Implementing DomainKeys/DKIM on Mac OS X 10.5.x/10.4.x

1. - Introduction

The purpose of this document is to provide instructions on how to implement DomainKeys/DKIM on OS X 10.5.x and OS X 10.4.x Server.

You will not find many explanations as to why something is done one way or the other. Also, I will not discuss whether DomainKeys/DKIM are useful or not. This is a decision you must make for yourself. There are enough discussions about this available on the internet.

DomainKeys/DKIM functionality has two sides to it. First, it is used to verify if a sender domain is using DomainKeys/DKIM signatures and if the incoming mail was correctly signed. Second, it allows you to sign outgoing messages with a digital signature for recipients to verify your mail server.

DomainKeys/DKIM functionality depends on several Perl modules and scripts to be installed.

Verification of signatures is done through amavisd-new/ SpamAssassin. This allows to integrate as closely as possible with the existing components on OS X 10.4.x server.

Signing is handled by an external content-filter which will be integrated with Postfix.

NOTE: Although not mandatory, for Mac OS X 10.4.x it is strongly recommended you first update amavisd-new and SpamAssassin. This is not necessary for Mac OS X 10.5.x. Some required Perl modules, may conflict with older versions of amavisd-new. Instructions for updating amavisd-new and SpamAssassin on OS X 10.4.x server can be downloaded from http://osx.topicdesk.com/

This document will require you to use the command line. If you do not feel comfortable with using the command line, you should look for a ready made installer package or for somebody to assist you.

This document is written for Mac OS X 10.5.x/10.4.x. It does not apply to 10.3.x. as it did not come with the same content filtering components pre-installed.

DISCLAIMER: Whatever you do based on this document, you do it at your own risk! Just in case you haven't understood: Whatever you do based on this document, you do it at your own risk!

This tutorial has been tested on a standard Mac OS X 10.5.x/ 10.4.x Server installation. If you have already tinkered with your system, be aware that things might differ. It is impossible for me to foresee all changes that one might have applied to a server.

This tutorial contains step-by-step instructions for the terminal. Although you could just type them in line by line, it is recommended you have a basic understanding of the terminal.

2. - Requirements

Before you get started, you need to make sure some basic requirements are met:

- You have made a backup of your system.
- You have the latest version of Apple's Developer Tools (Xcode 2.4 or higher for 10.4.x and XCode 3.0 or higher for 10.5.x) installed.
  Dev Tools are available on your Server DVD and as a free download from Apple's Developer Connection.
- You do have a backup
- You are running 10.4.x/10.5.x or greater
- Although not mandatory, for Mac OS X 10.4.x it is STRONGLY (very STRONGLY) recommended you first update amavisd-new and SpamAssassin.
  Some of the Perl modules necessary, may conflict with older versions of amavisd-new and SpamAssassin. Instructions for updating amavisd-new and SpamAssassin on OS X 10.4.x server can be downloaded from http://osx.topicdesk.com/
- Familiarity with a command line editor or alternatively a GUI plain text editor (do NOT use Word or similar)

- Not a requirement, but it is recommended you subscribe to our newsletter(s) to be informed when updated versions of this tutorial become available:
  http://osx.topicdesk.com/newsletter/

3. - Installing missing and required components

As mentioned, you will need a few perl modules to be able to use DomainKeys/DKIM. This chapter will guide you through getting and installing them.

So let's get going:
Make sure you are logged in as root (or alternatively use sudo).

Install the required modules by issuing the following commands (*in oblique type*). Issue them one after the other making sure you do not miss any dots or slashes. Also note that the download URLs given may change in the future. In that case just replace the URLs in this document with the current ones.

NOTE: Lines wrapping without line spacing are a single command.

The easiest way to install them is by using CPAN. Unfortunately, one of the modules tends to cause issues when installed through CPAN, so we will handle this separately.

First we will install Crypt::OpenSSL::RSA by issuing:

sudo mkdir -p /SourceCache

cd /SourceCache

sudo curl -O http://mirror.switch.ch/ftp/mirror/CPAN/authors/id/I/IR/IROBERTS/Crypt-OpenSSL-RSA-0.25.tar.gz

sudo tar xzf Crypt-OpenSSL-RSA-0.25.tar.gz

cd Crypt-OpenSSL-RSA-0.25

sudo perl Makefile.pl

sudo make install

Next we continue on to the modules that can be installed through CPAN.To do so issue:

sudo perl -MCPAN -e shell

If you have never used CPAN before you will be prompted to supply a few parameters. Just accept the default values. Once done, you should see the CPAN prompt (*cpan >*):

When at the CPAN prompt issue:

o conf prerequisites_policy ask

This will prompt you when a modules relies on other pre-requisites that have to be installed first. You should allow it to go ahead if asked.

Now you are ready to install the missing module(s). Be aware that some modules already exist on your server, but are outdated so it is best to install them all.

Just issue:

install Crypt::OpenSSL::Random

install Crypt::OpenSSL::Bignum

install Digest::SHA

install Digest::SHA1

install Error

install Mail::Address

install MIME::Base64

install Net::DNS (do NOT install on Mac OS X 10.5)

install Net::Server

install Mail::DKIM

This will install the modules and bring you back to the CPAN prompt.

Now issue

exit

to exit CPAN.

NOTE: It is possible that some of the modules will not install. In that case use "*force install*" instead of "*install*" at the CPAN prompt.

NOTE: If you had previously tried to use CPAN without having the Developer Tools installed, you will need to make sure that Developer Tools are now correctly installed and you will also need to re-configure CPAN. To do so get to the CPAN prompt and issue:

o conf init

You will be prompted to supply a few parameters. Just accept the default values.

4. - Getting and installing DomainKeys/DKIM components

This chapter will guide you through getting and installing the DomainKeys/DKIM components.

The components we will use are Mail::DKIM (to be used with SpamAssassin for DomainKeys/DKIM verification) and dkimproxy (to be use with Postfix to sign outgoing messages). Both modules are written and maintained by Jason Long and are available from http://jason.long.name/dkimproxy/

There are other tools and combinations available out there, but this one makes most sense for OS X 10.4.x server.

So let's get going:
Make sure you are logged in as root (or alternatively use sudo).

Install the latest version of DomainKeys/DKIM by issuing the following commands (*in oblique type*). Issue them one after the other making sure you do not miss any dots or slashes. Also note

that the download URLs given may change in the future. In that case just replace the URLs in this document with the current ones.

NOTE: Lines wrapping without line spacing are a single command.

mkdir -p /SourceCache

cd /SourceCache

sudo curl -O http://kent.dl.sourceforge.net/sourceforge/dkimproxy/dkimproxy-1.1.tar.gz

sudo tar xzf dkimproxy-1.1.tar.gz

cd dkimproxy-1.1

sudo ./configure --prefix=/usr/local/dkfilter

sudo make install

Now everything we need to use DomainKeys/DKIM has been installed.

The next step is to configure everything for verification of incoming messages and signing of outbound messages.

NOTE: You can use incoming verification only, outbound signing only or both. The choice is yours, they do not have to be used together.

5. - Using Mail::DKIM to verify incoming messages

As mentioned, we will use Mail::DKIM together with SpamAssassin to verify incoming messages.

Depending on the version of SpamAssassin you have, you will need to either uncomment or add a few instructions.

Check the contents of */private/etc/mail/spamassassin*

If you find a file called *v312.pre* edit it and uncomment:

loadplugin Mail::SpamAssassin::Plugin::DKIM

If you don't have *v312.pre*, just edit init.pre and add the following lines:

 # DKIM - perform DKIM verification
#
loadplugin Mail::SpamAssassin::Plugin::DKIM

When done save.

Having made changes to the SpamAssassin configuration that do require network access, we also need to make a change in: */etc/amavisd.conf*

Edit */etc/amavisd.conf* and change (Mac OS X 10.4.x only. 10.5.x is already set correctly):

$sa_local_tests_only = 1;
to
$sa_local_tests_only = 0;

When done save.

Having made changes to the SpamAssassin configuration, we need to restart amavisd-new:

To do so issue:

sudo /bin/launchctl unload /System/Library/LaunchDaemons/org.amavis.amavisd.plist

sudo /bin/launchctl load /System/Library/LaunchDaemons/org.amavis.amavisd.plist

You may see the following error:
"Workaround Bonjour: Unknown error: 0"
It's due to a bug introduced in 10.4.7 and safe to ignore.

Now everything is ready for incoming DKIM verification. Send yourself an e-mail from a domain that uses DomainKeys/DKIM (e.g. yahoo.com, gmail) and check the headers. You should see something along the lines of:
DKIM_SIGNED=0.001
DKIM_VERIFIED=-0.001
in the X-Spam-Status Tests.

The scores are low on purpose by default. It is up to you to change them if you would like action to be taken based on this information. Simply edit */private/etc/mail/spamassassin/local.cf* (or wherever you keep your score adjustments) and add:

```
score DKIM_POLICY_SIGNALL 0.001
score DKIM_POLICY_SIGNSOME 0.001
score DKIM_POLICY_TESTING 0.001
score DKIM_SIGNED 0.001
score DKIM_VERIFIED -0.001
```

(replace *0.001* with the score you want)

Remember to restart amavisd-new after score changes.

6. – Using dkimproxy to sign outgoing messages

As mentioned, we will use dkimproxy together with Postfix to sign outgoing messages.

6.1. The first step is to generate a set of keys to be used for our signature.

To do so issue:

cd /usr/local/dkfilter

sudo openssl genrsa -out private.key 1024

sudo openssl rsa -in private.key -pubout -out public.key

For 10.4.x:
sudo chown root:clamav private.key

For 10.5.x:
sudo chown root:_amavisd private.key

sudo chmod 640 private.key

6.2. The next step is to create a configuration file for dkimproxy.

cd /etc

sudo touch dkimproxy_out.conf

Above command created a new empty configuration file. Edit

/etc/dkimproxy_out.conf

and add the following content (NOTE: 10.4.x and 10.5.x use different system users for running amavisd, clamav and spamassassin. 10.4.x uses user "clamav", while 10.5.x uses user "_amavisd". Below snippet is ready for 10.4.x. Please substitute "clamav" with "_amavisd" for 10.5.x (2 occurrences)):

```
# specify what address/port DKIMproxy should listen on
listen    127.0.0.1:10027

# specify what address/port DKIMproxy forwards mail to
relay     127.0.0.1:10028

# specify what domains DKIMproxy can sign for (comma-
separated, no spaces)
domain    yourdomain.tld

# specify what signatures to add
signature dkim(c=relaxed)
signature domainkeys(c=simple)

# specify location of the private key
keyfile   /usr/local/dkfilter/private.key

# specify the selector (i.e. the name of the key record
put in DNS)
selector  default

user      clamav
group     clamav

pidfile   /var/run/dkimproxy_out.pid
```

NOTE: Make sure you replace *yourdomain.tld* with your domain name or a comma separated list of domain names.

NOTE: If you prefer only DKIM or only DomainKeys remove the signature type you don't want. The current version is capable of adding both types which is recommended.

6.3. The next step is to create a startup item for dkimproxy. Unfortunately, launchd doesn't play nicely with some daemonized scripts, so we'll use a Startup item.

NOTE: If you installed dkimproxy by following this tutorial for versions pre 1.0 of dkimproxy, you will need to change/re-do

your startup item. The configuration parameters are now read from the configuration file created in step 6.2

```
cd /System/Library/StartupItems

sudo mkdir -p Dkimproxy

cd /System/Library/StartupItems/Dkimproxy

sudo touch Dkimproxy
```

Above command created a new empty startup script. Edit

```
/System/Library/StartupItems/Dkimproxy/Dkimproxy
```

and add the following content:

```
#!/bin/sh

##
# DKIM Signing Daemon
##

. /etc/rc.common

StartService ()
{
/usr/local/dkfilter/bin/dkimproxy.out --daemonize --
conf_file=/etc/dkimproxy_out.conf;
}
StopService ()
{
if [ -f /var/run/dkimproxy_out.pid ]; then
kill `cat /var/run/dkimproxy_out.pid` && rm -f /var/run/
dkimproxy_out.pid
RETVAL=$?
[ $RETVAL -eq 0 ] && echo done. || echo failed.
  else
echo not running.
  fi
}
```

```
RestartService ()
{

 StopService
 sleep 1
 StartService
}
RunService "$1"
```

NOTE:

/usr/local/dkfilter/bin/dkimproxy.out --daemonize --conf_file=/etc/dkimproxy_out.conf;

and

kill `cat /var/run/dkimproxy_out.pid` && rm -f /var/run/dkimproxy_out.pid

are long single lines. Make sure you avoid line breaks when copy/pasting.

When done continue by issuing:

sudo chmod 755 /System/Library/StartupItems/Dkimproxy/Dkimproxy

cd /System/Library/StartupItems/Dkimproxy

sudo touch StartupParameters.plist

Above command created a new empty property list. Edit

/System/Library/StartupItems/Dkimproxy/StartupParameters.plist

and add the following content:

```
{
  Description     = "DKIM Signing Proxy";
  Provides        = ("DKIM Signing");
}
```

When done continue by issuing:

sudo /System/Library/StartupItems/Dkimproxy/Dkimproxy start

If all went well, dkimproxy is now running and ready to use.

To verify, issue:

For 10.5.x:
sudo ps -U _amavisd

For 10.4.x:
sudo ps -U clamav

Among other things you should see several lines starting with:
/usr/bin/perl -I/usr/local/dkfilter

6.4. The next step is to prepare your DNS records.

This procedure can differ based on what DNS software/provider you use. Many providers use different control panels, so you may have to adjust as needed. If you manage your own DNS, you'll know what to do.

In essence you need to create the following 2 TXT records for each domain you handle and want to sign. One for the DomainKeys policy record and one for the DomainKeys selector record.

_domainkey.yourdomain.tld TXT "o=~"

default._domainkey.yourdomain.tld TXT "k=rsa\;
p=MIGfMA0GCSqGSIb3DDOQAQUAA4GNADCBiQKBgQCdtxXkwuk2d8ZUeq
5W0gy3l39M9trMfI
+1ieMshy4DaIF6pFrGq9kiaNFZqcjFBoKdziEarHvcoY9IyaAFH5L6FO
xZsvyjniJW3Z76GWMH6JvQsl8vfqa4xM19YqNchBn/lU60V/
A7R0IDFgyk53Y4sPj4sscqFtR0FkUN+43bMQIDAQAB"

The long string looking like gibberish (after *p=*) is your public key and should be replaced with the contents of */usr/local/dkfilter/public.key.* Note that it is a single long line.

Also you should replace *yourdomain.tld* with your actual domain name.

To verify your policy record go here:
http://domainkeys.sourceforge.net/cgi-bin/check_policy?domain=yourdomain.tld&Submit=Submit

To verify your DomainKeys Selector record go here:
http://domainkeys.sourceforge.net/cgi-bin/check_selector?selector=default._domainkey. yourdomain.tld&Submit=Submit

6.5. The last step is to prepare Postfix for using dkimproxy.

This requires editing of */etc/postfix/master.cf*

NOTE: For this we will make the assumption that you are using (or are willing to use) a separate submission port for your outgoing mail. There are several advantages to this. First you will be able to send mail through your server while on the road (many Hotels and ISPs will block port 25, thus not allowing you to send through your server). Second, you can force authentication for your users. And last but not least, you keep CPU load down by signing only outgoing messages from your legit users.

The only disadvantage to this is that you will have to tell all your users to use port 587 (instead of 25) for sending through your mail server.

Now on to editing */etc/postfix/master.cf*

Add the following at the end of /etc/postfix/master.cf (sorry for the small print, but it is important no lines get wrapped, you can copy paste into a text editor for better visibility):

```
#
# Submission port 587
#
submission     inet    n       -       n       -       -       smtpd
   -o smtpd_etrn_restrictions=reject
   -o smtpd_sasl_auth_enable=yes
   -o content_filter=dksign:[127.0.0.1]:10027
   -o receive_override_options=no_address_mappings
   -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
#
# DKIM signing proxy
#
dksign    unix  -       -       n       -       10      smtp
   -o smtp_send_xforward_command=yes
127.0.0.1:10028 inet  n  -      n       -       10      smtpd
   -o content_filter=
   -o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
   -o smtpd_helo_restrictions=
   -o smtpd_client_restrictions=
   -o smtpd_sender_restrictions=
   -o smtpd_recipient_restrictions=permit_mynetworks,reject
   -o mynetworks=127.0.0.0/8
   -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

NOTE: If, you already have configured port 587 for submission, you only need to insert the section "DKIM signing proxy" and make sure the options for the submission port and their order match the ones showed above.

When done issue:

sudo postfix reload

You should be all set now.

Try and send an e-mail using your new submission port (587). If all went well, you should see your signature in the headers.

Something along these lines:

```
DomainKey-Signature: a=rsa-sha1; c=simple; d=domain.com; q=dns; s=default;
     b=Z6UN3jfZ/ylQ22pRRNnjXzp8jd45UYWuWZPtsNCURw+miK7D12SjslEkrNrwRL
     Ovj7hcugAxCDm03Pk0UHivIQpkK98zt6a3Fyy77rLmIeFIne/a0PpZDXHLEL1iq4
     PcwDII3ne0+AiaYvpXdZ3c72o4LR4tr67P6C+LHT9v6U0=
```

NOTE: The settings chosen are based on my personal preference and experience. You may want to change them as you deem fit. Check the dkimproxy documentation for information. See here: http://jason.long.name/dkimproxy/

7. - Caveats

There most frequent issues to watch out for are:

a) Incompatible perl modules
b) Typos made when applying this tutorial
c) Long lines seen as multiple lines. Watch for incorrect line breaks

Also, if you have modified any paths and or environment variables, make sure you check them against above instructions.

That's all folks.
Hope this helps.
Have fun,
Alex

Doc. Version 1.1.6, 8.5.2009
Athanasios Alexandrides
Lugano, Switzerland
tutorials -at- topicdesk.com
