meumobi Dev Blog

Install and Configure DKIM with sendmail on Debian

SEPTEMBER 18TH, 2015

This tutorial will focus on installing and configuring DomainKeys Identified Mail (DKIM) an open source implementation of the DKIM sender authentication system. OpenDKIM is really useful for signing your mail messages by generated pair of private key which is afterwards validated by public key stored as TXT DNS record for lookup by other servers.

I assume you have properly installed and configured sendmail. Check if it’s running:

```
$ service sendmail status
```

Install OpenDKIM

```
$ aptitude install opendkim opendkim-tools
```

Create new directory for your keys

```
$ mkdir -p /etc/opendkim/keys/your-domain.com
```

Generate Signing Keys

You need to generate a privateand a public key for each of the domains for which you wish to sign mail. The private key is stored away from prying eyes on your server, while the public key gets published in your domain’s DNS records so that receiving mail servers can verify your DKIM-signed mail.

If you’re really hard-core, you can always build the keys manually. Or, you can use the easy script included with OpenDKIM to do it for you.

Before running this script, decide now what the name of yourselector is going to be. A selector is a unique keyword that is associated with both keys (public and private), included in all the signatures, and published in your DNS records. For simplicity, I use the word default as my default selector.

```
$ opendkim-genkey -D /etc/opendkim/keys/your-domain.com -d your-domain.com -s default
```

The above command will create two files under the newly created directory – default.private and default.txt You can do a man opendkim-genkey if you’re interested in what additional options are available when creating your keys. In this example, I used the -D (directory) option, the -d (domain) option, and the -s (selector) options.

Change the ownership to opendkim

```
$ chown -R opendkim:opendkim /etc/opendkim/keys/your-domain.com
$ chmod 640 /etc/opendkim/keys/your-domain.com/default.private
$ chmod 644 /etc/opendkim/keys/your-domain.com/default.txt
```

Configure OpenDKIM

A couple of files must be created and edited in order to configure OpenDKIM. Open the openDKIM configuration file /etc/opendkim.conf , and set following options:

```
AutoRestart             Yes
AutoRestartRate         10/1h
UMask                   002
Syslog                  yes
SyslogSuccess           Yes
LogWhy                  Yes

Canonicalization        relaxed/simple

ExternalIgnoreList      refile:/etc/opendkim/TrustedHosts
InternalHosts           refile:/etc/opendkim/TrustedHosts
KeyTable                refile:/etc/opendkim/KeyTable
SigningTable            refile:/etc/opendkim/SigningTable

Mode                    sv
PidFile                 /var/run/opendkim/opendkim.pid
SignatureAlgorithm      rsa-sha256

UserID                  opendkim:opendkim

Socket  inet:8891@127.0.0.1

Domain your-domain.com
Selector default
```

To learn about options do `man opendkim.conf` or go here.

Next, you'll need to create the three files you just declared on opendkim.conf. Create and Open /etc/opendkim/KeyTable and add the following line:

```
default._domainkey.your-domain.com your-domain.com:default:/etc/opendkim/keys/your-domain.com/default.private
```

The KeyTable file tells OpenDKIM where to find your keys. Each entry in the KeyTable file is a single line for each key location (for example, all of the text in the above example should be on a single line in your file). If you're going to use multiple keys (to sign mail for virtual domains with different keys, for example), you'll need to create a separate line in the KeyTable file for each domain, like this:

```
default._domainkey.your-domain.com your-domain.com:default:/etc/opendkim/keys/your-domain.com/default.private
default._domainkey.your-domain2.com your-domain2.com:default:/etc/opendkim/keys/your-domain2.com/default.private
```

Create and Open /etc/opendkim/SigningTable and add the following lines:

```
*@your-domain.com default._domainkey.your-domain.com
# contact@your-domain2.com default._domainkey.your-domain2.com
# no-reply@your-domain2.com default._domainkey.your-domain2.com
```

This SigningTable file is used for declaring the domains/email addresses and their selectors. I'm saying that everyone (*) sending mail from the server "your-domain.com" should use the selector named "default". But, if you uncomment last two lines, only contact and no-reply can sign mail for "your-domain2.com" (also using a selector named default). It's important to note that the * wildcard symbol will only work if the SigningTable option uses the refile: prefix before the filename (see the opendkim.conf documentation for more details).

Create and Open /etc/opendkim/TrustedHosts and add the following lines:

```
127.0.0.1
localhost
your-domain.com
```

The TrustedHosts file tells OpenDKIM who to let use your keys. Because it's referenced by the ExternalIgnoreList directive in your conf file, OpenDKIM will ignore this list of hosts when verifyingincoming mail. And, because it's also referenced by theInternalHosts directive, this same list of hosts will be considered "internal," and OpenDKIM will sign their outgoing mail.

Tell sendmail about OpenDKIM

Edit the .mc configuration file (used to be on /etc/mail/sendmail.mc) that was used to build your current sendmail.cf file. Add the following line:

```
INPUT_MAIL_FILTER(`opendkim', `S=inet:8891@127.0.0.1')
```

Then build and install a new sendmail.cf by restarting sendmail with following command:

```
$ sendmailconfig
```

Start OpenDKIM and restart sendmail

It's time to fire things up! Assuming you're using bash, do `hash -r` to rehash your shell so you can find the init script.

Now start OpenDKIM with:

```
$ service opendkim start
```

Check opendkim is listening right port

```
$ netstat -nlp | grep 8891

tcp        0      0 127.0.0.1:8891          0.0.0.0:*               LISTEN      51
15/opendkim
```

and Sendmail users should do:

```
$ service sendmail restart
```

If everything looks good, I recommend running chkconfig (system tool to enable or disable system services, if not installed run aptitude install chkconfig) on OpenDKIM to make sure it starts when you boot your server:

```
$ chkconfig opendkim on
```

Adding DNS Records

Now that your mail server is signing outgoing mail and verifying incoming mail, you'll need to put some information in your DNS records to tell other mail servers how your keys are set up, and provide the public key for them to check that your mail is properly signed. Do:

```
$ cat /etc/opendkim/keys/your-domain.com/default.txt
default._domainkey IN TXT "v=DKIM1; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBg
QDHp6PyqC13hdWHKeD4lrX8okexumtWnZp2T80GYrYfHNAsCbD5r7CCdwYK850GcCawALpQpM2977HlktA
379Rt6kYZxsSCAA+D27PekdRe6NGD4WgQF2S1S7dJvgCB9Zus1x/rfsE22DVgigBim5qfc8X+4g4V7eBK7
e2Xn+TDvQIDAQAB" ; ----- DKIM key default for your-domain.com
```

If you're using a web-based DNS interface (like GoDaddy or CloudFlare), the Name of the TXT record would default._domainkey and the Value of the TXT record would beeverything from the first quote to the last quote (starting with "v=).

Testing Things Out

The best way to see that everything is working on the server side is to keep an eye on your/var/log/mail.log file.

When you send a mail that gets successfully signed, you should see:

```
opendkim[5115]: t8IFlgM8005195: DKIM-Signature header added (s=default, d=your-dom
ain.com)
```

To check that your signed mail is being authenticated and that your DNS records are properly set up send a signed email to free testing service sa-test@sendmail.net

```
$ sendmail sa-test@sendmail.net
From: your-username@your-domain.com
To: sa-test@sendmail.net
Hi
.
```

Et voila !!

Sources: http://www.knowledgepia.com/en/k-blog/linux-security/installing-opendkim-rpm-with-postfix-or-sendmail-rhel-centos-fedora

sendmail (1)

DKIM (1) , spam (1) , mail (1)

Share Post

Twitter Facebook Google+

Victor Dias

Sharing mobile Experiences

Follow me

← Previous

Next →