Email Best Practices, Postfix and Dovecot

Kevin Chege

SPF

- SPF – Sender Policy Framework
  - SPF allows administrators to specify which hosts are allowed to send mail from a given domain by creating a specific SPF record (or TXT record) in the Domain Name System (DNS).
- *@ IN TXT "v=spf1 include:gmail.com ip4:1.2.3.4 mx -all"*
- The above will only allow mail from IP 1.2.3.4 and any server in the domain with an MX record
- If not sure use a generation tool online
  - http://www.mtgsy.net/dns/spfwizard.php

Reverse Records

- Have reverse records (PTR) for your mail server so that it is resolveable from the IP
- Mandatory by most servers these days
- Used to verify authenticity of the sending mail server
- The IP Address must resolve back to the mail server name
- You can have multiple reverse records
- You can have an SPF record that states that any IP that has a reverse record can send email from your domain
- *IN TXT "v=spf1 ptr:domain.co.tz ip4:1.2.3.4 mx -all"*

Use Anti Spam and Anti Virus software

- Will reduce overall spam and email received
- You can also have a mail “firewall” or gateway aka Mail Filter to stop spam before it reaches your server
- Some softwares are:
  - Spamassassin (AntiSpam)
  - ClamAV (AntiVirus)
  - MailScanner and Amavisd (use the above)
  - Maia Mail Guard
- When setup try a penetration testing site to see how well your server can protect you from SPAM and Viruses

GreyListing

- Valid mail servers will have no problem if the receiving gives a soft error (4xx)
- They will attempt to send the mail again after some time
- Greylisting configured on a receiving mail server will give a soft error (4xx) to the sending server and store the IP/Hostname of the sending server in a file
- If the sending server returns again after some time (can be specified usually 5min) the email is accepted
- Used as a measure to deny mail from bots that are compromised to send mass mail. They often do not try again if the server did not accept the mail

Accept only well formatted messages

- Sender must be a valid name not an IP ie not user@192.14.5.6
- Mail server HELO name must be resolvable ie FQDN
- Server identification must resolve ie HELO/ EHLO name must be resolveable
- Email should be from a valid email address format eg: from tom@example.com and not from tom@example

Security

- Run secure pages from the mail server and secure SMTP to clients
  - Secure Webmail – port 443
  - Secure SMTP – port 465/587
- Force clients to use secure IMAP or Secure POP
  - Secure POP – port 995
  - Secure IMAP – port 993
- Require authentication on your mail server before a mail enters the queue from a sending client aka SMTP AUTH

Use Blacklist databases

- Use DNSBL – DNS Based Blackhole Lists or RBL (Real Time Blackhole lists) to deny mail from well known spamming machines
- Some well known good ones are
  - SORBS – http://sorbs.net
  - SPAMHAUS – http://spamhaus.org
  - SPAMCOP – http://spamcop.net
  - MANITU – http://manitu.net

Require strong Passwords

- Advise users to use strong passwords or passphrases for their email
- Alphanumeric passwords are better than normal passwords ie combine letters with numbers
- Passphrases are even better, more difficult to break

Backup and Redundancy

- Have multiple MX records so that your server is not the only able to receive mail for you
- Backup your mail, use tools like Rsync to copy mail to another server as often as you can
- Ensure your DNS records (MX, NS etc) are correct and test them when you complete you setup
- Use online tests like
  - http://intodns.net

References

- Wikipedia
- http://www.linuxmagic.com/best_practices

Postfix Mail Server

Kevin Chege
ISOC

What is Postfix?

- **Postfix** is a free and open-source mail transfer agent (MTA) that routes and delivers electronic mail, intended as an alternative to the widely used Sendmail MTA.
- Postfix is released under the IBM Public License 1.0 which is a free software licence.
- Originally written in 1997 by Wietse Venema at the IBM Thomas J. Watson Research Center and first released in December 1998, Postfix continues as of 2014 to be actively developed by its creator and other contributors. The software is also known by its former names **VMailer** and **IBM Secure Mailer**.
- In January 2013 in a study performed by E-Soft, Inc. found that approximately 25% of the publicly reachable mail-servers on the Internet ran Postfix.

Postfix

- Works on UNIX-like systems including AIX, BSD, HP-UX, Linux, MacOS X, Solaris, and more.
- It is the default MTA for the OS X, NetBSD[3] and Ubuntu operating systems
- Used by: AOL, Apple Server, Stanford University, United States Navy, NASA, Rackspace, many ISPs

Some Key Features

- SASL authentication
- Mail forwarding or delivery
- "Virtual" domains with distinct address-namespaces
- A large number of database lookup mechanisms including Berkeley DB, CDB, OpenLDAP LMDB, Memcached, LDAP and multiple SQL database implementations
- Extended
  - Deep content inspection before or after a message is accepted into the mail queue;
  - Mail authentication with DKIM, SPF, or other protocols;
  - SMTP-level access policies such as greylisting or rate control.

Postfix on FreeBSD

- Installed via: **$sudo pkg install postfix**
- Directories:
  /usr/local/etc/postfix
- Configuration files
  - main.cf - stores site specific Postfix configuration parameters while
  - master.cf – defines daemon processes

main.cf

- specifies a very small subset of all the parameters that control the operation of the Postfix mail system
- you will have to set up a minimal number of configuration parameters.
- Postfix configuration parameters resemble shell variables
  - parameter = value
  - other_parameter = $parameter
- Postfix uses database files for access control, address rewriting and other purposes

main.cf Key Settings

- myorigin = $myhostname
  - specifies the domain that appears in mail that is posted on this machine. Defaults to the value of the machine's hostname
- mydestination = $myhostname, localhost
  - specifies what domains this machine will deliver locally
  - if your machine is a mail server for its entire domain, you must list $mydomain as well in this setting
- The mydomain parameter specifies the parent domain of $myhostname. By default, it is derived from $myhostname by stripping off the first part (unless if the result would be a top-level domain)

Relaying Mail – From

- Postfix will forward mail from clients in authorized network blocks to any destination
- Authorized networks are defined with the mynetworks configuration parameter
- The default is to authorize all clients in the IP subnetworks that the local machine is attached to.
- By default, Postfix will NOT be an open relay ie it will not forward from IPs outside your network to the Internet
  - mynetworks_style = subnet
  - mynetworks = 127.0.0.0/8 168.100.189.2/32

Relaying mail - to

- By default, Postfix will forward mail from strangers (clients outside authorized networks) to authorized remote destinations only.
- Authorized remote destinations are defined with the relay_domains configuration parameter.
- The default is to authorize all domains (and subdomains) of the domains listed with the mydestination parameter.
- This means that by default, your Postfix mail server will accept mail from anyone to recipients to the local Postfix server

Outbound emails

- By default, Postfix tries to deliver mail directly to the Internet.
- Depending on your local conditions this may not be possible or desirable
- For example, your system may be behind a firewall, or it may be connected via a provider who does not allow direct mail to the Internet.
- In those cases you need to configure Postfix to deliver mail indirectly via a relay host.
  - relayhost = [mail.isp.tld]
  - Note that the [] disables MX lookups so is necessary

Reporting problems

- You should set up a postmaster alias in the aliases table that directs mail to a real person
- The postmaster address is required to exist, so that people can report mail delivery problems.
- While you're updating the aliases(5) table, be sure to direct mail for the super-user to a human person too.
  /etc/aliases:
  postmaster: afnog
  root: afnog
- After editing the aliases file, run the command *$sudo newaliases*

Default reports

- bounce
  - Inform the postmaster of undeliverable mail. Either send the postmaster a copy of undeliverable mail that is returned to the sender, or send a transcript of the SMTP
- 2bounce
  - When Postfix is unable to return undeliverable mail to the sender,
- delay
  - Inform the postmaster of delayed mail. In this case, the postmaster receives message headers only.
- policy
  - Inform the postmaster of client requests that were rejected because of (UCE) policy restrictions. The postmaster receives a transcript of the SMTP session.
- protocol
  - Inform the postmaster of protocol errors (client or server side) or attempts by a client to execute unimplemented commands.
- resource
  - Inform the postmaster of mail not delivered due to resource problems (for example, queue file write errors)
- software
  - Inform the postmaster of mail not delivered due to software problems.

Logging

- Postfix will log all messages to /var/log/maillog
- Done using the syslogd daemon
- All transactions of messages coming in being sent out of the server will be logged
- Logs will contain details like hostnames, recipients, time and date, and whether the email was queued or dropped

Postfix Daemon process chrooted

- Postfix daemon processes can be configured (via the master.cf file) to run in a chroot jail
- The processes run at a fixed low privilege and with file system access limited to the Postfix queue directories (/var/spool/postfix).
- This provides a significant barrier against intrusion.
- The barrier is not impenetrable (chroot limits file system access only)

Interfaces and Protocol

- The inet_interfaces parameter specifies all network interface addresses that the Postfix system should listen on
  - inet_interfaces = all
- inet_protocols parameter specifies which protocols Postfix will attempt to use
  - inet_protocols = ipv4, ipv6

Starting, stopping and logs

- Starting/Stopping
  $sudo service postfix start
  $sudo service postfix xtop

- Reloading rules
  $sudo postfix reload
- Checking logs
  $sudo tail –f /var/log/maillog

POP and IMAP

What is POP3

- POP3 stands for Post Office Protocol ver 3
- Described in RFC1913
- Runs on TCP Port 110 as a client server function
- Allows for a maildrop service (similar to the post box mail service) hence the name
- By design its limited in features to download and delete email from server
- Security was also limited to using APOP (md5 hash for authentication
- RFC 2449 proposed POP3 extensions which included SASL Mechanism, Expiry, Pipelining, etc.
- RFC 2595 describes using TLS with POP3 also known as POP3s and runs on port 995

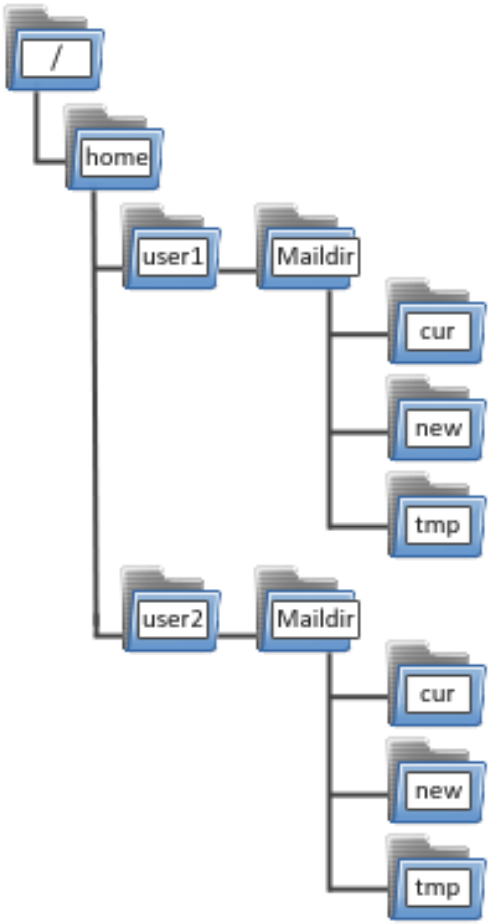
What is IMAP4?

- Internet Message Access Protocol version 4
- Described in RFC 1730
- Runs on TCP Port 143 as client-server function
- More advanced in features compared to POP3
- IMAP4 stores mail on server and copies can be transferred to the client on request.
- By default only the message headers are sent to the client, the rest of the message is accessed on opening the email.
- Allows client to access and manipulate email residing on a server, creation of folders, filters, etc.
- RFC 1731 describes the IMAP Authentication Mechanisms
- RFC 2595 describes using TLS with IMAP4 running on TCP port 993

Mail Storage Formats

- Mailbox Format (Mbox)
- Defined in RFC 4155
- All messages in an Mbox mailbox are concatenated and stored as a plain text in a single file
- Mails are stored in RFC822 format with a blank space separating each message (2 spaces as each message has one space) and “From” determining start of next message.
- Mbox has a distinct disadvantage in cases of large mailbox (a single large file) requires more resources to read/open and can be slow depending on the servers load.

Maildir Storage Format



- Mail Directory Format (Maildir)
- Each message is stored in a separate file with a unique name and each folder in a directory
- Maildir++ provides extension to the Maildir specification providing support for subfolders and quotas.
- Maildir directory has 3 folders **temp**, **new** and **current**

How Maildir Works

- The mail delivery agent stores all new emails to the mailbox in the tmp directory with a unique filename. (unique = time + hostname+ random generated number)
- The MDA creates a hard link to the file in tmp/ unique to new/unique
- The Mail User Agent will check for new emails in new folder and move them to current folder
- The MUA modifies the filename to add a colon (:), a ‘2’ and various flags to represent message status i.e read, replied, forwarded, deleted, etc
-

What is Dovecot?

- High-performance POP and IMAP server
- Developed by Timo Sirainen
- Unlike say UW IMAP it wasn't written in the 80s
- Transparently index's mailbox contents (Why is this important?)
- Supports both mbox and maildir formats
- Capable of operating in an environment with minimal locking. (Why is this important)
- Graceful around failures (index repair for example)
- Designed with Security in mind – support for Authentication Mechanism and SSL/TLS
- Install: `#cd pkg install dovecot`

Dovecot 2 Protocols Configuration

- Default dovecot config file
  - `$ sudo ee /usr/local/etc/dovecot/dovecot.conf`
- Note that the default listening services are:
  - `protocols = imap pop3 lmtp`
    - *The TCP listeners are on 110, 143, 993, and 995*
    - *If you need the unencrypted versions of the protocol for some reason (e.g. a webmail application) then you should firewall them off from the rest of your end users (end-user clients should never be be allowed to connect insecurely)*
- If you have working SSL Certificate (from Apache-SSL session), uncomment and add imaps and pop3s protocols as follows;

  `protocols = imap imaps pop3 pop3s lmtp`

- If you do NOT have working SSL Certificates, uncomment and retain the imap and pop3 as follows;

  `protocols = imap pop3 lmtp`

Dovecot 2 SSL Configuration

- ***If you do NOT have a working SSL Certificate, follow the next 3 steps***

1. ***1. Edit the file /usr/local/etc/dovecot/conf.d/10-ssl.conf and find line***
   - *# ssl = yes*
2. ***2. Uncomment the line and modify it to NO***
   - *ssl = no*
   - **3. Comment the following lines**
     - #ssl_cert = </etc/ssl/certs/dovecot.pem
     - #ssl_key = </etc/ssl/private/dovecot.pem

- **If you have SSL Certs Working during Apache Session, *edit the file / usr/local/etc/dovecot/conf.d/10-ssl.conf and find lines***
  - ssl_cert = </etc/ssl/certs/dovecot.pem
  - ssl_key = </etc/ssl/private/dovecot.pem

- ***MODIFY*** above lines and set ***PATH*** to point at the certificate and keyfile that was created during the apache tutorial. i.e
  - ssl_cert = </usr/local/etc/apache22/server.crt
  - ssl_key = </usr/local/etc/apache22/server.key

- Save and close the 10-ssl.conf file

Dovecot 2 Authentication Config

- ***Edit the file /usr/local/etc/dovecot/conf.d/10-auth.conf***
- Disable plaintext authentication by finding the line below
  - `#disable_plaintext_auth = no`
- Uncomment the line and Set the value to yes as below
  - `disable_plaintext_auth = yes`
    - Note: unencrypted connections can still be made from localhost!

Questions?