemen betreffen, automatisieren lassen.

Flexible Software-Verteilung. Da mittels der IPMI Spezifikation einzelne Systeme vollständig unabhängig vom installierten Betriebssystem konfiguriert werden können, ist teilweise eine Verteilung von einzelnen Applikationen oder besser kompletten Images[31] auf allen Komponenten möglich. Über ManageSite können beliebige Rechner – unabhängig davon, ob sie bereits über ein

[29] `www.amphus.com`

[30] „Bare-Metal“ bedeutet soviel wie „reines Metall“ und bezeichnet ein von jeglicher Software freies System. Ein gutes Beispiel dafür ist ein fabrikneuer Rechner ohne vorinstalliertes Betriebssystem.

[31] Das „Image“ ist hier als „Spiegelbild“ eines kompletten Systems inklusive Betriebssystem und Applikationen zu verstehen, das sich auf verschiedene Rechner aufbringen lässt.

Betriebssystem verfügen – auf einfache Weise mit einem zuvor erstellten Image versehen werden. Durch die Möglichkeit zur Automatisierung und zeitabhängigen Planung können diese Funktionen sogar während unkritischer Zeiträume und ohne Anwesenheit der Administratoren durchgeführt werden.

Intelligente Auslastungsverteilung. Durch die parallele Überwachung von Zustand und Auslastung aller Systeme im Netzwerk und die Möglichkeit zur beliebigen Konfiguration der Komponenten kann eine auslastungsabhängige Konfiguration die Nutzung der Systeme optimieren. Treten Auslastungsspitzen auf, die eine Umverteilung der Aufgaben einzelner Systeme erfordern, so können über ManageSite innerhalb kürzester Zeit untätige Rechner vollständig umkonfiguriert und mit neuen Images versehen werden, so dass sie die Spitzen bei der Auslastung abfangen können. Wird diese Funktionalität auch noch automatisiert, so passen sich die Systeme dynamisch dem Bedarf an.

Avocent: DSI5100

Das Unternehmen Avocent International Ltd.[32] hat neben vielen anderen Herstellern auch die Technologie von OSA Technologies lizenziert und in ihre eigenen Produkte integriert. Mittlerweile haben die beiden Unternehmen sogar eine dauerhafte Allianz gebildet, um ihre Interessen besser umsetzen zu können. Neben der reinen Software-Unterstützung DSView 3 existiert auch das Hardware-Proxy Gerät DSI5100, über das sich bis zu 64 verschiedene IPMI-fähige Systeme administrieren lassen. Bei einem idealen Einsatzszenario wird der DSI5100 IPMI Proxy vor Ort zusammen mit mehreren zu verwaltenden Systemen installiert. Von jeder der einzelnen Komponenten wird eine direkte Verbindung zum Proxy hergestellt, wobei Verbindungen sowohl über Ethernet, KVM Switch oder Serielle Schnittstelle möglich ist. Alle Informationen laufen auf dem zentralen Proxy zusammen und können anschließend konsolidiert von einer Managementstation aus abgefragt werden. Weitere Besonderheiten des DSI5100 Proxy Gerätes werden im Folgenden näher beschrieben:

Kontrolliertes Powermanagement. Durch den zentralen IPMI Proxy kann ein Powermanagement aller angeschlossenen Geräte durchgeführt werden. Auf diese Weise können Systeme kontrolliert heruntergefahren oder auch neuinitialisiert werden, selbst wenn das Betriebssystem dieser Komponente nicht mehr reagiert. Die IPMI Spezifikation erlaubt außerdem das Einschalten eines ausgeschalteten Gerätes, da der zuständige BMC auch in diesem Fall noch über eine Stromversorgung verfügt.

Out-of-Band Management. Mit dem DSI5100 Proxy lässt sich leicht ein Out-of-Band Management der verwalteten Systeme implementieren. Die einzelnen Komponenten können über redundante Übertragungswege mit dem Proxy verbunden werden, über die ausschließlich das Management abgehan-

[32] www.avocent.com

delt wird. Die übrigen Netzwerkverbindungen sind vom Systemmanagement vollständig entkoppelt.

Vereinheitlichte Schnittstelle. Über den DSI5100 Proxy können verschiedene Systeme, die über unterschiedliche IPMI Schnittstellen zur Kommunikation verfügen, über einen einheitlichen Weg administriert werden. Im Normalfall verläuft die Kommunikation einer entfernten IPMI-fähigen Komponente entweder über die Serielle Schnittstelle, einen KVM Switch oder über die Netzwerkverbindung. Während die beiden ersten Verbindungen einer Punkt-zu-Punkt Verbindung entsprechen, kann über die Netzwerkschnittstelle prinzipiell eine unbeschränkte Anzahl anderer Kommunikationsverbindungen bestehen. Da außerdem die IPMI Spezifikation für eine LAN-Verbindung das Protokoll UDP vorsieht, ist eine Auslieferung der Pakete über weite Strecken mit unbekannter Auslastung nicht gewährleistet. Der IPMI Proxy aus dem Hause Avocent ermöglicht zu diesem Zweck die Transformation der kompletten Kommunikation zwischen der entfernten Managementstation und den einzelnen angeschlossenen Endgeräten in das verbindungsorientierte Protokoll TCP. Auf diese Weise können Störungen der IPMI Kommunikation insbesondere bei längeren Übertragungswegen vermindert werden.

Avocent: DSView 3

Die Management-Software DSView 3 stammt ebenfalls vom Hersteller Avocent International Ltd.[33] und dient als Ergänzung zum DSI5100 Proxy. Die Systemmanagement-Software ermöglicht die zentrale Abfrage der auf dem Proxy gesammelten Daten sowie eine Konfiguration der angeschlossenen Systeme. Die weiteren Besonderheiten des Tools werden im Folgenden zusammengefasst:

Vereinheitlichte Benutzeroberfläche. Im selben Maße, wie der DSI5100 Proxy die physikalischen Verbindungen zu den einzelnen Geräten zentralisiert und zu einer vereinheitlichten Verbindung zusammenfasst, so liefert auch die DSView 3 Management-Software eine einheitliche graphische Oberfläche für alle angeschlossenen Geräte. Es werden lediglich die nach der IPMI Spezifikation bekannten Messgrößen und Parameter herstellerunabhängig erfasst und konfiguriert. Dem Administrator bietet sich daher eine einheitliche graphische Schnittstelle zu allen angeschlossenen Systemen unterschiedlichster Architektur.

Flexible Fernadministration. Durch die Implementierung der graphischen Oberfläche in Form einer Web-Schnittstelle ist eine Administration der an den DSI5100 Proxy angeschlossenen Geräte von nahezu jedem Punkt aus möglich. Die benutzerabhängige Zugriffsbeschränkung sorgt gleichzeitig für einen Schutz der IPMI-Daten vor dem Zugriff Unbefugter. Es können außerdem bis

[33] www.avocent.com

zu 64 Geräte gleichzeitig mit nur einer einzigen Hardware-Komponente verwaltet werden. Dies erfordert auch nur den Zugang zu einem einzigen System, über das dann alle Geräte administriert werden können.

11.2.3 OpenSource Werkzeuge

Zwar gilt die IPMI Spezifikation nicht als offener Standard, da sie maßgeblich von nur vier Herstellern in Eigeninitiative entwickelt worden ist, dennoch hat der große Nutzen der Spezifikation das Interesse von OpenSource Entwicklern geweckt, so dass sich auch mehrere Projekte zu diesem Thema finden. Die Spannweite reicht dabei von einer Integration in den Linux Kernel bis hin zu einem vollwertigen Managementsystem.

GNU: FreeIPMI

Unter den OpenSource Werkzeugen soll beispielhaft das GNU[34] FreeIPMI ausgewählt und näher beschrieben werden. Das Werkzeug bestehen im Wesentlichen aus einer Sammlung von Tools, mit denen verschiedenste IPMI Aufgaben erfüllt werden können. Die einzelnen Tools werden im Folgenden kurz beschrieben:

FISH. Über die FreeIPMI Shell (FISH) können die im Folgenden weiter erläuterten verschiedenen Erweiterungen und Befehle von FreeIPMI ausgeführt werden. Außerdem umfasst das Tool ein umfangreiches Hilfesystem für die Erweiterungen.

BMC-CONFIG. Mit dem BMC-CONFIG Befehl kann der BMC eines IPMI Systems konfiguriert werden. Die Besonderheit des Befehls liegt in der einfachen Möglichkeit zur Verwaltung der vollständigen BMC Konfiguration in einer externen Datei. Mit dem Kommandozeilenparameter `--checkout` werden sämtliche Konfigurationseinstellungen des BMC auf das lokale Administrationssystem übertragen und können dort nach Belieben in einer Datei oder auch Datenbank gespeichert werden. Zur Zeit werden noch nicht alle Einstellungen unterstützt, es können aber bereits ausführliche Angaben zu den IPMI Benutzern, den Netzwerkparametern und den Sicherheitsmechanismen verarbeitet werden. Der Kommandozeilenparameter `--commit` lädt in umgekehrter Weise den Inhalt einer Konfigurationsdatei in den entsprechenden BMC. Mit `--key-pair` können auch nur einzelne Wertepaare auf dem BMC konfiguriert werden. Schließlich kann mit dem Kommandozeilenparameter `--diff`

[34] Bei „GNU“ handelt es sich um eine rekursive Abkürzung mit der Definition „GNU's not Unix“. Diese Abkürzung bedeutet so viel, dass die Philosophie der quelloffenen Entwicklung und der Weitergabe von Software zur freien Benutzung nicht mit den Unix Betriebssystemen gleichzusetzen ist. Vielmehr existieren durchaus auch kommerzielle Unix Betriebssysteme, auch wenn diese vielfach OpenSource Tools integrieren.

ein einfacher Vergleich zwischen der Konfiguration des BMC und der lokal gespeicherten Konfigurationsdatei durchgeführt werden.

BMC-INFO. Der BMC-INFO Befehl dient nicht der Konfiguration des BMC, sondern mit diesem Kommando können lediglich generelle Informationen über den BMC eingeholt werden. Zu den zurückgelieferten Angaben gehören beispielsweise die Geräteidentifikationsnummern des BMC, die Version der Firmware, die unterstützte IPMI Versionsnummer und eine Auflistung der vom BMC unterstützten Geräte und Sensoren.

SENSORS. Der SENSORS implementiert eine der wichtigsten Funktionalitäten von FREEIPMI. Mit ihm können die Zustandswerte aller Sensoren im IPMI System abgefragt und dargestellt werden. Dazu zählen auch die Angaben zu den jeweiligen Schwellwerten der einzelnen Sensoren. Da in einem einzigen System sehr viele Sensoren untergebracht sein können, lässt sich die Ausgabe auch auf einzelne Sensorgruppen beschränken. Mit dem Kommandozeilenparameter `--list-groups` können die im BMC vorhandenen Sensorgruppen aufgelistet werden. Mit den Kommandozeilenparametern `--group` und `--sensors` kann dann die Ausgabe auf einzelne Gruppen oder sogar einzelne Sensoren beschränkt werden.

SEL. Der FREEIPMI Befehl SEL liefert eine Schnittstelle zum System Event Log (SEL) des BMC. Ein einfacher Aufruf des Befehls listet den vollständigen Inhalt der SEL Datenbank auf. Mit den Kommandozeilenparametern `--delete` und `--delete-all` können auch einzelne Einträge oder die gesamte Datenbank gelöscht werden.

IPMIPOWER. Um den Einschaltzustand von IPMI Geräten konfigurieren zu können, ist in FREEIPMI der Befehl IPMIPOWER implementiert. Die wichtigsten Kommandozeilenparameter dienen dem Einschalten, dem Ausschalten und dem Neustarten des Systems. Dazu werden die Parameter `--on`, `--off`, `--cycle`, `--reset` und `--soft` verwendet. Eine Abfrage des aktuellen Einschaltzustands ist mit dem Kommandozeilenparameter `--stat` ebenfalls möglich.

IPMIPING. Wie der Name des Tools bereits vermuten lässt, kann mit dem IPMIPING Befehl die Unterstützung der IPMI Spezifikation auf einzelnen Systemen verifiziert werden. Mit dem Kommandozeilenparameter `-v` kann nicht nur überprüft werden, ob ein System die IPMI Spezifikation unterstützt, sondern es kann auch eine Liste der unterstützten Sicherheitsmechanismen abgefragt werden.

RMCPPING. Der RMCPPING Befehl ist vergleichbar mit dem IPMIPING Befehl, unterstützt jedoch nicht die Abfrage der vom System unterstützten Sicherheitsmechanismen.

11.3 IEEE 802.1X Werkzeuge

Zur Implementierung einer IEEE 802.1X Infrastruktur werden prinzipiell drei verschiedene Komponenten benötigt. Zentraler Bestandteil ist dabei der Authenticator, dessen Ports es abzusichern gilt. Auf den Client-Rechnern liegen die Hauptaufgaben beim Supplicant, der für die Abwicklung der Authentifizierung auf den neu zum Netzwerk hinzugefügten Geräten zuständig ist. Als dritte und letzte Komponente verbleibt der Authentication Server, welcher die Überprüfung der von den Supplicants übermittelten Authentifizierungsparametern vornimmt und über den Erfolg der Authentifizierung entscheidet. Um bestehende Netzwerke um die 802.1X Funktionalität zu erweitern, müssen also alle drei Komponenten umgesetzt werden. Die zentrale Rolle des Authenticators muss entweder in Hardware integriert sein – beispielsweise im Betriebssystem eines Switches – oder aber mittels spezieller Software umgesetzt werden. Die zweite Lösung ist eher unüblich und eignet sich nur für kleinere Netzwerke. Im Normalfall wird die Variante der Implementierung aller IEEE 802.1X Funktionalitäten in den zentralen Netzwerkknoten eingesetzt. Aus diesem Grund zielen die meisten 802.1X Software-Produkte entweder auf die Umsetzung eines Supplicants in den Clients oder auf eine Implementierung eines Authentication Servers.

11.3.1 Kommerzielle Werkzeuge

Im Folgenden wird eine Auswahl an kommerziellen IEEE 802.1X Werkzeugen vorgestellt, ohne dabei eine Wertung vornehmen zu wollen. Die Liste ist auf keinen Fall vollständig und soll auch in der Auswahl der gezeigten Produkte keinerlei Wertung vornehmen. Außerdem wird die Rolle des Authenticators oftmals nicht von reiner Software ausgeübt, sondern ist vielfach in Hardware integriert, so dass hier nicht ausschließlich von Werkzeugen gesprochen werden kann.

Meetinghouse: AEGIS Client

Das Unternehmen Meetinghouse[35] liefert für die beiden typischen Ansatzpunkte Supplicant und Authentication Server des IEEE 802.1X Standards eine Lösung an. Mit dem AEGIS Client wird eine Implementierung für den Supplicant angeboten, mit dem sich nahezu beliebige Client-Geräte standardkonform an ein IEEE 802.1X Netzwerk anschließen lassen. Als Authentication Server bietet sich der AEGIS Server aus demselben Hause an, jedoch kann auch jedes andere EAP-fähige System verwendet werden. Die wichtigsten Eigenschaften des Clients werden im Folgenden zusammengefasst:

Breite Unterstützung von Betriebssystemen. Der AEGIS Client überzeugt vor allem durch seine breite Unterstützung von Betriebssystemen.

[35] www.mtghouse.com

Die Client-Software lässt sich auf den verschiedensten Systemen installieren, darunter eine große Anzahl an Betriebssystemen aus dem Hause Microsoft. Neben den Systemen Windows 98, Windows 98SE, Windows ME und Windows NT4 werden auch die neueren Systeme Windows 2000 und Windows XP unterstützt, für die auch herstellereigene 802.1X-Lösungen existieren. Außerdem werden mit einer Implementierung für Apples Mac OS 10, Red Hat Linux 8, Red Hat Linux 9 und Solaris 8 auch noch Betriebssysteme anderer Hersteller unterstützt. Interessant sind vor allem noch die Implementierungen für verschiedene PDAs, wie beispielsweise die Unterstützungen für Palm Tungsten C, Microsoft Pocket PC 2002 und Microsoft Windows CE.net 4, welche die Liste der möglichen Betriebssysteme abrunden.

Flexible Authentifizierungsmöglichkeiten. Auch die Unterstützung der im IEEE 802.1X Standard definierten Authentifizierungsmethoden ist im AEGIS Client auf Flexibilität ausgelegt. Implementiert sind die Authentifizierungsmethoden EAP-TLS, EAP-TTLS und PEAP sowie die beiden Mechanismen LEAP und EAP-MD5, die ausschließlich für den Palm PDA implementiert sind. Die PEAP Authentifizierungsmethode kann außerdem zusammen mit EAP-TLS, EAP-MSCHAPv2 und EAP-GTC eingesetzt werden; EAP-TTLS wiederum kann in Kombination mit PAP, CHAP, MS-CHAP, MS-CHAPv2 und EAP-MD5 verwendet werden.

Unabhängigkeit vom Authentication Server. Zwar arbeitet der AEGIS Client naturgemäß ideal mit dem AEGIS Server aus demselben Hause zusammen, jedoch können auch andere Authentication Server problemlos verwendet werden. Direkt unterstützt werden daher auch zusätzlich noch der Odyssey oder der Steel Belted RADIUS Server des Unternehmens Funk, der Internet Authentication Service (IAS) von Microsoft oder auch der Secure Access Control Server (ACS) von Cisco. Auf diese Weise lässt sich die Client-Software auch sehr gut in bestehende Netzwerkstrukturen integrieren.

Unterstützung für kabellose Clients. Zusätzlich zur kabelgebundenen IEEE 802.1X Implementierung, die vorwiegend über die zentralen Switches umgesetzt wird, arbeitet der AEGIS Client auch mit der WLAN Technologie zusammen. Unterstützt wird dabei vor allem der Wireless-Fidelity Protected Access (WPA) [229] Standard, der sowohl mit dynamischem WEP als auch mit vordefinierten Schlüsseln verwendet werden kann.

Meetinghouse: AEGIS Server

Neben der entsprechenden Implementierung eines Supplicant bietet das Unternehmen Meetinghouse[36] auch eine Lösung für den Authentifizierungsserver an. Der AEGIS Server implementiert die Anforderungen des 802.1X Standards an einen Authentication Server. Eine Zusammenarbeit mit dem AEGIS Client aus demselben Hause funktioniert reibungslos; andere IEEE 802.1X

[36] www.mtghouse.com

Clients werden aber gleichermaßen gut unterstützt. Die wichtigsten Eigenschaften des Servers werden im Folgenden aufgelistet:

Breite Unterstützung von Betriebssystemen. Wie auch der AEGIS CLIENT ist der AEGIS SERVER für eine Unterstützung einer breiten Palette von Betriebssystemen ausgelegt. Der Fokus liegt dabei aber besonders auf den Server-Betriebssystemen. Daher sind auch nur Versionen für die Microsoft-Server Windows 2000, Windows 2003 und Windows XP verfügbar. Darüber hinaus werden noch die Systeme Red Hat Linux 7, Red Hat Linux 8, Red Hat Linux 9 und Solaris 8 unterstützt.

Flexible Authentifizierungsmöglichkeiten. Im Wesentlichen werden vom AEGIS SERVER dieselben Authentifizierungsmethoden unterstützt, die auch im AEGIS CLIENT implementiert sind. Dies sind vor allem die Algorithmen EAP-TLS, EAP-TTLS, PEAP, LEAP, EAP-MSCHAPv2 und EAP-MD5. Auf diese Weise kann eine breite Unterstützung verschiedenster Client-Produkte garantiert werden.

Proxy Funktionalität. Sind im Netzwerk bereits andere RADIUS Server installiert, die nicht über eine IEEE 802.1X Unterstützung in Form des EAP Protokolls verfügen, so kann der AEGIS SERVER alternativ auch als Authentication Server Proxy fungieren. In diesem Fall nimmt der AEGIS SERVER die EAP Pakete entgegen und führt anschließend seinerseits eine Anfrage beim bereits bestehenden Authentifizierungsserver mit der Benutzerdatenbank durch. Da der AEGIS SERVER auch das Protokoll LDAP unterstützt, können als externe Authentifizierungsserver beispielsweise auch Microsoft Domainserver angesprochen werden. Eine doppelte Benutzerhaltung entfällt unter diesen Umständen, da auf die bereits bestehende Benutzerdatenbank problemlos zurückgegriffen werden kann.

Cisco: IOS

Als Hardware-Hersteller liegt der Fokus des Unternehmens Cisco Systems Inc.[37] primär auf der Unterstützung der Rolle des Authenticators im IEEE 802.1X Standard. Die Switches als zentrale Geräte in der Port-basierten Zugangskontrolle müssen eine ganze Reihe verschiedener Technologien implementieren, um eine 802.1X Umgebung aufbauen zu können. In Ciscos Betriebssystem Internet Operation System (IOS) sind in der neueren Version 12.4 bereits alle für die Implementierung einer Port-basierten Zugangskontrolle notwendigen Mechanismen integriert. Eine explizite Unterstützung von verschiedenen Authentifizierungsmethoden ist nicht zwingend notwendig, da der Switch in seiner Rolle als Authenticator lediglich Pakete zwischen Supplicant und Authentication Server transformiert und transportiert. Die Details der Authentifizierung müssen nur im Client und im Authentifizierungsserver implementiert sein. Als Ergebnis der Authentifizierung erhält der Authenticator

[37] www.cisco.com

schließlich eine positive – oder negative – Rückmeldung vom Authentication Server, welche dann den Status des Ports entweder auf *authenticated* oder *unauthenticated* setzt. Abbildung 11.1 zeigt ein Beispiel für die 802.1X Konfiguration auf einem Cisco Switch mit IOS 12.4 Betriebssystem.

```
Router# configure terminal
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default radius
Router(config)# radius-server host 172.20.2.107
Router(config)# radius-server host 172.20.2.108
Router(config)# radius-server retransmit 5
Router(config)# radius-server timeout 10
Router(config)# radius-server key myverysecretradiuskey
Router(config)# dot1x system-auth-control
Router(config)# dot1x reauthentication
Router(config)# dot1x timeout re-authperiod 1000
Router(config)# dot1x timeout quiet-period 30
Router(config)# dot1x timeout tx-period 60
Router(config)# dot1x timeout supp-timeout 20
Router(config)# dot1x timeout server-timeout 20
Router(config)# dot1x max-req 3
Router(config)# interface fastethernet 0
Router(config-if)# dot1x port-control auto
```

Abb. 11.1. Konfigurationsangaben für einen Cisco Switch mit installiertem Betriebssystem IOS 12.4 zur Aktivierung des 802.1X Authentifizierungsmechanismus.

Cisco: Secure Access Control Server

Neben der Hardware-seitigen Unterstützung des 802.1X Standards in den verschiedenen Switches von Cisco Systems Inc.[38] implementiert der Cisco Secure ACS auch einen 802.1X-fähigen Authentication Server. Der Server kann nur auf einem Windows Betriebssystem installiert werden, wird aber optional bereits vorinstalliert auf einer eigenen Appliance ausgeliefert. Die wichtigsten Merkmale der Server-Software sind im Folgenden zusammengefasst:

Flexible Benutzerdatenbank. Der Cisco Secure ACS kann sowohl als RADIUS Server als auch als Cisco proprietärer TACACS+ Server fungieren. Zusätzlich kann er die Aufgabe eines Proxy Servers übernehmen. Mit der Unterstützung des Protokolls LDAP kann die Benutzerdatenbank außerdem nicht nur lokal, sondern auch auf externen Systemen implementiert sein. Ei-

[38] www.cisco.com

ne Integration in Microsoft, Novell oder Unix Systeme ist damit problemlos möglich.

Flexible Authentifizierungsmöglichkeiten. Der Cisco SECURE ACS unterstützt eine Reihe von Authentifizierungsmethoden, zu denen vor allem die Standards PEAP, LEAP, EAP-MSCHAPv2, EAP-GTC, EAP-TLS und EAP-MD5 zählen. Außerdem wird das Cisco proprietäre Verfahren Extensible Authentication Protocol – Flexible Authentication Via Secure Tunneling (EAP-FAST) unterstützt.

Dynamischer Zugriffsschutz. Eine Besonderheit des Cisco SECURE ACS ist die Möglichkeit, mit der Authentifizierung eines Client gleichzeitig auch die Regeln für dessen weitere Zugriffsbeschränkungen zu installieren. In Kombination mit entsprechenden anderen Cisco Geräten, welche einen Zugriffsschutz mittels Access Control Lists (ACL) unterstützen, kann bei der Anmeldung eines Benutzers auch dynamisch eine zugehörige Zugangskontrollliste auf diesem Gerät installiert werden. Somit kann eine benutzerabhängige Zugangsbeschränkung eingerichtet werden.

11.3.2 OpenSource Implementierung

Da es sich beim IEEE 802.1X Standard um einen offenen Standard handelt, ist auch eine entsprechende OpenSource Umsetzung nicht ungewöhnlich. Das wohl beste 802.1X Tool findet sich beim OPEN1X Projekt, das zur Zeit vorrangig eine Implementierung für einen Supplicant bietet.

OPEN1X

Eine durchaus gute Umsetzung des IEEE 802.1X Standards findet sich beim OPEN1X Projekt[39], das sich um eine Linux Umsetzung für einen Supplicant und einen Authenticator bemüht. Zur Zeit befindet sich allerdings nur der Supplicant XSUPPLICANT in einem fortgeschrittenen Entwicklungsstadium. Die Besonderheiten sind im Folgenden aufgelistet:

IEEE 802.1X Unterstützung für Linux Clients. Der IEEE 802.1X Mechanismus setzt an einem sehr frühen Zeitpunkt beim Starten des Betriebssystems an. Beispielsweise kann die Implementierung erfordern, dass eine 802.1X Authentifizierung erfolgen muss, noch bevor ein Zugang zum DHCP Server zur Zuweisung einer IP Adresse besteht. Außerdem arbeitet der Mechanismus auf einem niedrigen Level in den Netzwerkschichten. Aus diesem Grund ist eine plattformübergreifende Entwicklung nur schwerlich möglich. Während die kommerziellen Betriebssysteme proprietäre Lösungen für die 802.1X Integration entwickeln, wird die Unterstützung für Linux Systeme auf Basis der quelloffenen OPEN1X Entwicklung vorangetrieben. Auf diese Weise können die

[39] www.open1x.org

verschiedenen Distributionen um eine IEEE 802.1X Unterstützung erweitert werden.

Vielfältige Authentifizierungsmöglichkeiten. Besonders überzeugend ist die breite Implementierung der verschiedensten Authentifizierungsmethoden im OpenSource Tool OPEN1X. Die unterstützte Palette umfasst die Methoden EAP-AKA, EAP-MD5, EAP-MSCHAPv2, EAP-OTP, EAP-SIM, EAP-TLS, LEAP, PEAP mit MS-CHAPv2 und EAP-TTLS mit einer Unterstützung der Verfahren CHAP, MS-CHAP, MS-CHAPv2 und PAP. Zukünftig sind auch noch eine Implementierung für den EAP-GTC Mechanismus sowie für Ciscos proprietären EAP-FAST Mechanismus geplant.

Flexible Unterstützung von Authentication Servern. Da eine separate Entwicklung für einen OpenSource Authentication Server im OPEN1X Projekt nicht vorgesehen ist, wird gleichzeitig auch eine große Anzahl von Authentifizierungsservern unterstützt. Die unterschiedlichen Server-Produkte unterstützen ihrerseits teilweise nur einen ausgewählten Anteil der möglichen EAP Authentifizierungsmethoden. Die der OPEN1X Dokumentation entnommene Tabelle 11.1 listet in einer Matrixform auf, welche speziellen Authentifizierungsmechanismen auf den einzelnen Authentifizierungsservern in Verbindung mit dem quelloffenen Tool unterstützt werden.

WLAN Erweiterbarkeit. Das OPEN1X Projekt liefert keine eigene Lösung für die Unterstützung von WLAN Clients und der WPA Spezifikation. Es existiert jedoch ein eigenes OpenSource Projekt[40], das mit dem WPA_SUPPLICANT diese Möglichkeit bietet. Dieser lässt sich außerdem leicht in Kombination mit dem XSUPPLICANT verwenden.

FREERADIUS

Zwar handelt es sich beim FREERADIUS[41] Projekt nicht um einen dedizierten Authentication Server, jedoch stellt der quelloffene RADIUS Server eine recht gute Ergänzung zum OPEN1X dar. Neben dem kommerziellen Server RADIATOR[42] bietet der OpenSource RADIUS Server eine Unterstützung der meisten Authentifizierungsmethoden. Alle weiteren Details finden sich im Folgenden:

Plattformunabhängigkeit. FREERADIUS lässt sich nicht auf allen Betriebssystemen installieren, jedoch werden beinahe alle Unix-Varianten unterstützt. Als Besonderheit kann hier die getestete Unterstützung von Tools angesehen werden, die eine Unix-Umgebung unter Microsoft Windows bieten.

Vielfältige Authentifizierungsmöglichkeiten. Vor allem an der Unterstützung der verschiedensten Authentifizierungsmethoden kann man erken-

[40] `hostap.epitest.fi/wpa_supplicant`
[41] `http://www.freeradius.org`
[42] `http://www.open.com.au/radiator`

Tabelle 11.1. Unterstützung verschiedener IEEE 802.1X Authentifizierungsmechanismen auf unterschiedlichen Authentifizierungsservern. Ein ‚×'-Zeichen in der Matrix steht für eine getestete Kompatibilität von OPEN1X und dem entsprechenden Authentifizierungsmechanismen auf dem jeweiligen Server. Ein ‚+'-Zeichen steht für einen Authentifizierungsmechanismus, der auf dem jeweiligen Server zwar implementiert ist, von OPEN1X aber noch nicht unterstützt wird. Ein ‚!'-Zeichen steht für eine Inkompatibilität zwischen OPEN1X und dem entsprechenden Authentifizierungsmechanismus auf dem jeweiligen Server und ein ‚-'-Zeichen steht für einen auf dem entsprechenden Server nicht unterstützten Authentifizierungsmechanismus.

	Cisco ACS	Free-Radius	Funk SBR	Infoblox	AEGIS	MS IAS	Radiator	Roving Planet
EAP-AKA	-	-	-	-	-	-	×	-
EAP-FAST	+	-	-	-	-	-	-	-
EAP-GTC	-	-	-	-	-	-	×	-
EAP-MD5	×	×	×	×	×	-	×	×
EAP-OTP	-	-	-	-	-	-	×	-
EAP-SIM	-	×	-	-	-	-	×	-
EAP-TLS	+	+	+	+	+	+	×	+
LEAP	+	+	+	-	+	-	×	+
PEAP-GTC	+	-	-	-	+	-	+	-
PEAP-MSCHAPv2	×	×	×	!	×	×	×	+
TTLS-CHAP	-	×	+	+	+	-	×	+
TTLS-MSCHAP	-	×	+	+	+	-	×	+
TTLS-MSCHAPv2	-	+	+	+	+	-	×	+
TTLS-PAP	-	+	×	!	×	-	×	+

nen, dass es sich bei FREERADIUS nicht um einen reinen IEEE 802.1X Authentication Server handelt. Auf der einen Seite besteht die Möglichkeit zur Authentifizierung via EAP-MD5, EAP-PEAP, EAP-MSCHAPv2, EAP-TLS, EAP-TTLS, EAP-SIM, EAP-GTC sowie LEAP. Auf der anderen Seite sind noch eine ganze Reihe anderer Authentifizierungsmethoden in den quelloffenen RADIUS Server integriert. Dazu zählen vorrangig PAP, CHAP, MS-CHAP, MS-CHAPv2 oder auch LDAP. Darüber hinaus kann eine Authentifizierung auch über verschiedene Andere Kommunikationswege erfolgen. Beispielsweise können Benutzer anhand der lokalen `/etc/passwd` Datei oder über Pluggable Authentication Modules (PAM) authentifiziert werden. Die Liste der unterstützten Authentifizierungsmethoden lässt sich beinahe beliebig fortsetzen; zur Verdeutlichung der außerordentlichen Flexibilität sollen die angeführten Beispiele aber ausreichen.

Proxy Funktionalität. Befinden sich noch andere RADIUS Server im Netzwerk, so kann FREERADIUS optional auch als Authentication Server Proxy fungieren. In diesem Fall werden die empfangenen EAP Pakete einfach an den vorhandenen Authentifizierungsserver weitergeleitet. Typischerweise handelt es sich bei Servern um LDAP Server oder auch Microsoft Domainserver.

Fernadministration. Der FreeRADIUS Server verfügt über eine Web-Schnittstelle, mit welcher sowohl die Benutzerdatenbank als auch globale Einstellungen verwaltet werden können. Außerdem können auf demselben Weg auch Statistiken des RADIUS Servers ausgelesen werden.

12

Bilanzierung

In einer perfekten Welt besitzt man den perfekten Schutz und die Sicherheit des Netzwerks und des Netzwerkmanagements ist immer gewährleistet. In einer perfekten Welt existieren allerdings auch keine Angreifer und niemand macht Fehler bei der Konfiguration seiner Komponenten. Aber die Welt ist alles andere als perfekt: Seit etwa 15 Jahren verfolgt das Computer Emergency Response Team / Coordination Center (CERT/CC) [40] die Entwicklungen im Internet. Auslöser für die Gründung des Teams war das Erscheinen des Wurms beim so genannten ‚Morris Wurm Zwischenfall' im Jahr 1988. Der Wurm hat damals etwa 10 Prozent des Internets zu einem vollständigen Halt gebracht, so dass die damals für das Internet verantwortlich zeichnende Defense Advanced Research Projects Agency (DARPA) [50] das Software Engineering Institute (SEI) an der Carnegie Mellon University in Pittsburgh mit der Bildung eines Zentrums beauftragte, welches die Kommunikation und den Informationsaustausch zwischen den Sicherheitsexperten weltweit fördern und unterstützen sollte. Eine der Hauptaufgaben des CERT/CC besteht in der Analyse von Verwundbarkeiten und anschließender Veröffentlichung der Angriffsarten in Form von „Advisories". Außerdem hat sich das CERT/CC zur Aufgabe gemacht, alle gesammelten und erarbeiteten Informationen mit dem Rest des Internets zu teilen.

Mit Erscheinen des „1i0n" oder „lion" [87] Wurms im Jahr 2001 wurde eine ganz andere Institution gegründet, die sich als Frühwarnsystem für globale Angriffe im Internet versteht. Eines der wichtigsten Werkzeuge des Internet Storm Center (ISC) [98] sind Intrusion Detection Systeme, die eine schnelle Erkennung von bekannten und auch unbekannten Angriffsformen ermöglichen. Durch die Koordination von Informationen aus mittlerweile über 50 Ländern und über 500.000 IP Adressen können Angriffe mit einer hohen Wahrscheinlichkeit aufgespürt und unmittelbar dem gesamten Internet bekannt gemacht werden. Die veröffentlichten Informationen beinhalten vor allem auch eine Übersichtsseite mit Angaben zu unerwünschten Aktivitäten. Ein Diagramm in Form einer Weltkarte liefert Aufschluss über die Herkunft aller erkannten Angriffe. Dabei wird zwischen den sechs Kontinenten Nordamerika, Europa,

Asien, Südamerika, Australien und Afrika (in der Reihenfolge ihres Netzwerkaufkommens) unterschieden. Ein Tortendiagramm für jeden Kontinent gibt zusätzlich an, welcher Art die Angriffe sind. Als Kriterium dienen hierfür die angegriffenen Portnummern.

12.1 Notwendigkeit

Die Geschichte der beiden Institutionen zeigt, dass sich das Bild des Internets gewandelt hat. Konnten früher einzelne Angriffe aufgespürt, analysiert und unschädlich gemacht werden, so übertrifft die Anzahl der heute bekannten Angriffe alle Vorstellungen. Es geht nicht mehr nur darum, detailliert über einzelne Angriffe zu informieren, sondern vielmehr auch über die aktuellen Trends bei den Angriffen zu berichten. Ebenso ist nicht mehr die Wahrscheinlichkeit relevant, mit der ein Angriff auf das Netzwerk und das Netzwerkmanagement erfolgt, sondern eher die Häufigkeit der Angriffe, die innerhalb einer Minute das Netzwerk treffen. Aus diesem Grund sind Schutzmechanismen beim Betrieb eines Netzwerks und des dazugehörigen Netzwerkmanagements absolut unverzichtbar geworden. Es stellt sich nun lediglich noch die Frage, mit welcher Intensität der Schutz des Netzwerks und des Netzwerkmanagements betrieben werden soll.

12.2 Kostenrechnung

Alle Betreiber von Netzwerken sind sich mittlerweile im Klaren über die Notwendigkeit zur Einrichtung von Schutzmechanismen für das Netzwerk. Aber die Errichtung von Schutzmechanismen kostet Geld, über das in vielen Fällen nicht die Betreiber und Betreuer des Netzwerks entscheiden, sondern Leute, die sich um die Bilanzierung kümmern. Dies ist ebenfalls wichtig, da sonst die Kosten für den Schutz des Netzwerks zu hoch für einen rentablen Betrieb werden können. Diese Einsicht hat sich in den letzten Jahren sowohl bei den Administratoren als auch bei den Controllern durchgesetzt. Man hat sich in den letzten Jahren durchaus Gedanken über die Bilanzierung eines Netzwerkes und dessen zugehöriger Schutzmechanismen gemacht.

12.2.1 Kosten für die Netzwerkinfrastruktur

Betrachtet man ein Netzwerk zunächst ohne Berücksichtigung von Schutzmechanismen ausschließlich als Infrastruktur zur Erfüllung der betrieblichen Aufgaben, so lassen sich – auf verschiedensten Wegen – Berechnungen zu den Kosten und zum Nutzen der Netzwerke durchführen. Zu den positiven Faktoren eines Netzwerkes zählen dabei vorrangig die Arbeitszeitersparnis und die damit verbundene Effizienzsteigerung der Mitarbeiter. Demgegenüber stehen

die Kosten, die ein Netzwerk verursacht. Das sind neben den Anschaffungskosten vor allem die Wartungskosten. An dieser Stelle sollen die genauen Rechenschritte nicht weiter vertieft werden; es sollte aber jedem klar sein, dass ein fähiger Betriebswirtschaftler aus diesen Zahlen genau errechnen kann, unter welchen Bedingungen der Betrieb eines Netzwerkes noch rentabel ist. Ähnliche Berechnungen lassen sich auch für Netzwerke anstellen, die nicht dem Eigenbedarf dienen, sondern deren Vermietung einen Gewinn einspielen soll.

12.2.2 Kosten für den Betrieb des Netzwerkes

In den berechneten Kosten findet sich immer auch ein Posten für den Betrieb des Netzwerks. Darunter finden sich Posten für spezielle Hardware und Software genauso wie die Schulungs- und Personalkosten. Es besteht im Allgemeinen Einigkeit darüber, dass die aufgewendeten Kosten für das Netzwerk und für den Betrieb des Netzwerkes in einem direkten Verhältnis zur erreichten Verfügbarkeit stehen. Nicht nur den Administratoren sondern inzwischen auch den Controllern ist bekannt, dass für eine hohe Verfügbarkeit auch eine entsprechende Redundanz implementiert werden muss. Die dadurch größere Infrastruktur erfordert wiederum mehr Personal, was zu höheren Kosten für Anschaffung und Betrieb führt. Die Vorgaben für die Verfügbarkeit kommen vielfach nicht von den Administratoren, sondern von anderer Stelle. Die Berechnung des daraus resultierenden Budgets für das Netzwerk sollten allerdings nicht ohne die Betreuer des Netzwerks gemacht werden, da sich lokale Besonderheiten immer auf die Kosten auswirken. Zwar betrachten sich deshalb die Betriebswirtschaftler und die Administratoren als Gegner in einem Kampf um das Netzwerk; solange beide Parteien jedoch die Meinungen des Gegenübers respektieren, kann man sich auf gemeinsame Lösungen verständigen.

12.2.3 Kosten für die Netzwerksicherheit

Schwieriger wird der Sachverhalt, wenn zusätzlich zur Netzwerkinfrastruktur noch die notwendigen Schutzmechanismen berücksichtigt werden müssen. Hier stoßen Administratoren und Betriebswirtschaftler noch häufiger zusammen. Im Normalfall sind sich beide Seiten darüber einig, dass jedes Netzwerk vor den Angriffen aus dem Internet geschützt werden muss. Die dafür aufzuwendenden Mittel liefern jedoch oftmals einen Streitpunkt. Die Errichtung von Schutzmechanismen ist immer mit Kosten verbunden – und das sind in vielen Fällen nicht nur einmalige Anschaffungskosten, sondern leider auch teuere Wartungskosten und Aufwendungen für den Betrieb der Schutzeinrichtungen. Betriebswirtschaftler vertreten in vielen Fällen die Meinung, dass sich Schutzmechanismen durch die einmalige Anschaffung von entsprechenden Geräten wie Firewalls, VPN Servern, IDS Systemen oder Virenscannern errichten lassen. Die Erfahrung hat allerdings gezeigt, dass eine dauerhafte und ununterbrochene Betreuung dieser Geräte unumgänglich ist. Damit beispielsweise

ein IDS System seine Wirkung beibehält, sind dauerhafte Investitionen in die Analyse der aufgezeichneten Daten notwendig. Dazu kommt noch der notwendige Zeitaufwand für die Beschaffung von aktuellen Informationen. Nur wenn die Administratoren bei den Angriffen und den zugehörigen Verteidigungsmaßnahmen ständig auf dem Laufenden bleiben, haben sie auch eine Chance, die Sicherheit des Netzwerkes langfristig zu wahren.

Viele der Probleme bei der Berechnung der Kosten für die Sicherheit des Netzwerks resultieren aus dem Sachverhalt, dass man nicht für eine Leistung bezahlt, sondern für das Nichteintreten eines mehr oder weniger wahrscheinlichen Ereignisses. Hier stoßen klassische Berechnungsverfahren an ihre Grenzen, da den Kosten nicht mehr eine eindeutig messbare Leistung gegenübersteht. Um nicht-funktionale Anforderungen wie Sicherheit dennoch kalkulieren zu können, muss man das erreichte Sicherheitsniveau in eine imaginäre Leistung umrechnen. Es hat sich als praxisnah erwiesen, die Kosten für den Erhalt der Sicherheit mit den potentiellen Kosten für das Durchbrechen der Sicherheitsmechanismen in Relation zu setzen. Dabei müssen alle der in Kapitel 9 beschriebenen Bedrohungen berücksichtigt werden. Berechnet man die möglichen Folgen eines erfolgreichen Angriffs, so lässt sich daraus direkt der zu schützende Wert ableiten. Für die Sicherheit sollte also immer berücksichtigt werden, welchen Wert es zu schützen gilt.

Leider stehen die Kosten für den Schutz vor einer Bedrohung nicht zwangsweise in Abhängigkeit zu den Kosten für das Durchbrechen des entsprechenden Sicherheitsmechanismus. Die Kostenrechnung kann manchmal zu Gunsten des „Verteidigers“ – also des Netzwerkbetreibers – aufgehen. Beispielsweise lässt sich ein sicheres Passwort ohne nennenswerte Kostenverursachung erzeugen. Abschnitt 9.2.2 verdeutlicht anschaulich, wie man ohne große Anstrengung ein sicheres Passwort erzeugen kann, dass durch seine hohe Merkbarkeit vom Benutzer nicht niedergeschrieben werden muss. Werden nun ausschließlich Speichermethoden und Übertragungsverfahren für Passwörter eingesetzt, wie es heute fast immer der Fall ist, so stehen die Kosten für eine Kompromittierung des Passwortes in keinem Verhältnis zu den Kosten der Erzeugung des Passwortes.

Häufig geht die Kostenrechnung jedoch zu Gunsten des Angreifers auf. Dies lässt sich einfach an dem Bild eines Zauns erklären, der ein Grundstück vor dem Zutritt Fremder schützen soll. Während der Besitzer des Grundstücks dafür sorgen muss, dass jeder Meter des Zauns gleichermaßen abgesichert wird, reicht dem Einbrecher bereits eine einzige Schwachstelle aus, um seine Chancen für ein unentdecktes Eindringen zu erhöhen. Es ist leicht verständlich, dass die Kosten für die Aufstellung, Wartung und Bewachung des Zauns ständig steigende Kosten verursachen, je länger der Zaun wird. Gleichzeitig bleiben die Investitionen für einen Einbrecher bei gleichbleibender Schutzwirkung des Zauns nahezu konstant. Je mehr Dienste der Betreiber eines Netzwerkes für das Internet oder zumindest für Teile des Internets zur Verfügung stellt, desto höher ist der zu betreibende Aufwand für das Aufrechterhalten der Sicherheit des Netzwerkes.

Liegen die Anforderungen für den Schutz des Netzwerkes sehr hoch, wie es beispielsweise auf nationaler Ebene eines Staates der Fall ist, so können die Kosten für den Schutz weit höher als die Kosten für einen Angriff werden. Bei diesen Dimensionen erreicht man den paradoxen Zustand, dass es deutlich günstiger ist, den Informationsrückstand gegenüber anderen durch Informationsdiebstahl auszugleichen. Gleichzeitig kann paradoxerweise sogar ein Informationsvorsprung durch Informationsverfälschung oder Informationsverlust bei den anderen aufgebaut werden.

Mit wesentlich weniger trivialen Berechnungen lassen sich aber auch für diesen Fall Kompromisse schließen, die ein gesundes Maß an Sicherheit mit einem vertretbaren Kostenaufwand liefern. Glücklicherweise lassen sich mit bereits geringen Kosten viele „Gelegenheitsangriffe" beispielsweise von den Script-Kiddies abwehren. Diese Investitionen liefern ein Mindestmaß an Sicherheit bei gleichzeitig minimalen Kosten. Man sollte sich jedoch immer im Klaren darüber sein, dass ein gezielter Angriff gegen derart gesicherte Netzwerke für einen versierten Angreifer kein Problem darstellt.

In der heutigen Zeit werden genau diese drei Kostenpositionen bei der Berechnung für den Betrieb eines Netzwerkes veranschlagt. Zu dem Netzwerk sind dabei sukzessive ein Management und verschiedene Sicherheitsmechanismen hinzugefügt worden. Leider wird dabei aber häufig gerade die Sicherheit des Netzwerkmanagements vernachlässigt. Ein deutliches Zeichen dafür ist die zu langsam sich ausbreitende Unterstützung moderner Sicherheitsmechanismen wie beispielsweise beim SNMPv3 Protokoll oder den SSH Verbindungen. Gerade das Netzwerkmanagement bietet einem Angreifer aber ein lohnendes Ziel. Hier kann er mit minimalem Aufwand einen maximalen Vorteil gewinnen. In den Kostenrechnungen sollte daher dieser Punkt entsprechend berücksichtigt werden.

12.3 Einfluss des Netzwerkmanagements auf das Netzwerk

Neben den bisher aufgeführten Kosten existiert noch ein weiterer Punkt, der in der Bilanzierung des Netzwerks zu berücksichtigen ist. Gemeint ist der negative Einfluss des Netzwerkmanagements auf das Netzwerk selber. Selbstverständlich besitzt ein gutes Netzwerkmanagement immer einen deutlichen positiven Einfluss auf das Netzwerk, da beispielsweise Fehler schneller identifiziert und behoben werden können oder da Dank besserer Planungsmöglichkeiten Engpässe schon im Vorfeld identifiziert und durch rechtzeitige Reaktionen verhindert werden können. Allerdings hat das Netzwerkmanagement immer auch einen negativen Einfluss auf das Netzwerk, was letztendlich in Form von gesteigerten Kosten deutlich wird.

Gerade bei der Implementierung eines Out-of-Band Managements leuchtet sofort ein, dass durch die redundanten Übertragungswege nicht nur höhere Anschaffungskosten anfallen, sondern es entstehen auch höhere Wartungskosten

für die zusätzliche Hardware und die neue Software. Steigt die Komplexität des Netzwerkes und damit auch die Komplexität des redundanten Managementnetzwerkes, so steigt zusätzlich noch die Notwendigkeit des „Selbstmanagements“. Damit soll gemeint sein, dass ein paralleles Netzwerk, das nur dem Management eines Hauptnetzwerkes dient, ab einer bestimmten Größe selbst ein Management benötigt. Auf diese Weise steigt der Overhead bei den Kosten für das Netzwerkmanagement, was sich schließlich nachteilig auf das gesamte Netzwerk auswirkt.

Bei einem In-Band Management wiederum fallen zwar auf den ersten Blick keine Kosten für eine zusätzliche redundante Infrastruktur an, jedoch verbraucht das Management immer einen gewissen Anteil der im Netzwerk zur Verfügung stehenden Bandbreite. Die unmittelbare Folge daraus ist nicht nur ein erhöhter Bedarf an Netzwerkkapazität, sondern durch die Nutzung des Hauptnetzwerkes für das Management ändern sich auch die Anforderungen an die Zuverlässigkeit des Netzwerkes. Schließlich soll das Management auch noch unter ungünstigen Bedingungen arbeiten. Als Folge daraus entsteht wiederum ein erhöhter Bedarf an Redundanz, der sich in zusätzlichen Kosten niederschlägt.

Dazu schlagen in allen Fällen noch die Kosten für die Absicherung des Netzwerkmanagements und deren Kommunikationswege zu Buche. Unter dem Strich bleibt allerdings festzuhalten, dass in den meisten Fällen der Einsatz eines Sicheren Netzwerkmanagements dennoch ein positives Kosten/Nutzen Verhältnis liefert. Die Herausforderung besteht in der möglichst genauen Abschätzung des Nutzens, um das ideale Maß an Sicherheit implementieren zu können.

13

Neue Entwicklungen

Der Fortschritt in der Informationstechnologie ist nicht aufzuhalten und ständig werden neue Technologien, Protokolle, Mechanismen und Anwendungen entwickelt. Das schließt selbstverständlich auch die Bereiche des Netzwerkmanagements und der IT Sicherheit ein. Daher müssen nicht nur jetzt, sondern auch in Zukunft die Netzwerkbetreiber und Administratoren ständig ihr Wissen auf dem Laufenden halten und die neuen Entwicklungen einsetzen und zu ihrem Vorteil nutzen. Ein deutliches Zeichen für den fortschreitenden Wandel sind beispielsweise die „Next Generation" Protokolle wie IPng oder SMIng. Noch haben sich diese Protokolle nicht durchgesetzt, aber früher oder später werden Vorteile der neuen Technologien siegen und die älteren, jetzt noch aktuellen Protokolle werden abgelöst. Vielleicht passiert das nicht heute, und vielleicht werden die zukunftsweisenden Technologien bis dahin noch einmal überarbeitet, aber die Ablösung der heutigen Mechanismen lässt sich langfristig nicht verhindern. Im Folgenden soll ein Ausblick auf Bereiche gewagt werden, die sich in der näheren Zukunft auf die heutigen Techniken auswirken können.

13.1 Mobile Geräte – Andere Verhaltensprofile

Ein bereits heute erkennbarer Trend ist die verstärkte Mobilisierung der Kommunikationssysteme. Früher waren Rechner statische Geräte, die bedenkenlos nach ihrem Standort kategorisiert werden konnten. Im Laufe der Zeit haben sich aber die Client-Rechner zu mobilen Systemen entwickelt. Viele der früheren Desktops[1] werden heute durch die flexibleren Laptops[2] ersetzt. Die vielen Vorteile der Mobilität haben aber auch ihre Auswirkungen auf das

[1] Der englische Begriff ‚Desktop' bedeutet übersetzt so viel wie ‚auf dem Schreibtisch'.

[2] Im Gegensatz zum ‚Desktop' bedeutet der englische Begriff ‚Laptop' so viel wie ‚auf dem Schoß'.

Netzwerkmanagement und in hohem Maße auch auf die Sicherheit des gesamten Netzwerks. Tragbare Computer können an einem Tag mühelos Teil von mehreren verschiedenen Netzwerken werden. Dabei sind sie auch unterschiedlichsten Bedrohungen ausgesetzt. Der Schutz der Endgeräte ließ sich früher in großem Maße am Zugangspunkt des Intranets zum Internet durchführen. Dort konnte ein großer Teil der Bedrohungen bereits abgefangen und neutralisiert werden. Die mobilen Geräte von heute ändern teilweise ständig ihren Aufenthaltsort und die Netzwerkumgebung, so dass die Schutzmechanismen nun vollständig auf die Endgeräte verlagert werden müssen. An dieser Stelle sei ein Vergleich der Computerviren mit den biologischen Viren erlaubt. In gleichem Maße, wie Flughäfen sowohl die Mobilität der Menschen als auch die Ausbreitungsgeschwindigkeit von Virenerkrankungen erhöht haben, erhöhen auch neue Technologien sowohl die Mobilität der Computer als auch die Ausbreitungsgeschwindigkeit von Computerviren. Befinden sich nun ständig Geräte im eigenen Netz, die potentielle Bedrohungen mitbringen, so spielt der Schutz des eigenen Netzwerkes und dessen Management eine immer größere Rolle. Die Anforderungen an den Schutz des Netzwerkmanagements vor den Gefahren aus dem eigenen Netzwerk nähern sich stetig den Anforderungen an den Schutz von Bedrohungen von außen an. Gleichzeitig verleiten die neuen Möglichkeiten zu einem leichtfertigen Umgang mit der Sicherheit. Wenn etwa Managementwerkzeuge die Möglichkeit zur Administration des Netzwerk von jedem beliebigen Ort der Welt aus erlauben, dann sollte man sich zunächst Gedanken darüber machen, welche Auswirkungen das auf die Sicherheit hat. Wenn ein Administrator sein Netzwerk von einem Internetcafé an seinem Urlaubsort aus bedienen kann, dann sollte er auch bedenken, dass die Geräte in diesem Internetcafé als äußert unsicher einzustufen sind. Nicht selten finden sich auf derart öffentlichen Systemen Key-Logger, mit denen Angreifer eingegebene Passwörter abfangen können.

13.2 Leistungsstärkere Rechner – Höherer Schutzaufwand

Die Bemühungen der Organisation DISTRIBUTED.NET, durch simples Ausprobieren aller möglichen Schlüssel kodierte Nachrichten zu entschlüsseln, zeigen heute erste Erfolge im Brechen von Sicherheitsmechanismen. Zwar werden zum Knacken von Schlüsseln mit einer Schlüssellänge von 64 Bit aktuell noch mehrere Jahre benötigt, das exponentielle Wachstum der Rechenleistungen von Computern in Verbindung mit dem Anstieg der Gesamtzahl aller Rechner lässt aber auch hier mittelfristig eine deutliche Beschleunigung erwarten. Gleichzeitig verlieren heutige Verschlüsselungsalgorithmen an Sicherheit. Beispielsweise konnte für den bis heute als sicher geltenden Verschlüsselungsalgorithmus SHA-1 eine Hashwert-Kollision nach nur 2^{63} Operationen ermittelt werden [222]. Bereits heute wäre es also möglich, zu einer mit SHA-1 signierten Nachricht in etwa zwei Jahren eine andere Nachricht zu finden, die densel-

ben Hashwert aufweist. Eine weitere Leistungssteigerung der Rechner würde diesen Zeitraum noch einmal signifikant nach unten korrigieren können. Der Umstieg auf sicherere Algorithmen ist damit unumgänglich.

A

Request For Comments für das Simple Network Management Protocol

A.1 SNMP

Tabelle A.1. RFC Definitionen der verschiedenen Versionen des Simple Network Management Protocols.

Version	RFCs
SNMP (SNMPv1)	1067 1098 1157
SNMPv2p	1441 1445 1446
SNMPv2u	1910
SNMPv2*	—
SNMPv2c (SNMPv2)	1901
SNMPv3	(2261-2265) (2271-2275) (2570-2575) 3410-3417 + 3430

A.2 MIB

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur.

MIB-Typ	RFCs
Management Information Base for Network Management of TCP/IP-based Internets (MIB-I)	(1066) 1156
Management Information Base for Network Management of TCP/IP-based Internets: MIB-II	(1158) **1213**

(Fortsetzung auf nächster Seite)

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur. (Fortsetzung)

MIB-Typ	RFCs
CLNS MIB for use with Connectionless Network Protocol (ISO 8473) and End System to Intermediate System (ISO 9542)	(*1162*) *1238*
OSI Internet Management	1214
The Interfaces Group MIB	(1229) (1573) (2233) **2863**
IEEE 802.4 Token Bus MIB	1230
IEEE 802.5 MIB using SMIv2	(1231) (**1743**) **1748**
Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types	(1232) (1406) (2495) 3895
Definitions of Managed Objects for the DS3/E3 Interface Type	(1233) (1407) (2496) 3896
AppleTalk Management Information Base II	(1243) 1742
OSPF Version 2 Management Information Base	(1248) (1252) (1253) **1850**
Definitions of Managed Objects for the Border Gateway Protocol: Version 3	1269
Remote Network Monitoring Management Information Base	(1271) (**1757**) **2819**
Definitions of Managed Objects for the Ethernet-like Interface Types	(1284) (**1398**) (**1623**) (1643) 3638
FDDI Management Information Base	1285 + 1512
Definitions of Managed Objects for Bridges	(1286) **1493** 4188
Definitions of Managed Objects for Source Routing Bridges	(1286) 1525
DECnet Phase IV MIB Extensions	(1289) **1559**
Definitions of Managed Objects for SMDS Interfaces using SMIv2	(1304) **1694**
Management Information Base for Frame Relay DTEs Using SMIv2	(1315) **2115**
Definitions of Managed Objects for Character Stream Devices using SMIv2	(1316) **1658**
Definitions of Managed Objects for RS-232-like Hardware Devices using SMIv2	(1317) **1659**
Definitions of Managed Objects for Parallel-Printer-like Hardware Devices using SMIv2	(1318) **1660**
Definitions of Managed Objects for Administration of SNMP Parties	1353
IP Forwarding Table MIB	(1354) 2096
Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIv2	(1368) (**1516**) 2108
SNMP MIB Extension for X.25 LAPB	1381

(Fortsetzung auf nächster Seite)

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur. (Fortsetzung)

MIB-Typ	RFCs
SNMP MIB Extension for the X.25 Packet Layer	1382
RIP Version 2 MIB Extension	(1389) **1724**
Identification MIB	1414
Manager-to-Manager Management Information Base	1451
SNMP MIB extension for Multiprotocol Interconnect over X.25	1461
The Definitions of Managed Objects for the Link Control Protocol of the Point-to-Point Protocol	1471
The Definitions of Managed Objects for the Security Protocols of the Point-to-Point Protocol	1472
The Definitions of Managed Objects for the IP Network Control Protocol of the Point-to-Point Protocol	1473
The Definitions of Managed Objects for the Bridge Network Control Protocol of the Point-to-Point Protocol	1474
Host Resources MIB	(1514) **2790**
Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)	(1515) (2239) (2668) 3636
Network Services Monitoring MIB	(1565) (2248) 2788
Mail Monitoring MIB	(1566) (2249) 2789
Directory Server Monitoring MIB	(1567) 2605
Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type	(1595) (2558) **3592**
Definitions of Managed Objects for Frame Relay Service	(1596) (1604) 2954
DNS Server MIB Extensions	1611
DNS Resolver MIB Extensions	1612
UPS Management Information Base	1628
Definitions of Managed Objects for the Ethernet-like Interface Types	(1650) (2358) (2665) 3635
Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2	**1657**
Definitions of Managed Objects for SNA NAUs using SMIv2	(1665) 1666
Definitions of Managed Objects for ATM Management	(1695) 2515

(Fortsetzung auf nächster Seite)

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur. (Fortsetzung)

MIB-Typ	RFCs
Modem Management Information Base (MIB) using SMIv2	1696
Relational Database Management System (RDBMS) Management Information Base (MIB) using SMIv2	1697
Definitions of Managed Objects for SNA Data Link Control (SDLC) using SMIv2	1747
IEEE 802.5 Station Source Routing MIB using SMIv2	1749
Printer MIB v2	(1759) 3805
TCP/IPX Connection Mib Specification	*1792*
The Definitions of Managed Objects for IP Mobility Support using SMIv2	2006
SNMPv2 Management Information Base for the Internet Protocol using SMIv2	2011
Management Information Base for the Transmission Control Protocol (TCP)	(2012) 4022
SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2	(2013) 4113
IEEE 802.12 Interface MIB	2020
Remote Network Monitoring Management Information Base Version 2 using SMIv2	2021
Definitions of Managed Objects for Data Link Switching using SMIv2	2024
Entity MIB (Version 3)	(2037) (2737) 4133
Definitions of Managed Objects for APPC using SMIv2	2051
Traffic Flow Measurement: Meter MIB	(*2064*) 2720
Remote Network Monitoring MIB Protocol Identifier Reference	(2074) **2895** + 3395
ISDN Management Information Base using SMIv2	2127
Dial Control Management Information Base using SMIv2	2128
Definitions of Managed Objects for APPN	(2155) 2455
RSVP Management Information Base using SMIv2	2206
Integrated Services Management Information Base using SMIv2	2213

(Fortsetzung auf nächster Seite)

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur. (Fortsetzung)

MIB-Typ	RFCs
Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2	2214
Definitions of Managed Objects for DLUR using SMIv2	2232
Definitions of Managed Objects for HPR using SMIv2	2238
Definitions of Managed Objects for IEEE 802.12 Repeater Devices	2266
Definitions of System-Level Managed Objects for Applications	2287
Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA-MIB)	2320
Definitions of Managed Objects for Multicast over UNI 3.0/3.1 based ATM Networks	(2366) 2417
IP Version 6 Management Information Base for the User Datagram Protocol	(2454) 4113
Definitions of Managed Objects for APPN TRAPS	2456
Definitions of Managed Objects for Extended Border Node	2457
Management Information Base for IP Version 6: Textual Conventions and General Group	2465
Management Information Base for IP Version 6: ICMPv6 Group	2466
Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals	(2493) **3593**
Definitions of Managed Objects for the DS0 and DS0 Bundle Interface Type	2494
Managed Objects for Controlling the Collection and Storage of Accounting Information for Connection-Oriented Networks	2513
Base Definitions of Managed Objects for TN3270E Using SMIv2	2561
Definitions of Protocol and Managed Objects for TN3270E Response Time Collection Using SMIv2 (TN3270E-RT-MIB)	2562
Application Management MIB	2564

(Fortsetzung auf nächster Seite)

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur. (Fortsetzung)

MIB-Typ	RFCs
Definitions of Managed Objects for APPN/HPR in IP Networks	2584
Definitions of Managed Objects for Scheduling Management Operations	(2591) 3231
Definitions of Managed Objects for the Delegation of Management Scripts	(2592) 3165
Script MIB Extensibility Protocol Version 1.1	(*2593*) *3179*
Definitions of Managed Objects for WWW Services	2594
Remote Network Monitoring MIB Extensions for Switched Networks Version 1.0	2613
RADIUS Authentication Client MIB	2618
RADIUS Authentication Server MIB	2619
RADIUS Accounting Client MIB	2620
RADIUS Accounting Server MIB	2621
Definitions of Managed Objects for the ADSL Lines	2662
IP Tunnel MIB	(2667) 4087
DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems	2669
Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces	2670
Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions	2674
Definitions of Managed Objects for the NBMA Next Hop Resolution Protocol (NHRP)	2677
Definitions of Managed Objects for Extensible SNMP Agents	**2742**
Definitions of Managed Objects for Service Level Agreements Performance Monitoring	*2758*
Diffie-Helman USM Key Management Information Base and Textual Convention	*2786*
Definitions of Managed Objects for the Virtual Router Redundancy Protocol	2787
Definitions of Managed Objects for the Fabric Element in Fibre Channel Standard	(2837) 4044

(Fortsetzung auf nächster Seite)

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur. (Fortsetzung)

MIB-Typ	RFCs
The Inverted Stack Table Extension to the Interfaces Group MIB	2864
Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations	2925
IPv4 Multicast Routing MIB	2932
Internet Group Management Protocol MIB	2933
Protocol Independent Multicast MIB for IPv4	*2934*
Definitions of Managed Objects for Common Open Policy Service (COPS) Protocol Clients	2940
Definitions of Managed Objects for Monitoring and Controlling the Frame Relay/ATM PVC Service Interworking Function	2955
Real-Time Transport Protocol Management Information Base	2959
Event MIB	2981
Distributed Management Expression MIB	2982
Notification Log MIB	3014
IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol	3019
Definitions of Managed Objects for Monitoring and Controlling the UNI/NNI Multilink Frame Relay Function	3020
Management Information Base for the PINT Services Architecture	3055
Remote Monitoring MIB Extensions for Interface Parameters Monitoring	3144
Definitions of Managed Objects for Circuit to Interface Translation	3201
Definitions of Managed Objects for Frame Relay Service Level Definitions	3202
Remote Network Monitoring Management Information Base for High Capacity Networks	3273
Definitions of Managed Objects for High Bit-Rate DSL - 2nd generation (HDSL2) and Single-Pair High-Speed Digital Subscriber Line (SHDSL) Lines Processing	3276
Remote Monitoring MIB Extensions for Differentiated Services	3287
Management Information Base for the Differentiated Services Architecture	3289

(Fortsetzung auf nächster Seite)

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur. (Fortsetzung)

MIB-Typ	RFCs
Definitions of Managed Objects for the General Switch Management Protocol (GSMP)	3295
Layer Two Tunneling Protocol „L2TP“ Management Information Base	3371
Entity Sensor Management Information Base	3433
Remote Monitoring MIB Extensions for High Capacity Alarms	3434
Definitions of Extension Managed Objects for Asymmetric Digital Subscriber Lines	3440
Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures	3498
Multicast Address Allocation MIB	3559
Definitions of Managed Objects for the Optical Interface Type	3591
Definitions of Supplemental Managed Objects for ATM Interface	3606
Power Ethernet MIB	3621
Definitions of Managed Objects for the Ethernet WAN Interface Sublayer	3637
High Capacity Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals	3705
Definitions of Managed Objects for Very High Speed Digital Subscriber Lines (VDSL)	3728
Application Performance Measurement MIB	3729
The Differentiated Services Configuration MIB	3747
Printer Finishing MIB	3806
Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)	3812
Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)	3813
Multiprotocol Label Switching (MPLS) Forwarding Equivalence Class To Next Hop Label Forwarding Entry (FEC-To-NHLFE) Management Information Base (MIB)	3814

(Fortsetzung auf nächster Seite)

Tabelle A.2. Management Information Bases definiert von der IETF. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard. RFCs in *Kursivschrift* sind experimenteller Natur. (Fortsetzung)

MIB-Typ	RFCs
Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)	3815
Definitions of Managed Objects for RObust Header Compression (ROHC)	3816
Management Information Base for Telephony Routing over IP (TRIP)	3872
Alarm Management Information Base (MIB)	3877
Alarm Reporting Control Management Information Base (MIB)	3878
A Traffic Engineering (TE) MIB	3970
Policy Based Management MIB	4011
Management Information Base for Data Over Cable Service Interface Specification (DOCSIS) Cable Modem Termination Systems for Subscriber Management	4036
Definitions of Managed Object Extensions for Very High Speed Digital Subscriber Lines (VDSL) Using Single Carrier Modulation (SCM) Line Coding	4069
Definitions of Managed Object Extensions for Very High Speed Digital Subscriber Lines (VDSL) Using Multiple Carrier Modulation (MCM) Line Coding	4070
Management Information Base for Data Over Cable Service Interface Specification (DOCSIS) Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus	4131
Definition of Managed Objects for Synthetic Sources for Performance Monitoring Algorithms	4149
Transport Performance Metrics MIB	4150

A.3 SMI

Tabelle A.3. Structure of Management Information Definitionen der IETF. RFCs in runden Klammern sind veraltet.

Version	RFCs
SMI	(1065) 1155 + 1212, 1215
SMIv2	(1442) (1902) 2578 + 2579, 2580
SMIng	(3216) 3780, 3781

A.4 RMON

Tabelle A.4. Management Information Base Definitionen für das Remote Network Monitoring. RFCs in runden Klammern sind veraltet. RFCs in **Fettschrift** gelten aktuell als Standard.

Remote Network Monitoring MIB	RFCs
RMONv1	(1271) (**1757**) **2819**
Token Ring Extensions to the RMON MIB	1513
RMONv2	2021
RMON MIB Protocol Identifier Reference	(2074) **2895** + 3395
RMON MIB Extensions for Switched Networks Version 1.0	2613
RMON MIB Protocol Identifier Macros	2896
RMON MIB Extensions for Interface Parameter Monitoring	3144
RMON MIB for High Capacity Networks	3273
RMON MIB Extensions for Differentiated Services	3287
RMON MIB Extensions for High Capacity Alarms	3434
Introduction to the RMON Family of MIB Modules	3577
Application Performance Measurement MIB	3729
IANA Guidelines for the Registry of RMON MIB Modules	3737
RMON Protocol Identifiers for IPv6 and MPLS	3919

B

IPMI-konforme Hersteller

Zur Zeit der Erstellung dieser Liste (August 2005) zählte die Liste der IPMI-konformen Hersteller von Hardware und Software beinahe 200 Unternehmen aus der ganzen Welt. Als Quelle wurde die „IPMI Adopter List“ auf den Webseiten der Intel Corporation [97] verwendet:

- ABIT Computer Corp.
- Acer Incorporated
- Acxiom Corporation
- ADTRON
- Advanced Micro Devices, Inc.
- Agilent Technologies GmbH
- Alberta Microelectronic Corporation
- Alliance Semiconductor Corporation
- Allion Computer Inc.
- American Megatrends Incorporated
- Amphus Inc.
- Ample Communications Inc.
- Arima Computer Corp.
- Artesyn Communication Products, Inc.
- ASIS LTD.
- ASUSTek Computer Incorporated
- Aventail Corporation
- Avian Communications
- Avocent Corporation
- Axil Computer, Inc.
- Blue Wave Systems
- Bull S.A.
- C&D Technologies, Inc.
- California Digital Corp.
- Celestica
- ColoWATCH, Inc

- Communication Automation Corporation
- Concurrent Technologies PLC
- Compellent Technologies, Inc.
- CorEdge Networks Inc.
- C-MAC Engineering
- CyberGuard Corporation
- Cyclades Corporation
- Data General Corporation
- Decru, Inc.
- Dell Computer Corporation
- Demac Associates
- Digi International
- Egenera, Inc.
- ElanVital Corporation
- Ericsson UAB
- ESO Technologies
- Evans & Sutherland
- Eversys Corporation
- Exabyte Corporation
- Extreme Engineering Solutions, Inc. (XES)
- Fabric7 Systems, Inc.
- First International Computer, Inc.
- Flextel SpA
- FORCE Computers GmbH
- Forward Technologies
- Freedom Technologies Corporation (FreeBSD Systems)
- FreeIPMI Core Team
- Fujitsu Limited
- Fujitsu Siemens Computers
- Gambit Communications
- Gluon Networks, Inc.
- GoAhead Software, Inc.
- HADCO Corporation
- HCL Infosystems Ltd.
- Hewlett-Packard Company
- Hewlett-Packard GmbH
- Hitachi Ltd.
- Hybricon Corporation
- IBM Corporation
- I-Bus/Phoenix Corporation
- ICP Electronics Inc.
- InnoMediaLogic Inc
- Integra Micro Systems (P) Ltd.
- Intel Corporation
- Interphase Corporation

- InterWorks Computer Products
- Inventec Corporation
- Ipex ITG
- JMC PRODUCTS
- Kaparel Corporation
- Katana Technology, Inc.
- Kealia, Inc.
- L-3 Communications Corporation
- LANDesk Software
- LANTRONIX
- Legend (Beijing) Limited
- LeoStream Corp.
- Linux NetworX, Inc.
- Lynux Works, Inc.
- Macrolink, Inc.
- MagneTek, Inc.
- Marvell International Ltd.
- MEGWARE Computer GmbH
- Mercury Computer Systems, Inc.
- Microsoft
- Micro-Star Int'l
- Mirapoint, Inc.
- MiTAC International Corp.
- Mitsubishi Electric Corporation, Information Systems Engineering Center
- Motorola Computer Group
- National Semiconductor Corporation
- NEC Corporation
- Nematron Corporation
- Network Appliance, Inc.
- Network Engines, Inc.
- Network Storage Solutions, Inc.
- NEWISYS, Inc.
- NOCpulse, Inc.
- Novell, Inc.
- Olidata S.p.A
- Olivetti Computers Worldwide
- Open Source Development Lab
- Opus Innovations LLC
- OSA Technologies
- PEP Modular Computers
- Peppercon AG
- PetaStream Inc.
- Performance Technologies, Inc.
- PFU Limited
- Phoenix Technologies Ltd.

- Pigeon Point Systems
- Pinnacle Data Systems, Inc.
- Praim, Inc.
- Qlogic Corporation
- Quanta Computer Inc.
- RadiSys Corporation
- RADVISION
- RAMIX Inc.
- Raritan Computer, Inc.
- Reliance Computer Corporation
- RLX Technologies, Inc.
- Samsung Electronics Co., LTD
- Sanera Systems, Inc.
- SANgate Systems, Inc.
- Sanritz Automation Co., Ltd.
- SBS Technologies, Inc.
- Scenix Semiconductor, Inc.
- Sena Technologies Inc.
- Sentry Software
- ServerEngines LLC
- Siemens Nixdorf Informationssysteme AG
- Silicon Graphics, Inc.
- SKY Computers, Inc
- SMIS R&D INC.
- Snap Appliance
- Stan Cox & Associates
- Standard Microsystems Corporation
- StrataLight Communications, Inc.
- Stratus Computer Systems Ireland Ltd.
- Summit Microelectronics, Inc.
- Sun Microsystems
- Super Micro Computer, Inc.
- SyAM Software Inc.
- Symbium Corporation
- Symphony Group International Co., Ltd
- Synergy Microsystems
- T-NETIX Inc.
- TATUNG Co.
- Technobox, Inc.
- Teknor Applicom Inc.
- Tektronix
- Texas Micro Corporation
- Togabi Technologies Inc.
- Toshiba Corporation
- Trilogic Systems, LLC

- Trimm Technologies
- Tyan Computer Corporation
- Unisys Corporation
- Universal Scientific Industrial Corporation
- University of New Hampshire - Interoperability Laboratory
- USAR Systems Inc.
- VadaTech Inc.
- VIA Technologies, Inc.
- Vitesse Semiconductor Corporation
- Vividon, Inc.
- Vooha Inc.
- Watrin System Design
- Wellsyn Technology, Inc.
- Winbond Electronics Corporation
- Wind River Systems
- WIPRO INFOTECH
- Wistron Corporation
- Wyselec
- Xiotech Corporation
- Ziatech Corporation
- ZNYX Networks, Inc.

C

Verzeichnis verwendeter Abkürzungen

Tabelle C.1. Verzeichnis der in diesem Buch verwendeten Abkürzungen mit ihren Erläuterungen.

AAA	Authentication, Authorization, and Accounting
ACCM	Async-Control-Character-Maps
ACL	Access Control List
ACPI	Advanced Configuration and Power Interface
API	Application Program Interface
ARP	Address Resolution Protocol
AS	Autonomous System
ASCII	American Standard Code for Information Interchange
ASF	Alerting Standard Formats
ASIC	Application-Specific Integrated Circuit
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
BEEP	Blocks Extensible Exchange Protocol
BER	Basic Encoding Rules
BIOS	Basic Input/Output System
BMC	Baseboard Management Controller
BOOTP	Bootstrap Protocol
BSD	Berkley Software Distribution
BT	Block Transfer
CA	Certification Authority
CBC	Cipher Block Chaining
CBC-DES	Cipher Block Chaining Data Encryption Standard
CBCP	Callback Control Protocol
CERT/CC	Computer Emergency Response Team / Coordination Center
CHAP	Challenge Handshake Authentication Protocol
CIDR	Classless Inter-Domain Routing

CIM	Common Information Model
CLTS	Connectionless-Mode Transport Service
COPS-PR	Common Open Policy Service Provisioning
CSV	Comma Separated Values
DARPA	Defense Advanced Research Projects Agency
DDoS	Distributed Denial of Service
DDP	Datagram Delivery Protocol
DES	Data Encryption Standard
DES3	Triple Data Encryption Standard
DF	Don't fragment
DH	Diffie-Hellman
DHCP	Dynamic Host Control Protocol
DMTF	Distributed Management Task Force
DNS	Domain Name Service
DoD	Department of Defense
DoS	Denial of Service
DS	Differentiated Services
EAP	Extensible Authentication Protocol
EAP-AKA	Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement
EAP-FAST	Extensible Authentication Protocol – Flexible Authentication Via Secure Tunneling
EAP-GTC	Extensible Authentication Protocol – Generic Token Card
EAP-MD5	Extensible Authentication Protocol – Message Digest Number 5
EAP-MSCHAPv2	Extensible Authentication Protocol – Microsoft Challenge Handshake Authentication Protocol Version 2
EAP-OTP	Extensible Authentication Protocol – One Time Password
EAP-SIM	Extensible Authentication Protocol – Subscriber Identification Module
EAP-TLS	Extensible Authentication Protocol – Transport Layer Security
EAP-TTLS	Extensible Authentication Protocol Tunneled Transport Layer Security
EAPOL	Extensible Authentication Protocol Over Local Area Network
ECN	Explicit Congestion Notification
EGP	Exterior Gateway Protocol
FBI	Federal Bureau of Investigation
FIFO	First-In-First-Out
FPGA	Field-Programmable Gate Array
FQDN	Full Qualified Domain Name

FRU	Field Replacable Unit
FTP	File Transfer Protocol
GGP	Gateway-to-Gateway Protocol
GMT	Greenwich Mean Time
GNU	GNU's not Unix
GSSAPI2	Generic Security Service Application Program Interface Version 2
HDLC	High-Level Data Link Control
HMAC	Hash-based Message Authentication Code
HSRP	Hot Standby Router Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
I^2C	Inter Integrated Circuit
IANA	Internet Assigned Numbers Authority
ICMB	Intelligent Chassis Management Bus
ICMP	Internet Control Message Protocol
IDRS	Intrusion Detection/Response System
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineering
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IIS	Internet Information Server
IMAP	Internet Message Access Protocol
I/O	Input/Output
IOS	(Cisco) Internet Operation System
IP	Internet Protocol
IPC	Interprocess Communications
IPMB	Intelligent Platform Management Bus
IPMI	Intelligent Platform Management Interface
IPng	Internet Protocol Next Generation
IPSec	IP Security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IPX	Internetwork Packet Exchange
IRTF	Internet Research Task Force
IRTP	Internet Reliable Transaction Protocol
ISC	Internet Storm Center
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Informationstechnologie
IV	Initialization Vector
J2EE	Java 2 Platform, Enterprise Edition
KCS	Keyboard Controller Style

KVM	Keyboard-Video-Mouse
L2TP	Layer Two Tunneling Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LEAP	Lightweight Extensible Authentication Protocol
MAC	Media Access Control
MAU	Medium Attachment Unit
MD4	Message Digest 4
MD5	Message Digest 5
MIB	Management Information Base
MIB-I	Management Information Base Version 1
MIB-II	Management Information Base Version 2
MPLS	Multiprotocol Label Switching
MSCHAP	Microsoft Challenge Handshake Authentication Protocol
MSCHAPv2	Microsoft Challenge Handshake Authentication Protocol Version 2
MTU	Maximum Transfer Unit
MUA	Mail User Agent
NAT	Network Address Translation
NBP	Name Binding Protocol
NetBIOS	Network Basic Input/Output System
NMRG	Network Management Research Group
NNTP	Network News Transfer Protocol
NOP	No Operation
NTP	Network Time Protocol
ODBC	Open Data Base Connectivity
OID	Object Identifier
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSPF 2	Open Shortest Path First Version 2
PAM	Pluggable Authentication Modules
PAP	Password Authentication Protocol
PCI	Peripheral Component Interconnect
PDA	Personal Digital Assistent
PDF	Portable Document Format
PDU	Protocol Data Unit
PEAP	Protected Extensible Authentication Protocol
PEF	Platform Event Filtering
PGP	Pretty Good Privacy
PID	Process Identification Number
PKI	Public Key Infrastructure
PMTU	Path Maximum Transmission Unit
PNG	Portable Network Graphics

POP3	Post Office Protocol Version 3
PPP	Point-to-Point Protocol
RADIUS	Remote Authentication Dial In User Service
RAKP	Remote Management Control Protocol+ Authenticated Key-Exchange Protocol
RC2	Rivest Cipher Version 2
RC4	Rivest Cipher Version 4
RC5	Rivest Cipher Version 5
RCP	remote copy
RDIST	remote distribution
RDP	Reliable Data Protocol
RFC	Request for Comments
RIP	Routing Information Protocol
RIP-2	Routing Information Protocol Version 2
RLOGIN	remote login
RMCP	Remote Management Control Protocol
RMCP+	Remote Management Control Protocol+
RMON	Remote Monitoring
RMONv1	Remote Monitoring Version 1
RMONv2	Remote Monitoring Version 2
RSA	Rivest, Shamir, Adleman
RSH	remote shell
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time
SASL	Simple Authentication and Security Layer
SCP	secure copy
SDR	Sensor Data Record
SEI	Software Engineering Institute
SEL	System Event Log
SFTP	Simple File Transfer Protocol
SFTP	secure file transfer program
SHA-1	Secure Hash Algorithm 1
SIK	Session Integrity Key
SMI	Structure of Management Information
SMI1	Structure of Management Information Version 1
SMI2	Structure of Management Information Version 2
SMIng	Structure of Management Information Next Generation
SMIC	System Management Interface Chip
S/MIME	Secure Multipurpose Internet Mail Extensions
SMS	Short Message System
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol

SNMPv1	Simple Network Management Protocol Version 1
SNMPv2	Simple Network Management Protocol Version 2
SNMPv2*	Simple Network Management Protocol Version 2 Star
SNMPv2c	Community-Based Simple Network Management Protocol Version 2
SNMPv2p	Simple Network Management Protocol Version 2 Party
SNMPv2u	User-Based Security Model for Simple Network Management Protocol Version 2
SNMPv3	Simple Network Management Protocol Version 3
SOL	Serial Over Local Area Network
SPPI	Structure of Policy Provisioning Information
SQL	Structured Query Language
SSH	secure shell
SSL	Secure Socket Layer
ST2	Internet Stream Protocol Version 2
TACACS	Terminal Access Controller Access Control System
TACACS+	Terminal Access Controller Access Control System+
TAP	Telocator Alphanumeric Protocol
Tcl	Tool Command Language
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
ToS	Type of Service
TTL	Time-to-Live
UCD	University of California, Davis
UDP	User Datagram Protocol
USM	User-Based Security Model
USV	Unterbrechungsfreien Stromversorgung
UTC	Universal Time Constant
UTF-8	Universal Character Set Transformation Format 8
UUCP	Unix-to-Unix Communications Package
VACM	View-based Access Control Model
VLAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAP	Wireless Application Protocol
WEP	Wired Equivalent Privacy
WINS	Windows Internet Naming Service
WLAN	Wireless Local Area Network
WPA	Wireless-Fidelity Protected Access
XML	Extensible Markup Language

Literaturverzeichnis

1. Bernard Aboba, Larry Blunk, John Vollbrecht, James Carlson, and Henrik Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748, Internet Engineering Task Force, June 2004.
2. Bernard Aboba and Dan Simon. PPP EAP TLS Authentication Protocol. RFC 2716, Internet Engineering Task Force, October 1999.
3. Philip Almquist. Type of Service in the Internet Protocol Suite. RFC 1349, Internet Engineering Task Force, July 1992.
4. Anonymous Digital Coalition. Call for Virtual Sit-Ins at five Mexico Financial Web Sites. `http://www.thing.net/~rdom/ecd/anondigcoal.html`.
5. ANSI. Telecommunications Integrated Services Digital Network (ISDN) - Core Aspects of Frame Protocol for use with Frame Relay Bearer Service. ANSI Standard T1.618-1991, American National Standards Institute, 1991.
6. Asynchronous Transfer Mode (ATM). `http://www.atmforum.com`.
7. Fred Baker. IP Forwarding Table MIB. RFC 1354, Internet Engineering Task Force, July 1992.
8. Fred Baker. Requirements for IP Version 4 Routers. RFC 1812, Internet Engineering Task Force, June 1995.
9. Fred Baker. IP Forwarding Table MIB. RFC 2096, Internet Engineering Task Force, January 1997.
10. Fred Baker and Rob Coltun. OSPF Version 2 Management Information Base. RFC 1248, Internet Engineering Task Force, July 1991.
11. Fred Baker and Rob Coltun. OSPF Version 2 Management Information Base. RFC 1850, Internet Engineering Task Force, November 1995.
12. Andy Bierman. Remote Monitoring MIB Extensions for Differentiated Services. RFC 3287, Internet Engineering Task Force, July 2002.
13. Andy Bierman, Chris Bucci, Russell Dietz, and Albin Warth. Remote Network Monitoring MIB Protocol Identifier Reference Extensions. RFC 3395, Internet Engineering Task Force, September 2002.
14. Andy Bierman, Chris Bucci, and Ribon Iddon. Remote Network Monitoring MIB Protocol Identifier Macros. RFC 2896, Internet Engineering Task Force, August 2000.

15. Andy Bierman, Chris Bucci, and Ribon Iddon. Remote Network Monitoring MIB Protocol Identifier Reference. RFC 2895, Internet Engineering Task Force, August 2000.
16. Andy Bierman and Keith McCloghrie. Remote Monitoring MIB Extensions for High Capacity Alarms. RFC 3434, Internet Engineering Task Force, December 2002.
17. Matt Blaze, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, and Michael Wiener. Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security, January 1996.
18. Uri Blumenthal and Bert Wijnen. User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3). RFC 3414, Internet Engineering Task Force, December 2002.
19. Larry Blunk and John Vollbrecht. PPP Extensible Authentication Protocol (EAP). RFC 2284, Internet Engineering Task Force, March 1998.
20. Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proceedings of the seventh annual international conference on Mobile computing and networking*, July 2001.
21. Steve Bostock. SNMP over IPX. RFC 1420, Internet Engineering Task Force, March 1993.
22. Robert Braden. Requirements for Internet Hosts – Communication Layers. RFC 1122, Internet Engineering Task Force, October 1989.
23. Caralyn Brown, Fred Baker, and Charles Carvalho. Management Information Base for Frame Relay DTEs. RFC 1315, Internet Engineering Task Force, April 1992.
24. Berkeley Software Distribution. `http://www.bsd.org`.
25. Steve Burnett and Stephen Paine. *Kryptographie. RSA Security's Official Guide.* RSA Press, September 2001.
26. Pat Calhoun, John Loughney, Erik Guttman, Glen Zorn, and Jari Arkko. Diameter Base Protocol. RFC 3588, Internet Engineering Task Force, September 2003.
27. Ross Callon. Use of OSI IS-IS for Routing in TCP/IP and Dual Environments. RFC 1195, Internet Engineering Task Force, December 1990.
28. Jeffrey Case, Mark Fedor, Martin Schoffstall, and James Davin. A Simple Network Management Protocol. RFC 1157, Internet Engineering Task Force, May 1990.
29. Jeffrey Case, Keith McCloghrie, Marshall Rose, and Steve Waldbusser. Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2). RFC 1442, Internet Engineering Task Force, April 1993.
30. Jeffrey Case, Keith McCloghrie, Marshall Rose, and Steve Waldbusser. Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2). RFC 1904, Internet Engineering Task Force, January 1996.
31. Jeffrey Case, Keith McCloghrie, Marshall Rose, and Steve Waldbusser. Introduction to Community-based SNMPv2. RFC 1901, Internet Engineering Task Force, January 1996.
32. Jeffrey Case, Keith McCloghrie, Marshall Rose, and Steve Waldbusser. Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2). RFC 1907, Internet Engineering Task Force, January 1996.

33. Jeffrey Case, Keith McCloghrie, Marshall Rose, and Steve Waldbusser. Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2). RFC 1905, Internet Engineering Task Force, January 1996.
34. Jeffrey Case, Keith McCloghrie, Marshall Rose, and Steve Waldbusser. Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2). RFC 1902, Internet Engineering Task Force, January 1996.
35. Jeffrey Case, Keith McCloghrie, Marshall Rose, and Steve Waldbusser. Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2). RFC 1903, Internet Engineering Task Force, January 1996.
36. Jeffrey Case, Keith McCloghrie, Marshall Rose, and Steve Waldbusser. Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2). RFC 1906, Internet Engineering Task Force, January 1996.
37. Evan Caves, Pat Calhoun, and Ross Wheeler. Layer Two Tunneling Protocol "L2TP" Management Information Base. RFC 3371, Internet Engineering Task Force, August 2002.
38. Jeffrey Caves, David Harrington, Randy Presuhn, and Bert Wijnen. Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). RFC 3412, Internet Engineering Task Force, December 2002.
39. Vint Cerf. The Catenet Model for Internetworking. IEN 48, Defense Advanced Research Projects Agency, July 1978.
40. Computer Emergency Response Team / Coordination Center. `http://www.cert.org`.
41. Kwok Chan, David Durham, Silvano Gai, Shai Herzog, Keith McCloghrie, Francis Reichmeyer, John Seligson, Raj Yavatkar, and Andrew Smith. COPS Usage for Policy Provisioning (COPS-PR). RFC 3084, Internet Engineering Task Force, March 2001.
42. Inc. Cisco Systems. *Cisco PIX Firewall Command Reference—Version 6.3.* Cisco Systems, Inc., 2004.
43. JTC 1/SC 2 Committee. Information technology – ISO 7-bit coded character set for information interchange. Standard X3.4-1986, International Organization for Standardization, 1986.
44. The Paging Technical Committee. Telocator Alphanumeric Protocol. `http://www.pagingcarriers.org/PTCdocs/TAP%20v1.8.pdf`, February 1997. Version 1.8.
45. Compaq, Intel, Microsoft, Phoenix, and Toshiba. *Advanced Configuration and Power Interface Specification*, 2.0 edition, July 2000.
46. IBM Corporation. SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2, 1985.
47. Bill Croft and John Gilmore. Bootstrap Protocol (BOOTP). RFC 951, Internet Engineering Task Force, September 1985.
48. William M. Daley and Raymond G. Kammer. Data Encrytion Standard. Federal Information Processing Standards Publication 46-, National Institute of Standards and Technology, October 1999.
49. Mike Daniele. IP Version 6 Management Information Base for the Transmission Control Protocol. RFC 2452, Internet Engineering Task Force, December 1998.
50. Defense Advanced Research Projects Agency. `http://www.darpa.mil`.
51. John D. Day and Hubert Zimmermann. The OSI Reference Model. *Proceedings of the IEEE*, 71(12):1334–1340, December 1983.

52. Kathryn de Graaf, Dan Romascanu, Donna McMaster, and Keith McCloghrie. Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIv2. RFC 2108, Internet Engineering Task Force, February 1997.
53. Stephen Edward Deering. Host Extensions for IP Multicasting. RFC 1112, Internet Engineering Task Force, August 1989.
54. Stephen Edward Deering. Internet Group Management Protocol. RFC 1112, Internet Engineering Task Force, August 1989.
55. Stephen Edward Deering. ICMP Router Discovering Messages. RFC 1256, Internet Engineering Task Force, September 1991.
56. Stephen Edward Deering and Robert M. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, Internet Engineering Task Force, December 1998.
57. Luca Delgrossi and Louis Berger. Internet Stream Protocol Version 2 (ST2) Protocol Specification—Version ST2+. RFC 1819, Internet Engineering Task Force, August 1995.
58. Dell, Inc. `http://www.dell.com`.
59. Peter Deutsch and Jean-Loup Gailly. ZLIB Compressed Data Format Specification Version 3.3. RFC 1950, Internet Engineering Task Force, May 1996.
60. Tim Dierks and Christopher Allan. The TLS Protocol Version 1.0. RFC 2246, Transport Layer Security Working Group, January 1999.
61. Whitfield Diffie and Martin Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, pages 644–654, November 1976.
62. Distributed.Net. `http://www.distributed.net/`.
63. Distributed Management Task Force. `http://www.dmtf.org`.
64. Ralph Droms. Dynamic Host Configuration Protocol. RFC 2131, Internet Engineering Task Force, March 1997.
65. Stephan Emile and Jordi Palet. Remote Monitoring (RMON) Protocol Identifiers for IPv6 and Multi Protocol Label Switching (MPLS). RFC 3919, Internet Engineering Task Force, October 2004.
66. Paul Ferguson and Daniel Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2267, Internet Engineering Task Force, January 1998.
67. Richard Feynman. *Sie belieben wohl zu scherzen, Mr. Feynman.* Piper, 7. edition, July 1994.
68. Roy Thomas Fielding, James Gettys, Jeffrey Clifford Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. Hypertext Transfer Protocol—HTTP/1.1. RFC 2616, Internet Engineering Task Force, June 1999.
69. Craig Finseth. An Access Control Protocol, Sometimes Called TACACS. RFC 1492, Internet Engineering Task Force, July 1993.
70. Distributed Management Task Force. *Alert Standard Format Specification*, April 2003. Version 2.0.
71. The Apache Software Foundation. Apache. `http://www.apache.org/`.
72. Alan O. Freier, Paul C. Kocher, and Philip L. Karlton. The SSL Protocol Version 3.0. Internet-Draft—Work in Progress draft-freier-ssl-version3-02.txt, Internet Engineering Task Force, November 1996.
73. Paul Funk and Simon Blake-Wilson. EAP Tunneled TLS Authentication Protocol Version 1 (EAP-TTLSv1). Internet-Draft—Work in Progress draft-funk-eap-ttls-v1-00.txt, Internet Engineering Task Force, February 2005.

74. Martin Gergeleit, Michael Mock, and Edgar Nett. *Das drahtlose Ethernet . Der IEEE 802.11 Standard: Grundlagen und Anwendung.* Addison-Wesley, February 2001.
75. The Open Group. The Single UNIX Specification, Version 2. POSIX 1003.2, The Open Group, 1997.
76. W3C Group. Portable Network Graphics (PNG) Specification. W3c recommendation, World Wide Web Consortium, November 2003.
77. W3C Group. Extensible Markup Language (XML) 1.1. W3c recommendation, World Wide Web Consortium, February 2004.
78. X9F1 Cryptographic Tools Working Group. Triple Data Encryption Algorithms Modes of Operation. Standard X9.52:1998, X9F Data & Information Security, 1998.
79. Neil Haller. The S/KEY One-Time Password System. RFC 1760, Internet Engineering Task Force, February 1995.
80. Charles Leonard Hamblin. Translation to and from Polish Notation. *The Computer Journal*, 5(1):210–213, April 1962.
81. David Harrington, Randy Presuhn, and Bert Wijnen. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. RFC 3411, Internet Engineering Task Force, December 2002.
82. Charles Hedrick. Routing Information Protocol. RFC 1058, Internet Engineering Task Force, June 1988.
83. Hewlett-Packard Development Company, L. P. `http://www.hp.com`.
84. Hewlett-Packard. Digital Equipment Corporation Network, 1975.
85. Robert M. Hinden and Alan Sheltzer. The DARPA Internet Gateway. RFC 823, Bolt Beranek and Newman Inc., September 1982.
86. John Holliman, Bruce Francis, and Marsha Walton. Starr report causes Internet slowdown, but no meltdown. `http://www.cnn.com/TECH/computing/9809/11/internet.congestion/`, September 1998.
87. Kevin Houle, George Weaver, and Ian Finlay. Exploitation of BIND Vulnerabilities. CERT Incident Note IN-2001-03, CERT Coordination Center, March 2001.
88. Hewlett-Packard HP-UX. `http://www.hp.com/go/hpux`.
89. Institute of Electrical and Electronics Engineering, Inc. `http://www.ieee.org`.
90. Microsoft Internet Information Server (IIS). `http://www.microsoft.com/WindowsServer2003/iis/default.mspx`.
91. Intel Corporation. `http://www.intel.com`.
92. Intel. *8742 Universal Periphal Interface 8-Bit Slave Microcontroller*, November 1991.
93. Intel, Hewlett-Packard, NEC, and Dell. *IPMI—Intelligent Platform Management Bos Communications Protocol Specification V1.0*, 1 edition, November 1999.
94. Intel, Hewlett-Packard, NEC, and Dell. *IPMI—Intelligent Chassis Management Bus Bridge Specification V1.0*, 1.3 edition, April 2003.
95. Intel, Hewlett-Packard, NEC, and Dell. *IPMI—Intelligent Platform Management Interface Specification Second Generation V2.0*, 1 edition, February 2004.
96. International Organization for Standardization. Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework. International Standard 7498-4, International Organization for Standardization, 11 1989.

97. Industry Promoters, Adopters and Contributors (updated 8/2/2005). `http://www.intel.com/designer/servers/ipmi/adopterlist.htm`.
98. The Internet Storm Center. `http://isc.sans.org`.
99. ITU-T Study Group 17. Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1). International Standard 8824, International Organization for Standardization, December 1987.
100. ITU-T Study Group 17. Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Notation One (ASN.1). International Standard 8825, International Organization for Standardization, December 1987.
101. Java Technology. `http://java.sun.com`.
102. Java Technology. `http://java.sun.com/j2ee/`.
103. Brian Jewell and David Chuang. Definitions of Managed Objects for the Virtual Router Redundancy Protocol. RFC 2787, Internet Engineering Task Force, March 2000.
104. Stephen Kent and Randall Atkinson. Security Architecture for the Internet Protocol. RFC 2401, Internet Engineering Task Force, November 1998.
105. Brian W. Kernighan and Dennis MacAlistair Ritchie. *The C Programming Language*. Englewood Cliffs, 2. edition, 1978.
106. Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, Internet Engineering Task Force, February 1997.
107. John Linn. Generic Security Service Application Program Interface, Version 2. RFC 2078, Internet Engineering Task Force, January 1997.
108. Brian Lloyd and William Allen Simpson. PPP Authentication Protocols. RFC 1334, Internet Engineering Task Force, October 1992.
109. Chris Lonvick. The BSD syslog Protocol. RFC 3164, Internet Engineering Task Force, August 2001.
110. Mark K. Lottor. Simple File Transfer Protocol. RFC 913, Internet Engineering Task Force, September 1984.
111. Gary Scott Malkin. Traceroute Using an IP Option. RFC 1393, Internet Engineering Task Force, January 1993.
112. Gary Scott Malkin. RIP Version 2. RFC 2453, Internet Engineering Task Force, November 1998.
113. Douglas R. Mauro and Kevin J. Schmidt. *Essential SNMP*. O'Reilly, July 2001.
114. Keith McCloghrie. SNMPv2 Management Information Base for the Internet Protocol using SMIv2. RFC 2011, Internet Engineering Task Force, November 1996.
115. Keith McCloghrie. SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2. RFC 2012, Internet Engineering Task Force, November 1996.
116. Keith McCloghrie. SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2. RFC 2013, Internet Engineering Task Force, November 1996.
117. Keith McCloghrie, Michael Fine, John Seligson, Kwok Chan, Scott Hahn, Ravi Sahita, Andrew Smith, and Francis Reichmeyer. Structure of Policy Provisio-

ning Information (SPPI). RFC 3159, Internet Engineering Task Force, August 2001.
118. Keith McCloghrie and Frank Kastenholz. Evolution of the Interfaces Group of MIB-II. RFC 1573, Internet Engineering Task Force, January 1994.
119. Keith McCloghrie, David Perkins, and Jürgen Schönwälder. Conformance Statements for SMIv2. RFC 2580, Internet Engineering Task Force, April 1999.
120. Keith McCloghrie, David Perkins, and Jürgen Schönwälder. Structure of Management Information Version 2 (SMIv2). RFC 2578, Internet Engineering Task Force, April 1999.
121. Keith McCloghrie, David Perkins, and Jürgen Schönwälder. Textual Conventions for SMIv2. RFC 2579, Internet Engineering Task Force, April 1999.
122. Keith McCloghrie and Marshall Rose. Management Information Base for network management of TCP/IP-based internets. RFC 1066, Internet Engineering Task Force, August 1988.
123. Keith McCloghrie and Marshall Rose. Structure and identification of management information for TCP/IP-based internets. RFC 1065, Internet Engineering Task Force, August 1988.
124. Keith McCloghrie and Marshall Rose. Management Information Base for network management of TCP/IP-based internets. RFC 1156, Internet Engineering Task Force, May 1990.
125. Keith McCloghrie and Marshall Rose. Management Information Base for Network Management of TCP/IP-based internets:MIB-II. RFC 1213, Internet Engineering Task Force, March 1991.
126. Donna McMaster and Keith McCloghrie. Definitions of Managed Objects for IEEE 802.3 Repeater Devices. RFC 1368, Internet Engineering Task Force, October 1992.
127. Alfred John Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, October 1996.
128. Microsoft Corporation. `http://www.microsoft.com`.
129. Trudy Miller. Internet Reliable Transaction Protocol Functional and Interface Specification. RFC 938, Internet Engineering Task Force, February 1985.
130. David L. Mills. Exterior Gateway Protocol Formal Specification. RFC 904, Internet Engineering Task Force, April 1984.
131. Kevin David Mitnick. *The Art of Deception: Controlling the Human Element of Security*. Wiley, October 2002.
132. Jeffrey Clifford Mogul and Jon Postel. Internet Standard Subnetting Procedure. RFC 950, Internet Engineering Task Force, August 1985.
133. John T. Moy. OSPF Specification. RFC 1131, Internet Engineering Task Force, October 1989.
134. John T. Moy. OSPF Version 2. RFC 2328, Internet Engineering Task Force, April 1998.
135. John Myers. Simple Authentication and Security Layer (SASL). RFC 2222, Internet Engineering Task Force, October 1997.
136. National Instituts of Standards and Technology. Secure Hash Standard. FIPS PUB 180-2, National Instituts of Standards and Technology, August 2002.
137. NEC Corporation. `http://www.nec.com`.
138. Net-SNMP. `http://www.net-snmp.org`.

139. Clifford Neuman, Tom Yu, Sam Hartman, and Kenneth Raeburn. The Kerberos Network Authentication Service (V5). RFC 4120, Internet Engineering Task Force, July 2005.
140. Darren New and Marshall Rose. Reliable Delivery for syslog. RFC 3195, Internet Engineering Task Force, November 2001.
141. Kathleen Nichols, Steven Blake, Fred Baker, and David L. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Header. RFC 2474, Internet Engineering Task Force, December 1998.
142. Network Mapper (nmap). `http://www.insecure.org/nmap/`.
143. Novell, Inc. `http://www.novell.com`.
144. Inc. Novell. Internet Packet Exchange. `http://www.novell.com`.
145. Tobias Oetiker. RRDtool (Round Robin Database Tool). `http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/`.
146. Open Secure Shell (OpenSSH). `http://www.openssh.org/`.
147. George Orwell. *Nineteen Eighty-Four*. Secker and Warburg, June 1949.
148. IBM OS/390. `www.s390.ibm.com`.
149. OSA Technologies. `http://www.osatechnologies.com`.
150. Craig Partridge and Robert M. Hinden. Version 2 of the Reliable Data Protocol (RDP). RFC 1151, Internet Engineering Task Force, April 1990.
151. Conventional Peripheral Component Interconnect. `http://www.pcisig.com/members/downloads/specifications/conventional/PCI_LB3.0-2-6-04.pd`.
152. Charles Perkins. IP Encapsulation Within IP. RFC 2003, Internet Engineering Task Force, October 1996.
153. David Perkins and Evan McGinnis. *Understanding SNMP MIBs*. Prentice Hall, 1997.
154. Perl. `http://www.perl.org`.
155. Pretty Good Privacy (PGP). `http://www.pgp.com/`.
156. Philips Semiconductors. `http://www.semiconductors.philips.com`.
157. PHP: Hypertext Preprocessor (PHP). `http://www.php.net`.
158. David C. Plummer. An Ethernet Address Resolution Protocol – or – Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826, Internet Engineering Task Force, November 1982.
159. Jon Postel. User Datagram Protocol. RFC 768, Information Science Institute of the University of California, August 1980.
160. Jon Postel. Internet Control Message Protocol. RFC 792, Internet Engineering Task Force, September 1981.
161. Jon Postel. Internet Protocol. RFC 791, Information Science Institute of the University of California, September 1981.
162. Jon Postel. Transmission Control Protocol. RFC 793, Information Science Institute of the University of California, September 1981.
163. Jon Postel and Joyce K. Reynolds. Telnet Protocol Specification. RFC 854, Internet Engineering Task Force, May 1983.
164. Jon Postel and Joyce K. Reynolds. File Transfer Protocol (FTP). RFC 959, Internet Engineering Task Force, October 1985.
165. Jon Postel and Joyce K. Reynolds. Assigned Numbers. RFC 1700, Internet Engineering Task Force, October 1994.

166. Randy Presuhn. Management Information base (MIB) for the Simple Network Management Protocol (SNMP). RFC 3418, Internet Engineering Task Force, December 2002.
167. Randy Presuhn. Transport Mappings for the Simple Network Management Protocol (SNMP). RFC 3416, Internet Engineering Task Force, December 2002.
168. Randy Presuhn. Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP). RFC 3417, Internet Engineering Task Force, December 2002.
169. The Honeynet Project. *Know Your Enemy : Learning about Security Threats.* Addison-Wesley, 2nd edition, May 2004.
170. Rajiv Raghunarayan. Management Information Base for the Transmission Control Protocol (TCP). RFC 4022, Internet Engineering Task Force, March 2005.
171. Kadangode K. Ramakrishnan, Sally Floyd, and David L. Black. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, Internet Engineering Task Force, September 2001.
172. Eric Rescorla. HTTP over TLS. RFC 2818, Internet Engineering Task Force, May 2000.
173. Carl Rigney, Steve Willens, Allan Rubens, and William Simpson. Remote Authentication Dial In User Service (RADIUS). RFC 2865, Internet Engineering Task Force, June 2000.
174. Ronald L. Rivest. The MD4 Message-Digest Algorithm. RFC 1320, Internet Engineering Task Force, April 1992.
175. Ronald L. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, Internet Engineering Task Force, April 1992.
176. Ronald L. Rivest. The RC4 Encryption Algorithm, March 1992.
177. Ronald L. Rivest. The RC5 Encryption Algorithm. In Bart Preneel, editor, *Proceedings of the 1994 Leuven Workshop on Fast Software Encryption.* Springer, January 1996.
178. Ronald L. Rivest. A Description of the RC2(r) Encryption Algorithm. RFC 2268, Internet Engineering Task Force, March 1998.
179. Ronald L. Rivest, Adi Shamir, and Len Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. In *Communications of the ACM*, pages 120–126, February 1978.
180. Dan Romascanu. Remote Monitoring MIB Extensions for Interface Parameters Monitoring. RFC 3144, Internet Engineering Task Force, August 2001.
181. Marshall Rose. Management Information Base for network management of TCP/IP-based internets: MIB-II. RFC 1158, Internet Engineering Task Force, May 1990.
182. Marshall Rose. A Convention for Defining Traps for Use with the SNMP. RFC 1215, Internet Engineering Task Force, March 1991.
183. Marshall Rose. The Blocks Extensible Exchange Protocol Core. RFC 3080, Internet Engineering Task Force, March 2001.
184. Marshall Rose and Keith McCloghrie. Structure and identification of management information for TCP/IP-based internets. RFC 1155, Internet Engineering Task Force, May 1990.
185. Marshall Rose and Keith McCloghrie. Concise MIB definitions. RFC 1212, Internet Engineering Task Force, March 1991.

186. Eric C. Rosen, Arun Viswanathan, and Ross Callon. Multiprotocol Label Switching Architecture. RFC 3031, Internet Engineering Task Force, January 2001.
187. Florian Rötzer. FBI bestätigt Entwicklung des Schnüffelprogramms Magic Lantern. `http://www.heise.de/tp/deutsch/inhalt/te/11333/1.html`, December 2001. Heise Newsticker.
188. Gerhard Schmid. Bericht über die Existenz eines globalen Abhörsystems für private und wirtschaftliche Kommunikation (Abhörsystem ECHELON) (2001/2098 (INI)). Sitzungsdokument A5-0264/2001, Europäisches Parlament – Nichtständiger Ausschuss über das Abhörsystem Echelon, July 2001.
189. Bruce Schneier. *Angewandte Kryptographie . Protokolle, Algorithmen und Sourcecode in C.* Addison-Wesley, 5. edition, May 1996.
190. Jürgen Schönwälder. Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping. RFC 3430, Internet Engineering Task Force, December 2002.
191. Henning Schulzrinne, Anup Rao, and Robert Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326, Internet Engineering Task Force, April 1998.
192. Philips Semiconductors. *The I^2C Bus Specification—Version 2.1*, January 2000. Document 9398 393 40011.
193. Colleen Shannon and David Moore. The Spread of the Witty Worm. Technical report, Cooperative Association for Internet Data Analysis, April 2004. `http://www.caida.org/analysis/security/witty/`.
194. William Simpson. PPP in HDLC-like Framing. RFC 1662, Internet Engineering Task Force, July 1994.
195. William Simpson. The Point-to-Point Protocol (PPP). RFC 1661, Internet Engineering Task Force, July 1994.
196. William Allen Simpson. PPP Challenge Handshake Authentication Protocol (CHAP). RFC 1994, Internet Engineering Task Force, August 1996.
197. Nigel Smart. *Cryptography.* McGraw-Hill Education - Europe, September 2002.
198. Secure Multipurpose Internet Mail Extensions (S/MIME). `http://www.ietf.org/html.charters/smime-charter.html`.
199. Andrew Smith, John Flick, Kathryn de Graaf, Dan Romascanu, Donna McMaster, Keith McCloghrie, and Sam Roberts. Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs). RFC 2668, Internet Engineering Task Force, August 1999.
200. SNMP Link. `http://www.snmplink.org`.
201. Snort. `http://www.snort.org/`.
202. Sherry Sontag, Christopher Drew, and Annette Lawrence Drew. *Blind Man's Bluff.* PublicAffairs, November 1998.
203. Jennifer Steiner, Clifford Neuman, and Jeffrey Schiller. Kerberos: An Authentication Service for Open Network Systems. In *USENIX Conference Proceedings*, pages 191–202. USENIX, February 1988.
204. Sven Stillich and Dirk Liedtke. Der Wurm von der Wümme. `http://www.stern.de/computer-technik/internet/?id=525454&eid=501069`, June 2004. stern.de.
205. Frank Strauß and Jürgen Schönwälder. Next Generation Structure of Management Information (SMIng) Mappings to the Simple Network Management Protocol (SNMP). RFC 3781, Internet Engineering Task Force, May 2004.

206. Frank Strauß and Jürgen Schönwälder. SMIng - Next Generation Structure of Management Information. RFC 3780, Internet Engineering Task Force, May 2004.
207. Sun Solaris. `http://www.sun.com/solaris`.
208. Jonathan Swift. *Gulliver's Travels*. David Price, george bell and sons edition, 1892.
209. Cisco Systems. Hot Standby Router Protocol Features and Functionality. `http://www.cisco.com/warp/public/619/hsrpguidetoc.pdf`, August 2005. Document ID: 9234.
210. Technical Committee Group NCITS H2, Database. Database Language SQL. ANSI/ISO/IEC Standard 9075:1999, American National Standards Institute, 1999.
211. The 802.1 Working Group. IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control. IEEE Standard IEEE 802.1X-2001, Institute of Electrical and Electronics Engineers, 2001.
212. The 802.1 Working Group. IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks. IEEE Standard IEEE 802.1Q-2003, Institute of Electrical and Electronics Engineers, 2003.
213. The 802.1 Working Group. IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Bridges. IEEE Standard IEEE 802.1D-2004, Institute of Electrical and Electronics Engineers, 2004.
214. The 802.11 Working Group. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher speed Physical Layer Extension in the 2.4 GHz band. IEEE Standard IEEE 802.11b-1999, Institute of Electrical and Electronics Engineers, 1999.
215. The Honeynet Project. `http://project.honeynet.org/`.
216. Steve Waldbusser. Token Ring Extensions to the Remote Network Monitoring MIB. RFC 1513, Internet Engineering Task Force, September 1993.
217. Steve Waldbusser. Remote Network Monitoring Management Information Base Version 2 Using SMIv2. RFC 2021, Internet Engineering Task Force, January 1997.
218. Steve Waldbusser. Remote Network Monitoring Management Information Base. RFC 2819, Internet Engineering Task Force, May 2000.
219. Steve Waldbusser. Remote Network Monitoring Management Information Base for High Capacity Networks. RFC 3273, Internet Engineering Task Force, July 2002.
220. Steve Waldbusser. Application Performance Measurement MIB. RFC 3729, Internet Engineering Task Force, March 2004.
221. Steve Waldbusser, Karl Kalbfleisch, Robert Cole, and Dan Romascanu. Introduction to the Remote Monitoring (RMON) Family of MIB Modules. RFC 3577, Internet Engineering Task Force, August 2003.
222. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In *CRYPTO 2005*, August 2005.
223. Wireless Application Protocol (WAP). `http://www.openmobilealliance.org`.
224. Richard Waterman, Bill Lahaye, Dan Romascanu, and Steve Waldbusser. Remote Network Monitoring MIB Extensions for Switched Networks Version 1.0. RFC 2613, Internet Engineering Task Force, June 1999.

225. Nicholas Weaver. The Spread of the Sapphire/Slammer SQL Worm. http://www.securityfocus.com/archive/1/309776, February 2003. SecurityFocus BugTraq.
226. Bert Wijnen and Andy Bierman. IANA Guidelines for the Registry of Remote Monitoring (RMON) MIB Modules. RFC 3737, Internet Engineering Task Force, April 2004.
227. Bert Wijnen, Randy Presuhn, and Keith McCloghrie. View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). RFC 3415, Internet Engineering Task Force, December 2002.
228. Andreas Wilkens. Vivendi Universal verzögert Ballerspiel Half-Life 2 weiter. http://www.heise.de/newsticker/meldung/40899, October 2003. Heise Newsticker.
229. Wireless-Fidelity Protected Access (WPA). http://www.wi-fi.org.
230. Francois Yergeau. UTF-8, a Transformation Format of ISO 10646. RFC 3629, Internet Engineering Task Force, November 2003.
231. Tatu Ylonen and Chris Lonvick. SSH Protocol Architecture. Internet-Draft—Work in Progress draft-ietf-secsh-userauth-27.txt, Internet Engineering Task Force, March 2005.
232. Glen Zorn. Microsoft PPP CHAP Extensions, Version 2. RFC 2759, Internet Engineering Task Force, January 2000.
233. Glen Zorn and Steve Cobb. Microsoft PPP CHAP Extensions. RFC 2433, Internet Engineering Task Force, October 1998.
234. IBM zero down-time Operating System (z/OS). http://www-03.ibm.com/servers/eserver/zseries/library/specsheets/pdf/gm%130122.pdf.

Sachverzeichnis