How to enable Domain Keys (DKIM) in Qmail toaster based mail server install on Debian Linux

Author : admin

Recently the Emails sent by one of the *Qmail mail servers running on a Debian host started suddenly delivering in Spam* folder in both **Gmail.com** and **yahoo.com** public mail services.

This is pretty nasty as many of the websites which used the local qmail server to deliver emails concerning subscriptions and other kind of services provided by the websites started ending in **Span** and thus many of the users who used their *Yahoo Mail* account and *Google Mail - gmail* accounts was unable to read emails mailed by the various websites forms and scripts which were sending emails.
You can imagine the negative effect all this "minor" mail issues had on website visitors count and the overall websites functionality.
To come up with some kind of solution to this mail issues, I did quite a lot of research to understand if *Yahoo and Google Mail services* has some kind of **mail server delist form** or **some reporting service** where one can delist a specific mail server as a spammer one or get some kind of help, but unfortunately it seems neither google nor yahoo has any kind of web based way to remove hosts or ip addresses of legit mail servers who has mistakenly been recognized as spam servers.

During my efforts to find a solution to the situation I red a lot of posts and forums online as well as **Google's Bulk Sender Guidelines**, none if it was too helpful though.

The QMAIL server had a proper:
1. MX Record
2. TXT SPF records
3. PTR Record
4. There are proper correct mail message headers
5. Proper mails charset and encoding
6. The mail server IP is not listed anywhere in any mail blacklists
(e.g. www.mxtoolbox.com/blacklists.aspx / spamhaus.org)
7. A correct SMTP greeting which matched the mail server domain name

The only thing which was missing on the mail server (checked against Google's Bulk Sender Guidelines) was a properly configured DKIM and Domainkeys.

Thus in order to get around the situation I went the way and configured the qmail server to include and send in the mail header also **Domain Keys**

In this article I will briefly explain step by step how I configured **Domain keys (DKIM) signing** of my mails:

There are few ways **domain keys** signing can be implemented with *Qmail*.

1. By patching qmail binaries to support domain keys signing

I wanted to omit any interventions concerning the well running qmail install so I decided not to go this way.
Plus there are plenty of add-ons for qmail and as I have no time to test them the idea not to temper the existing qmail installation looked wise to me.
2. Use a wrapprer script around qmail-remote that invokes externally domainkeys binaries

This kind of solution was fitting me better and therefore I took this route to enable my qmail DKIM signing.

There are few approaches one can take described online:

- Use a perl script wrapper which does does the DKIM signing (*http://manuel.mausz.at/coding/qmail-dkim/*)

I tried using the **qmail-dkim-0.2.pl** wrapper script following the exact steps described to be fulfilled to enable my outgoing mails dkim signature, however for some reason after substituting the qmail-remote with qmail-dkim.pl and setting the proper permissions, my outgoing mails failed completely and each mail I sent was returned back by the *qmail MAILER-DAEMON*

- Use a bash shell script wrapper in combination with **libdomainkeys**'s with a Mail-DKIM-0.39 .

I gave a try to this approach and thanksfully it worked after a bit of struggle to tune it up.

Here is what exactly I had to do to in order to have the domain keys signing to work using the above described **qmail-remote.sh shell script wrapper**

1. Install openssl related required debian packages

debian:~# apt-get install openssl libcrypt-openssl-rsa-perl libcrypt-openssl-bignum-perl \
libmail-dkim-perl
...

2. Create necessery directories and RSA key pairs for DomainKeys

```
debian:~# mkdir -p /etc/domainkeys/mydomain.com
debian:~# cd /etc/domainkeys/mydomain.com
debian:/etc/domainkeys/mydomain.com# openssl genrsa -out rsa.private_default 768
debian:/etc/domainkeys/mydomain.com# openssl rsa -in rsa.private_default \
-out rsa.public_default -pubout -outform PEM
debian:/etc/domainkeys/mydomain.com# ln -sf rsa.private_default default
debian:/etc/domainkeys/mydomain.com# touch selector
debian:/etc/domainkeys/mydomain.com# echo 'default' >> selector
```

Where *mydomain.com* is the mail domain I need the DKIM signatures for.

I have written a small shell script which automates the task of adding new domainkeys directories and generating the RSA keys with openssl, you can download my generate_qmail_domainkey_rsa automator script here

3. Set proper permissions and owner to /etc/domainkeys directory

```
debian:~# chmod -R 0600 /etc/domainkeys
debian:~# chown -R qmailr:qmail /etc/domainkeys
```

4. Generate public domain key for DNS TXT records

```
debian:/etc/domainkeys/mydomain.com# grep -v ^- rsa.public_default | perl -e 'while(){chop;$l.=$_;}print
"k=rsa; t=y; p=$l;\n";'

"k=rsa; t=y; p=MHwwDQYJKoZIhvcNAQEBBQADawAwaAJhAMlDcYMrpWP9ouQOlFVtCHcFY
+gxrSQ6SegYeP4eeG7NECT/3jBqDtxANIVhaS9ASkEO4yNisGu4yX/DRclTm
nPWknoDtCDiD7IFEzT37qn1JLzcuknTncmFBFMDRUJq6wIDAQAB;"
```

The above key is used in next step 5 to set it as a **TXT** DNS record.

5. Create the DNS records in Name server

With BIND DNS server you need to place a records like:

```
_domainkey.example.com. IN TXT "k=rsa; t=y; o=-;"
default._domainkey.example.com. IN TXT "
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC2RkvFHbhqM/bVbb
kBtZ1cZUSYcjC4q+PWjd1tFopT+HXXR9Ctx7FZ1guX5fGboiwYmhCYdVroI
KRM3I48/YyQoXxtn3iYZ086v8BHaNtkcBMY+68JeEQ3K0WQkbqQXp/tsnLY
```

SQW1yXiEo9CywxVdpwH+OY94HxK4fAbw6V11cwIDAQAB"

! The above **p=** key specified is the one generated in **step 5.**

6. Download and compile & install Mail-DKIM-0.39 **'s perl extension**

As of time of writting latest Mail-DKIM is ver. 0.39, however it's a good idea to check and install the latest available version available on *http://www.cpan.org*

a) Download Mail-DKIM

```
debian:~# cd /usr/local/src
debian:/usr/local/src# wget http://www.pc-freak.net/files/Mail-DKIM-0.39.tar.gz
...
2011-05-25 15:09:37 (264 KB/s) - `Mail-DKIM-0.39.tar.gz' saved [87375/87375]
...
```

b) Compile & Install Mail-DKIM

```
debian:/usr/local/src# chown -R hipo:hipo Mail-DKIM-0.39
debian:/usr/local/src# cd Mail-DKIM-0.39

debian:/usr/local/src/Mail-DKIM-0.39# su hipo
debian:/usr/local/src/Mail-DKIM-0.39$ perl Makefile.PL
debian:/usr/local/src/Mail-DKIM-0.39$ make
...
debian:/usr/local/src/Mail-DKIM-0.39$ exit
debian:/usr/local/src/Mail-DKIM-0.39# make install
debian:/usr/local/src/Mail-DKIM-0.39# cd script
debian:/usr/local/src/Mail-DKIM-0.39/script# cp -rpf * /usr/local/bin; cd /usr/local/src
```

Note that the **dkimsign.pl** which is in the *Mail-DKIM-0.39* is a very important tool used later by the qmail-remote wrapper script. This perl script is copied in the last command issued in above chunk of code.
In the up-command lines I use my unprivileged username *hipo* to compile, here use any non-root user is appropriate.
For instance it's possible that the *cpan* user is used as a compile time user, I was lazy to configure CPAN thus I choose to use my normal unprivileged user.

c) configure rsa domain key paths in **dkimsing.pl**

Another thing to do here is to make sure the */usr/local/bin/dkimsign.pl* which was just recently installed has a correct set location for it's **KeyFile** variable.
This vailable is in the script is located online 64, I changed it to include my rsa domain key file, after I changed it, now it looks like so:

KeyFile => "/etc/domainkeys/mydomain.com/default"

7. Download and install libdomainkeys

a) Download libdomainkeys
For latest version of libdomainkeys make sure you check on *http://domainkeys.sourceforge.net/*

```
debian:/usr/local/src# wget http://www.pc-freak.net/files/libdomainkeys-0.69.tar.gz
debian:/usr/local/src# tar -zxvvf libdomainkeys-0.69.tar.gz
...
debian:/usr/local/src# chown -R hipo:hipo libdomainkeys-0.69
debian:/usr/local/src# cd libdomainkeys-0.69; su hipo
```

b) Compile and install libdomainkeys binaries

```
debian:/usr/local/src/libdomainkeys-0.69$ echo '-lresolv' > dns.lib
debian:/usr/local/src/libdomainkeys-0.69$ make clean & & make
debian:/usr/local/src/libdomainkeys-0.69$ exit
debian:/usr/local/src/libdomainkeys-0.69# cp -rpf dktest dknewkey expected makeheader /usr/local/bin/
```

There is a note to make here, one of the programs part of libdomainkeys called **dnstest** is not compiled while doing *make* for unknown reasons?!
I was not able to compile manually dnstest either using gcc like so:

```
debian:/usr/local/src/libdomainkeys-0.69$ gcc -o dnstest dnstest.c dnstest.c: In function 'main':
dnstest.c:11: warning: incompatible implicit declaration of built-in function 'strle'
/tmp/ccH78KZ1.o: In function 'main':
dnstest.c:(.text+0x2b): undefined reference to 'dns_text'
collect2: ld returned 1 exit status
```

I have absolutely no clue why it fails o_O, but it doesn't matter since I figured out that domainkeys header signature is properly set even without dnstest.

Let me mark here that **echo '-lresolv' > dns.lib** you see in above code chunk is absolutely necessery in order to be able to compile libdomainkeys on Debian based distributions. If the '-lresolv > dns.lib' is omitted the libdomainkeys build fails with error:

```
gcc -DBIND_8_COMPAT -O2 -o dktest dktest.o -L. -ldomainkeys -lcrypto
`cat dns.lib` `cat socket.lib`
./libdomainkeys.a(dns_txt.o): In function `dns_text':
dns_txt.c:(.text+0x2d): undefined reference to `__res_query'
dns_txt.c:(.text+0xc4): undefined reference to `__dn_expand'
```

```
dns_txt.c:(.text+0x184): undefined reference to `__dn_expand'
collect2: ld returned 1 exit status
make: *** [dktest] Error 1
```

8. Install libdkim (source of the libdkimtest binary later used by qmail-remote wrapper script)

```
debian:/usr/local/src# su hipo
debian:/usr/local/src$ wget http://www.pc-freak.net/files/qmail/libdkim-1.0.19.zip
debian:/usr/local/src$ wget http://www.pc-freak.net/files/qmail/libdkim-1.0.19-linux.patch
debian:/usr/local/src$ wget http://www.pc-freak.net/files/qmail/libdkim-1.0.19-extra-options.patch
debian:/usr/local/src$ unzip libdkim-1.0.19.zip
debian:/usr/local/src$ cd libdkim/src
debian:/usr/local/src/libdkim/src$ patch -p2 debian:/usr/local/src/libdkim/src$ patch -p2
debian:/usr/local/src/libdkim/src$ make && exit
debian:/usr/local/src/libdkim/src# make install
```

The above install will install **libdkimtest** binary, used by the wrapper script to do the actual DKIM-Signature, the binary gets installed in /usr/local/bin/libdkimtest.
Here is a link to patched version of libdkim 1.0.19 , to use it instead of patching as described above download the archive untar and do a **make clean && make && install**

9. Download qmail-remote.wrapper (qmail-remote wrapper shell script) and set it to wrap qmail-remote

a) Copy original qmail-remote to qmail-remote.orig

```
debian:~# cd /var/qmail/bin
debian:/var/qmail/bin# cp -rpf qmail-remote qmail-remote.orig
```

b) Download qmail-remote.wrapper script

Here is the qmail-remote.sh wrapper script that worked for me
Originally the wrapper script is taken from *http://www.memoryhole.net/qmail/*, big thanks to **Russ Nelson** for writting the awesome wrapper script.

```
debian:~# cd /var/qmail/bin/
debian:/var/qmail/bin# wget http://www.pc-freak.net/files/qmail-remote.wrapper
Saving to: `qmail-remote.wrapper'

100%[============================>] 1,164 --.-K/s in 0s

2011-05-25 15:46:54 (142 MB/s) - `qmail-remote.wrapper' saved [1164/1164]
```

c) Set proper permissions to the qmail-remote.wrapper script

The permissions of qmail-remote should look like so:

*-rwxr-xr-x 1 root qmail 1164 2011-05-25 11:05 /var/qmail/bin/qmail-remote**

To set this permissions I used:

```
debian:/var/qmail/bin# chmod 755 qmail-remote.wrapper
debian:/var/qmail/bin# chown qmailq:qmail qmail-remote.wrapper
```

d) Create /var/domainkeys directory (necessery for proper qmail remote wrapper script operations)

```
debian:~# mkdir /var/domainkeys
debian:~# chown -R qmailr:qmail /var/domainkeys
debian:~# chmod 700 -R /var/domainkeys
```

e) Substitute original **qmail-remote** binary with the wrapper script:

```
debian:~# qmailctl stop
Stopping qmail...
qmail-send
qmail-smtpd
debian:/var/qmail/bin# cp -rpf qmail-remote.wrapper qmail-remote
debian:/var/qmail/bin# qmailctl start
Starting qmail
```

10. Send test email to @gmail.com or @yahoo.com to test if DKIM-Signature is included in the mail header

I used my installed webmail interface **squirrelmail** and send a test email to my home mail server and as well as to yahoo.com

The headers of the email looked fine, here is how my DKIM signed mail headers looked like:

```
From - Thu May 26 15:31:37 2011
X-Account-Key: account11
X-UIDL: 1306413169.97071.pcfreak,S=1244
X-Mozilla-Status: 0001
X-Mozilla-Status2: 00000000
X-Mozilla-Keys:

Return-Path:
```

Delivered-To: hipo@pc-freak.net
Received: (qmail 97068 invoked by uid 1048); 26 May 2011 12:32:49 -0000
Received: from mail.mydomain.com (83.170.100.100)
by mail.pc-freak.net with SMTP; 26 May 2011 12:32:49 -0000
Comment: DomainKeys? See http://domainkeys.sourceforge.net/
DomainKey-Signature: a=rsa-sha1; q=dns; c=nofws;
s=default; d=mydomain.com;

b=ZnTDdUexnt8fmuHbVNXIC+JDvNLYO1zjzlI3PODe3e1oMS5dRHzVGujrS1
Yk0qqs2oW7DseZg/iHE9KOLZBeInksOnsmLsDBq1Lvzfv2xejikR52LBg6a/uK
ewECJy4jQA4cwMJ/qUxmK8EDwbgj7jqCVB95FK3Z5EdR4HoaqGQ=;

h=Received:Date:Message-ID:From:To:Subject;
DKIM-Signature: v=1; a=rsa-sha1; c=relaxed; d=mydomain.com; h=date
:message-id:from:to:subject; s=default; q=
/etc/domainkeys/mydomain.com/default;
bh=G32d6y8oiLehRzcuIWr9s
S+Jy+g=;
b=TPW5VXq3vlOUf1T7lfxQC00MN0kPJxaASE/gq7LbHyV1Gj/Xj+GLF
UN6hYVeKnsoKKeV108JVGcfvfTaLogsxGyS9XzUXKlLtESBj4wr/DAOQy
OcHCj75
bEOOd9nv+RehOYinXGmx0JUZpCNHGndNZ1AEabbVEiX/NQAL7iKDnE
=
Received: (qmail 28771 invoked by uid 0); 26 May 2011 12:31:30 -0000
Date: 26 May 2011 12:31:30 -0000
Message-ID:
From: root@mail.mydomain.com
To: hipo@pc-freak.net
Subject: testing 123

baklavavav
tatta

Notice the two lines in the above pasted header:

DomainKey-Signature: a=rsa-sha1; q=dns; c=nofws;
DKIM-Signature: v=1; a=rsa-sha1; c=relaxed; d=mydomain.com; h=date

This clearly shows that now both Domainkeys and DKIM are being applied to outgoing mail messages :)

However to be completely 100% sure the *domainkeys and DKIM* signing is correct, you should check the online websites which offer a mail domainkey check and DKIM signature check, an example for such a website that I used in order to test that my SSL RSA Domainkeys and DKIM correspond correctly to the ones specified in the DNS server is:
here

The idea for writting this small guide on configuring Domainkeys with Qmail and Linux is seriously inspired by Mariuz's Blog post dkim wrapper that works using dk Hope this is helpful to somebody, it took me quite a while until I come up with the exact steps of a workable install of Domain Keys, there are so many tutorials and ways to implement this that at a certain point it's a hell.
Like always with Qmail, even simple things are so complex, the only good thing about qmail is once you make it work well, it works forever ... until the next time you will have to spend few days trying to figure it out ;)

I'm very much looking to hear if people followed the tutorial succesfully.

Any feedback concerning the article is mostly welcome!
Cheers!