Implementing DomainKeys/DKIM on OS X 10.8.x Mountain Lion With Server 2.x and OS X 10.9.x Mavericks With Server 3.x

> DISCLAIMER: The author(s) claim(s) no responsibility for any damage that may occur from the use of any information found here or found on links followed from this document. Please make sure you have a backup before applying modifications to your server.

1. Introduction

The purpose of this document is to provide instructions on how to implement DomainKeys/DKIM on OS X 10.8.x Mountain Lion with Server 2.x and OS X 10.9.x Mavericks with Server 3.x.

You will not find many explanations as to why something is done one way or the other. Neither will I discuss whether DomainKeys/DKIM are useful or not. This is a decision you must make for yourself. There are plenty of 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.

Unlike previous versions of this tutorial which were based on dkimproxy and amavisd, this tutorial relies only on amavisd. dkimproxy is still a valid solution, but it hasn't been maintained much in the past couple of years. On the other hand, Apple is now including an updated version of amavisd-new with its Server OS, so why rely on extra components when all we need is already available.

Verification of signatures is done through amavisd-new/SpamAssassin. This allows to integrate as closely as possible with the existing components on OS X 10.8.x Mountain Lion with Server 2.x and OS X 10.9.x Mavericks Server 3.x.

Signing is handled by amavisd-new and Postfix.

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 OS X 10.8.x Mountain Lion with Server 2.x and OS X 10.9.x Mavericks Server 3.x. It does not apply to earlier versions. Separate versions of this tutorial are available for earlier Mac OS X Server versions.

If you have used Mac OS X Server releases prior to OS X 10.8.x Mountain Lion with Server 2.x or OS X 10.9.x Mavericks Server 3.x, you are most likely used to a series of standard paths. While some of them are still the same, many have changed in an attempt by Apple to better consolidate server related files and binaries. So always be very careful and double-check which file you are editing.

This tutorial has been tested on a standard OS X 10.8.x Mountain Lion with Server 2.x or OS X 10.9.x Mavericks Server 3.x 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.

> 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!

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 are running OS X 10.8.x Mountain Lion with Server 2.x. or OS X 10.9.x Mavericks Server 3.x
- You do have a backup
- Familiarity with a command line editor or alternatively a GUI plain text editor (do NOT use Word or similar)
- While not a requirement, it is recommended you subscribe to our newsletter or follow us on Twitter or App.net to be informed when updated versions of this and other tutorials become available:
  Newsletter: http://topicdesk.com/newsletter/
  Twitter: @topicdesk

3. Using Mail::DKIM to verify incoming messages

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

Nothing needs to be done in OS X 10.8.x Mountain Lion with Server 2.x or OS X 10.9.x Mavericks Server 3.x. Everything is already correctly configured.

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:

```
/Library/Server/Mail/Config/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.

Note: If you don't see any `X-Spam-Status` in your e-mail's headers, you need to edit:

```
/Library/Server/Mail/Config/amavisd/amavisd.conf
```

and make sure the following parameter is set (by default it is set to `2.0` which will not tag low scoring mails):

```
$sa_tag_level_deflt  = -999.0;
```

Remember to restart amavisd-new after changes to amavisd.conf.

4. Using amavisd-new to sign outgoing messages

As mentioned, we will use amavisd-new together with Postfix to sign outgoing messages.

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

To do so issue:

```
mkdir -p /var/db/dkim

chown _amavisd /var/db/dkim

sudo -u _amavisd -H amavisd genrsa /var/db/dkim/mydomain.tld.default.pem

sudo chown root:_amavisd /var/db/dkim/mydomain.tld.default.pem

sudo chmod 640 /var/db/dkim/mydomain.tld.default.pem
```

The following file:

```
/var/db/dkim/mydomain.tld.default.pem
```

now contains the private key used for DKIM signing.

4.2. The next step is to modify the configuration files for amavisd-new and

Postfix.

Edit:

```
/Library/Server/Mail/Config/amavisd/amavisd.conf
```

and make sure the following 2 settings are enabled as shown:

```
$enable_dkim_verification = 1;
$enable_dkim_signing = 1;
```

next , right below

```
$enable_dkim_signing
```

add:

```
dkim_key('mydomain.tld', 'default',
'/var/db/dkim/mydomain.tld.default.pem');

@dkim_signature_options_bysender_maps = (
{ '.' => { ttl => 21*24*3600, c => 'relaxed/simple' } } );
```

Note: you need to replace mydomain.tld with your actual domain.

So far so good. Now all we need to do is to add the keys to our DNS and we are all set.

To display the key(s), issue:

```
sudo -u _amavisd -H amavisd -c
/Library/Server/Mail/Config/amavisd/amavisd.conf showkeys
```

You should see something along the lines of:

```
; key#1, domain mydomain.tld, /var/db/dkim/mydomain.tld.default.pem
default._domainkey.mydomain.tld.    3600 TXT (
"v=DKIM1;
p=MIGfCSXUZqGSIb7DKIBQLOQA6GNAMNWiQKBgQCdtxXkwuk2d8ZUeq5W0gy3l39M9trMfI+1ieMshy
```

This is your public key inside a DNS TXT record.

4.3. 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.mydomain.tld  TXT  "o=~"

default._domainkey.mydomain.tld TXT "v=DKIM1;
p=MIGfCSXUZqGSIb7DKIBQLOQA6GNAMNWiQKBgQCdtxXkwuk2d8ZUeq5W0gy3l39M9trMfI+1ieMshy
```

The long string looking like gibberish (after p=) is your public key and should be replaced with the contents of:

```
/var/db/dkim/mydomain.tld.default.pem
```

Note that it is a single long line.

Also you should replace mydomain.tld with your actual domain name.

When done and after you are sure your new DNS records have propagated, issue:

```
sudo -u _amavisd -H amavisd -c
/Library/Server/Mail/Config/amavisd/amavisd.conf testkeys
```

If all is well, you'll see something like:

```
TESTING#1: default._domainkey.mydomain.tld => pass
```

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

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

If all checks out, you are set and from now on your outgoing e-mail will be signed with your DKIM key and amavisd-new/spamassassin will check incoming mails for valid DKIM keys.

Try and send an e-mail using your server. If all went well, you should see your signature in the full/raw headers of your message.

Something along these lines:

```
DomainKey-Signature: a=rsa-sha1; c=simple; d=mydomain.tld; q=dns; s=default;
MIGfCSXUZqGSIb7DKIBQLOQA6GNAMNWiQKBgQCdtxXkwuk2d8ZUeq5W0gy3l39M9trMfI+1ieMshy4I
```

5. Advanced configuration options

Above configuration will make sure that all outgoing mail for a configured domain will be signed as well as scanned for spam and viruses.

Sometimes it is preferable to offer multiple paths through the content filter. For example you might want to sign all of your outgoing mail, but at the same time would prefer if mail from your authenticated users is not scanned for spam or maybe assigned more lenient scores (this can be important when trying to send through your server from a dynamic IP).

I prefer having a setup where the content filter treats incoming and outgoing mail differently and thus will show you how to differentiate it through Postfix checks and separate amavisd-new policy banks. There are several advantages to this. First you will be able to send mail through your server without the risk of any false positives. Second, you keep CPU load down by signing only outgoing messages from your legit users.

This requires editing of

```
/Library/Server/Mail/Config/postfix/master.cf
/Library/Server/Mail/Config/postfix/main.cf
/Library/Server/Mail/Config/amavisd/amavisd.conf
```

Now on to editing:

```
/Library/Server/Mail/Config/postfix/main.cf
```

At the end of the file, add:

```
smtpd_sender_restrictions = check_sender_access
regexp:/Library/Server/Mail/Config/postfix/tag_for_signing
permit_mynetworks, permit_sasl_authenticated, reject_non_fqdn_sender,
check_sender_access
regexp:/Library/Server/Mail/Config/postfix/tag_for_scanning permit
```

Save `main.cf` and create a new file called `tag_for_signing`:

```
sudo touch /Library/Server/Mail/Config/postfix/tag_for_signing
```

Edit it and add:

```
/^/  FILTER smtp-amavis:[127.0.0.1]:10026
```

Save it and create a new file called `tag_for_scanning`:

```
sudo touch /Library/Server/Mail/Config/postfix/tag_for_scanning
```

Edit it and add:

```
/^/  FILTER smtp-amavis:[127.0.0.1]:10024
```

Save and next edit

```
/Library/Server/Mail/Config/postfix/master.cf
```

add a new block:

```
127.0.0.1:10027 inet n  -       y       -       -       smtpd
   -o content_filter=
   -o smtpd_tls_security_level=none
   -o smtpd_delay_reject=no
   -o smtpd_client_restrictions=permit_mynetworks,reject
   -o smtpd_helo_restrictions=
   -o smtpd_sender_restrictions=
   -o smtpd_recipient_restrictions=permit_mynetworks,reject
   -o smtpd_data_restrictions=reject_unauth_pipelining
   -o smtpd_end_of_data_restrictions=
   -o smtpd_restriction_classes=
   -o mynetworks=127.0.0.0/8
   -o smtpd_error_sleep_time=0
   -o smtpd_soft_error_limit=1001
   -o smtpd_hard_error_limit=1000
   -o smtpd_client_connection_count_limit=0
   -o smtpd_client_connection_rate_limit=0
   -o
receive_override_options=no_header_body_checks,no_unknown_recipient_checks,no_r

   -o local_header_rewrite_clients=
   -o smtpd_milters=
   -o local_recipient_maps=
   -o relay_recipient_maps=
```

Save it and now edit:

```
/Library/Server/Mail/Config/amavisd/amavisd.conf
```

Look for `$policy_bank{'ORIGINATING'}`

inside the policy bank block, add:

```
bypass_spam_checks_maps   => [1],
```

Save and issue:

```
sudo postfix reload


sudo -u _amavisd -H amavisd -c
/Library/Server/Mail/Config/amavisd/amavisd.conf reload
```

Now try sending mail from and to your server. Outgoing mail from authenticated users of yours should now be signed, but not scanned. Incoming mail from outside senders will be scanned but not signed.

The command we entered into the policy bank above prevents mail for outgoing mail from being scanned. You can of course add anything you like to the policy bank. For example instead of not scanning you could assign lower spam scores if mail is coming from your users. The amavisd-new documentation is a good starting point for this.

NOTE: The settings chosen are based on my personal preference and experience. You may want to change them as you deem fit.

6. Caveats

The 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.

Hope this helps.

Document Version 1.1, 15.11.2013