FreeBSD Administration - Support #653

Install Dovecot and Postfix on FreeBSD

09/01/2015 08:26 PM - Daniel Curtis

Status:	Closed	**Start date:**	09/01/2015
Priority:	High	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Mail Server	**Estimated time:**	8.00 hours
Target version:	FreeBSD 9	**Spent time:**	41.00 hours

Description

This is a guide on installing a Dovecot and Postfix mail server along with Nginx, PostgreSQL, Maia Mailguard, Postfixadmin, SpamAssassin, Distributed Checksum Clearinghouse, Sender Policy Framework, DomainKeys Identified Mail, and Fail2ban on FreeBSD 9.3. This guide is adapted from the excellent mail server setup at purplehat.org

Prepare the System

- Make sure the system is up to date:

```
pkg update && pkg upgrade
```

- Install portmaster and screen:

```
pkg install portmaster screen
```

- Update the ports tree:

```
portsnap fetch extract
pkg2ng
```

- This builds ClamAV to allow our vscan user access to it. Add ClamAV build options to /etc/make.conf file:

```
echo "CLAMAVUSER=vscan" >> /etc/make.conf
echo "CLAMAVGROUP=vscan" >> /etc/make.conf
```

- Add BATCH option to /etc/make.conf file:

```
echo "BATCH=yes" >> /etc/make.conf
```

- Edit pear-Net_SMTP installation menu:

```
cd /usr/ports/net/pear-Net_SMTP
make config
```

  - **NOTE**: Make sure **[X]PEAR_AUTH_SASL** is selected.

- Edit pear-Auth Options installation menu:

```
cd /usr/ports/security/pear-Auth
make config
```

    - **NOTE**: Make sure **[X]PEAR_DB** and **[X]PEAR_LOG** are selected.

- Edit pear-Log installation menu:

```
cd /usr/ports/sysutils/pear-Log
make config
```

    - **NOTE**: Make sure **[X]PEAR_DB** is selected.

- Edit Dovecot installation menu:

```
cd /usr/ports/mail/dovecot2
make config
```

    - **NOTE**: Make sure **[X]PGSQL** is selected.

- Edit Postfix installation menu:

```
cd /usr/ports/mail/postfix
make config
```

    - **NOTE**: Make sure **[X]BDB**, **[X]PGSQL**, **[X]SPF**, **[X]TLS**, **[X]VDA** and **[X]DOVECOT2** are selected.

- Edit Postfixadmin installation menu:

```
cd /usr/ports/mail/postfixadmin
make config
```

    - **NOTE**: Make sure **[X]PGSQL** is selected.

- Edit SpamAssassin installation menu:

```
cd /usr/ports/mail/spamassassin
make config
```

    - **NOTE**: Make sure **[X]PGSQL**, **[X]DKIM**, **[X]RAZOR**, **[X]RELAY_COUNTRY** and **[X]SPF_QUERY** are selected.

- Edit Maia-Mailguard installation menu:

```
cd /usr/ports/security/maia
make config
```

    - **NOTE**: Make sure the **[X]DOVECOT2**, **[X]FUZZYOCR**, **[X]PGSQL**, **[X]PFA**, **[X]POSTFIX** and **[X]WEBHOST** options are selected. Also make sure to unset the **[ ]MYSQL** option. Feel free to select any additional options you may want.

Install Maia Mailguard

- Install Maia-Mailguard:

```
portmaster security/maia
```

- Set password for “vscan” user to **SuperSecretPassword**:

```
passwd vscan
```

Install PostgreSQL

- This environment will be setup with PostgreSQL 9.4:

  ```
  portmaster databases/postgresql94-server
  ```

- **NOTE**: I needed to edit the main postgresql config file:

  ```
  vi /usr/local/pgsql/data/postgresql.conf
  ```

  - And change the bytea_output parameter to escape:

    ```
    bytea_output = 'escape'
    ```

Setup Maia Database

- Login to PostgreSQL:

  ```
  sudo -u postgres psql
  ```

  - Create a user for maiauser:

    ```
    CREATE USER maiauser WITH PASSWORD 'SuperSecretPassword';
    ```

  - Create the maiadb database & grant all privileges on database

    ```
    CREATE DATABASE maiadb OWNER maiauser;
    ```

Setup Postfix Database

- Create PostfixAdmin database, login to PostgreSQL:

  ```
  sudo -u postgres psql
  ```

  - Create a user for postfix:

    ```
    CREATE USER postfix WITH PASSWORD 'SuperSecretPostfixPassword';
    ```

  - Create the postfix database & grant all privileges on database

    ```
    CREATE DATABASE postfix OWNER postfix;
    ```

Setup Roundcube Database

- Create PostgreSQL database and user for Roundcube *, login to PostgreSQL:

  ```
  sudo -u postgres psql
  ```

  - Create a user for roundcubeuser:

    ```
    CREATE USER roundcubeuser WITH PASSWORD 'SuperSecretRoundcubePassword';
    ```

  - Create the roundcubedb database & grant all privileges on database

```
CREATE DATABASE roundcubedb OWNER roundcubeuser;
```

- Quit out of the postgresql prompt and exit out the postgresql user:

```
\q
exit
```

Configure Dovecot

Dovecot is an open source IMAP and POP3 email server for Linux/UNIX-like systems, written with security primarily in mind. Dovecot is an excellent choice for both small and large installations. It's fast, simple to set up, requires no special administration and it uses very little memory.

- Install Dovecot Pigeonhole:

```
portmaster mail/dovecot2-pigeonhole
```

- Edit /etc/rc.conf so Dovecot starts at boot:

```
echo 'dovecot_enable="YES"' >> /etc/rc.conf
```

- Copy Dovecot configuration files:

```
cd /usr/local/etc/dovecot/example-config
cp -Rp * ../
```

Auth config

- Edit the dovecot auth config file:

```
vi /usr/local/etc/dovecot/conf.d/10-auth.conf
```

  - And edit the following:

```
disable_plaintext_auth = no

auth_mechanisms = plain login

#!include auth-system.conf.ext
!include auth-sql.conf.ext
```

Mail config

- Edit the dovecot mail config file:

```
vi /usr/local/etc/dovecot/conf.d/10-mail.conf
```

  - And modify the following:

```
mail_location = maildir:/usr/local/virtual/%d/%n

namespace inbox {
  type = private
  separator = /

  mailbox Sent {
    auto = subscribe
    special_use = \Sent
  }
  mailbox Drafts {
    auto = subscribe
    special_use = \Drafts
  }
  mailbox Trash {
    auto = subscribe
    special_use = \Trash
  }
  mailbox Spam {
    auto = subscribe
    special_use = \Junk
  }

first_valid_uid = 110
last_valid_uid = 110

first_valid_gid = 110
last_valid_gid = 110

mail_plugins = mail_log notify
```

Master config

- Edit the dovecot master config file:

```
vi /usr/local/etc/dovecot/conf.d/10-master.conf
```

  - And modify the following

```
  unix_listener auth-userdb {
    mode = 0660
    user = vscan
    group = vscan
  }

  #Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }
```

LDA config

- Edit the dovecot lda config file:

```
vi /usr/local/etc/dovecot/conf.d/15-lda.conf
```

  - And modify the following:

```
postmaster_address = postmaster@example.com

hostname = mail.example.com

sendmail_path = /usr/local/sbin/sendmail

lda_mailbox_autocreate = yes

 protocol lda {
   # Space separated list of plugins to load (default is global mail_plugins).
  mail_plugins = $mail_plugins sieve
```

IMAP config

- Edit the dovecot imap config file:

```
vi /usr/local/etc/dovecot/conf.d/20-imap.conf
```

  - And modify the following:

```
 protocol imap {
   # Space separated list of plugins to load (default is global mail_plugins).
  mail_plugins = $mail_plugins quota imap_quota zlib
```

POP3 config

- Edit the dovecot pop3 config file:

```
vi /usr/local/etc/dovecot/conf.d/20-pop3.conf
```

  - And modify the following:

```
pop3_client_workarounds = outlook-no-nuls oe-ns-eoh

mail_plugins = $mail_plugins
```

Plugin config

- Edit the dovecot plugin config file:

```
vi /usr/local/etc/dovecot/conf.d/90-plugin.conf
```

  - And modify the following:

```
 plugin {
   #setting_name = value
  expire = Trash
  mail_log_events = delete undelete expunge copy mailbox_delete mailbox_rename
  mail_log_fields = uid box msgid size
}

plugin {
  sieve = /usr/local/virtual/home/%d/%n/.dovecot.sieve
  sieve_dir = /usr/local/virtual/home/%d/%n/sieve
  sieve_global_path = /usr/local/virtual/home/default.sieve
  mail_home = /usr/local/virtual/home/%d/%n
}
```

Quota config

- Edit the dovecot quota config file:

```
vi /usr/local/etc/dovecot/conf.d/90-quota.conf
```

  - And modify the following:

```
service quota-warning {
  executable = script /usr/local/bin/quota-warning.sh
  user = dovecot
  unix_listener quota-warning {
    user = vscan
  }
}

// Add to end of file...
plugin {
  #Where is quota applied ?
  quota = maildir:User quota
  # the default quota storage bytes, overrides are fetched from userdb [userdb_quota_ruleX]
  quota_rule = *:storage=1G
  #Storage bytes overrides
  quota_rule2 = Trash:storage=+30%%
  quota_rule3 = Sent:storage=+30%%
  quota_warning = storage=90%% quota-warning 90 %u
  quota_warning2 = storage=75%% quota-warning 75 %u
  #What message to send to IMAP clients (and SMTP senders) when quota is exceeded?
  quota_exceeded_message = Storage quota for this account has been exceeded, please try again later.
}
```

SQL config

- Edit the dovecot sql config file:

```
vi /usr/local/etc/dovecot/dovecot-sql.conf.ext
```

  - And modify the following:

```
driver = pgsql

connect = host=pg.example.com dbname=postfix user=postfix password=SuperSecretPostfixPassword

default_pass_scheme = MD5

password_query = SELECT password, '*:bytes=' || quota AS userdb_quota_rule FROM "mailbox" WHERE username = '%u' AND active = TRUE

user_query = SELECT '/usr/local/virtual/' || maildir as home, 110 AS uid, 110 AS gid, '*:bytes=' || quota AS quota_rule FROM mailbox WHERE username = '%u' AND active = TRUE
```

  - **NOTE**: The user_query line contains a bit in the query to allow Dovecot to return quota usage. If you don’t want or don’t need quota usage returned, you can just remove that bit from the query…

Main config

- Edit the main dovecot config file:

```
vi /usr/local/etc/dovecot/dovecot.conf
```

  - And modify the following:

    ```
    protocols = imap pop3 sieve

    login_greeting = example.com Mail Server Ready...
    ```

SSL config

- Edit the dovecot ssl config file:

```
vi /usr/local/etc/dovecot/conf.d/10-ssl.conf
```

  - And modify the following:

    ```
    ssl = yes

    ssl_cert = </usr/local/etc/ssl/mail.example.com.crt
    ssl_key = </usr/local/etc/ssl/mail.example.com.key

    ssl_ca = </usr/local/etc/ssl/postfix/sub.class1.server.ca.pem

    ssl_verify_client_cert = yes

    ssl_protocols = !SSLv2 !SSLv3

    ssl_cipher_list=ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA

    ssl_prefer_server_ciphers = yes

    # dhparam file regenerates every week
    ssl_dh_parameters_length = 2048
    ```

Setup SSL key

- Install openssl:

```
portmaster security/openssl
```

- Start by generating a new 2048-bit Diffie-Hellman param file:

```
cd /usr/local/etc/ssl
openssl dhparam -out dhparams.pem 2048
```

  - **NOTE**: This command will take some time to complete

- Create SSL/TLS key and CSR to have signed for a certificate for secure connections:

```
cd /usr/local/etc/ssl
openssl req -sha512 -out mail.example.com.csr -new -newkey rsa:4096 -nodes -keyout mail.example.com.key
```

- Send mail.example.com.csr to a Certificate Authority to receive a signed certificate created, then create the certificate file:

```
vi mail.example.com.crt
```

  - **NOTE**: I use StartSSL, but there are many different CAs to choose from.

- Once the SSL certificate has been created on the mail server, download and add the intermediate certificate to the SSL certificate:

```
cd /usr/local/etc/ssl
fetch https://www.startssl.com/certs/sub.class1.server.ca.pem
cat sub.class1.server.ca.pem >> mail.example.com.crt
```

Virtual Mail User

- Create Sieve home directory:

```
mkdir -p /usr/local/virtual/home
```

- Create the default.sieve file:

```
vi /usr/local/virtual/home/default.sieve
```

  - And add the following:

```
require ["fileinto"];
# rule:[Spam]
if header :contains "X-Spam-Status" "Yes" 
{
  fileinto "Spam";
  stop;
}
```

- Run the sievec command against our default sieve file:

```
sievec /usr/local/virtual/home/default.sieve
```

- Set proper permissions on our virtual directory:

```
chown -R vscan:vscan /usr/local/virtual
chmod 0750 /usr/local/virtual
```

- Start dovecot:

```
service dovecot start
```

Configure Postfix

Postfix attempts to be fast, easy to administer, and secure. The outside has a definite Sendmail-ish flavor, but the inside is completely different.

- Disable Sendmail and start Postfix at boot:

```
echo 'sendmail_enable="NO"' >> /etc/rc.conf
echo 'sendmail_submit_enable="NO"' >> /etc/rc.conf
echo 'sendmail_outbound_enable="NO"' >> /etc/rc.conf
echo 'sendmail_msp_queue_enable="NO"' >> /etc/rc.conf
echo 'postfix_enable="YES"' >> /etc/rc.conf
```

- Create and add Postfix stuffs to the /etc/periodic.conf file:

```
echo 'daily_clean_hoststat_enable="NO"' >> /etc/periodic.conf
echo 'daily_status_mail_rejects_enable="NO"' >> /etc/periodic.conf
echo 'daily_status_include_submit_mailq="NO"' >> /etc/periodic.conf
echo 'daily_submit_queuerun="NO"' >> /etc/periodic.conf
```

Main config

- Edit the main postfix config file:

```
vi /usr/local/etc/postfix/main.cf
```

  - And modify the following:

```
soft_bounce = no

# Adjusted message size limit to 25MB.
message_size_limit = 25600000

myhostname = mail.example.com

mydomain = example.com

mynetworks = 192.168.0.0/24, 127.0.0.0/8

mydestination = localhost.$mydomain, localhost, mail.example.com

relay_domains = proxy:pgsql:/usr/local/etc/postfix/pgsql_relay_domains_maps.cf

relay_recipient_maps = proxy:pgsql:/usr/local/etc/postfix/pgsql_virtual_mailbox_maps.cf

# Add to end of file
#
# SASL CONFIG
broken_sasl_auth_clients = yes
smtpd_sender_restrictions = permit_sasl_authenticated, permit_mynetworks
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
smtpd_recipient_restrictions =
  permit_mynetworks,
  permit_sasl_authenticated,
  reject_non_fqdn_hostname,
  reject_non_fqdn_sender,
  reject_non_fqdn_recipient,
  reject_unauth_destination,
  reject_unauth_pipelining,
  reject_invalid_hostname,
  reject_rbl_client bl.spamcop.net,
  reject_rbl_client sbl-xbl.spamhaus.org,
  reject_rbl_client zen.spamhaus.org,
  reject_rbl_client dnsbl.sorbs.net,
  reject_rbl_client rhsbl.sorbs.net,
  reject_rbl_client db.wpbl.info,
  reject_rbl_client cbl.abuseat.org,
  reject_rbl_client proxies.blackholes.wirehub.net,
  reject_rbl_client query.bondedsender.org
smtpd_sasl_auth_enable = yes
smtpd_sasl_authenticated_header = yes
smtpd_sasl_local_domain = $myhostname
smtpd_sasl_security_options = noanonymous
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth

# TLS CONFIG
#smtp_use_tls = yes
#smtpd_use_tls = yes
smtp_tls_security_level=may
smtpd_tls_security_level=may
smtpd_tls_auth_only = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /usr/local/etc/ssl/mail.example.com.key
smtpd_tls_cert_file = /usr/local/etc/ssl/mail.example.com.crt
smtpd_tls_CAfile = /usr/local/etc/ssl/sub.class1.server.ca.pem
smtpd_tls_exclude_ciphers = aNULL, eNULL, EXPORT, DES, RC4, MD5, PSK, aECDH, EDH-DSS-DES-CBC3-SHA, EDH-RSA-DES-CDB3-SHA, KRB5-DES, CBC3-SHA
smtpd_tls_dh1024_param_file = /usr/local/etc/ssl/dhparams.pem
smtpd_tls_loglevel = 0
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_mandatory_protocols=!SSLv2,!SSLv3
tls_random_source = dev:/dev/urandom

#PostgreSQL Configuration
virtual_alias_maps = proxy:pgsql:/usr/local/etc/postfix/pgsql_virtual_alias_maps.cf
virtual_gid_maps = static:125
virtual_mailbox_base = /usr/local/virtual
virtual_mailbox_domains = proxy:pgsql:/usr/local/etc/postfix/pgsql_virtual_domains_maps.cf
virtual_mailbox_limit = 51200000
virtual_mailbox_maps = proxy:pgsql:/usr/local/etc/postfix/pgsql_virtual_mailbox_maps.cf
virtual_minimum_uid = 125
virtual_transport = dovecot
virtual_uid_maps = static:125

# Additional for quota support
virtual_mailbox_limit_maps = proxy:pgsql:/usr/local/etc/postfix/pgsql_virtual_mailbox_limit_maps.cf
proxy_read_maps = $local_recipient_maps $mydestination $virtual_alias_maps
  $virtual_alias_domains $virtual_mailbox_maps $virtual_mailbox_domains
  $relay_recipient_maps $relay_domains $canonical_maps $sender_canonical_maps
  $recipient_canonical_maps $relocated_maps $transport_maps $mynetworks
  $virtual_mailbox_limit_maps
virtual_mailbox_limit_override = yes
virtual_maildir_limit_message = Sorry, this user has overdrawn their diskspace quota. Please try again later.
virtual_overquota_bounce = yes

maximal_queue_lifetime = 4h
bounce_queue_lifetime = 4h

# TRANSPORT MAP
 #
 # See the discussion in the ADDRESS_REWRITING_README document.
```

```
dovecot_destination_recipient_limit = 1
```

Master config

- Edit the master postfix config file:

```
vi /usr/local/etc/postfix/master.cf
```

  - And modify the following:

```
submission inet n       -       n       -       -       smtpd
  -o smtpd_enforce_tls=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject

smtps     inet  n       -       n       -       -       smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject

# Add to end of file
dovecot unix - n n - - pipe
  flags=DRhu user=vscan:vscan argv=/usr/local/libexec/dovecot/deliver -f ${sender} -d ${recipient}
```

Virtual alias map

- Create the postgresql virtual alias map file:

```
vi /usr/local/etc/postfix/pgsql_virtual_alias_maps.cf
```

  - And add the following:

```
user = postfix
password = SuperSecretPostfixPassword
hosts = pg.example.com
dbname = postfix
query = SELECT goto FROM alias WHERE address='%s' AND active = '1'
```

Virtual domain map

- Create the postgresql virtual domain map file:

```
vi /usr/local/etc/postfix/pgsql_virtual_domains_maps.cf
```

  - And add the following:

```
user = postfix
password = SuperSecretPostfixPassword
hosts = pg.example.com
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' and backupmx = '0' and active = '1'
```

Virtual mailbox map

- Create the postgresql virtual mailbox map file:

```
vi /usr/local/etc/postfix/pgsql_virtual_mailbox_maps.cf
```

  - And add the following:

```
user = postfix
password = SuperSecretPostfixPassword
hosts = pg.example.com
dbname = postfix
query = SELECT maildir FROM mailbox WHERE username='%s' AND active = '1'
```

Virtual mailbox limit map

- Create the postgresql virtual mailbox limit map file:

```
vi /usr/local/etc/postfix/pgsql_virtual_mailbox_limit_maps.cf
```

  - And add the following:

```
user = postfix
password = SuperSecretPostfixPassword
hosts = pg.example.com
dbname = postfix
query = SELECT quota FROM mailbox WHERE username='%s'
```

Relay domain map

- Create the postgresql relay domain map file:

```
vi /usr/local/etc/postfix/pgsql_relay_domains_maps.cf
```

  - And add the following:

```
user = postfix
password = SuperSecretPostfixPassword
hosts = pg.example.com
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' and backupmx = '1'
```

Finish configuring Postfix

- Secure Postfix’s PostgreSQL files:

```
chmod 640 /usr/local/etc/postfix/pgsql_*
chgrp postfix /usr/local/etc/postfix/pgsql_*
```

- Create the transport file and update the transport map database:

```
touch /usr/local/etc/postfix/transport
postmap /usr/local/etc/postfix/transport
```

- Edit aliases file, uncomment and change “root” to an email address you want system messages to be mailed to:

```
vi /etc/aliases
```

- And modify the following:

  ```
  root: user@example.com
  ```

- Create the aliases.db file:

```
newaliases
```

Install Nginx

- Install Nginx

```
portmaster www/nginx
```

- Start and enable nginx at boot:

```
echo 'nginx_enable="YES"' >> /etc/rc.conf
service nginx start
```

- Create a configuration directory to make managing individual server blocks easier

```
mkdir /usr/local/etc/nginx/conf.d
```

- Edit the main nginx config file:

```
vi /usr/local/etc/nginx/nginx.conf
```

- And strip down the config file and add the include statement at the end to make it easier to handle various server blocks:

  ```
  #user  nobody;
  worker_processes  1;
  error_log  /var/log/nginx-error.log;

  events {
    worker_connections  1024;
  }

  http {
    include       mime.types;
    default_type  application/octet-stream;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    # Load config files from the /etc/nginx/conf.d directory
  ```

```
  include /usr/local/etc/nginx/conf.d/*.conf;

}
```

Configure PHP

- Install pear-MDB2_Driver_pgsql:

```
portmaster databases/pear-MDB2_Driver_pgsql security/php56-openssl
```

- Configure the default PHP settings

```
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
```

- Edit PHP config file:

```
vi /usr/local/etc/php.ini
```

  - And modify the following:

    ```
    ; UNIX: "/path1:/path2"
    include_path = ".:/usr/local/share/pear"

    post_max_size = 25M

    upload_max_filesize = 25M

    date.timezone = "America/Los_Angeles"

    session.use_only_cookies = 0

    session.save_path = "/tmp"
    ```

- Edit /usr/local/etc/php-fpm.conf:

```
vi /usr/local/etc/php-fpm.conf
```

  - Make the following changes:

    ```
    events.mechanism = kqueue
    listen = /var/run/php-fpm.sock
    listen.owner = www
    listen.group = www
    listen.mode = 0666
    ```

- Start and enable PHP-FPM at boot:

```
echo 'php_fpm_enable="YES"' >> /etc/rc.conf
service php-fpm start
```

- Restart nginx:

```
service nginx restart
```

Configure Postfixadmin

Postfix Admin is a web based interface used to manage mailboxes, virtual domains and aliases. It also features support for vacation/out-of-the-office messages.

- Secure PostfixAdmin files:

```
cd /usr/local/www/postfixadmin
find . -type f -exec chmod 640 {} \;
find . -type d -exec chmod 750 {} \;
```

- Edit the postfixadmin config file:

```
vi /usr/local/www/postfixadmin/config.inc.php
```

  - And modify the following:

```
$CONF['configured'] = true;

$CONF['postfix_admin_url'] = 'https://mail.example.com/postfixadmin/';

$CONF['database_type'] = 'pgsql';
$CONF['database_host'] = 'pg.exmple.com';
$CONF['database_user'] = 'postfix';
$CONF['database_password'] = 'SuperSecretPostfixPassword';
$CONF['database_name'] = 'postfix';

$CONF['admin_email'] = 'postmaster@example.com';

$CONF['default_aliases'] = array (
        'abuse' => 'abuse@example.com',
        'hostmaster' => 'hostmaster@example.com',
        'postmaster' => 'postmaster@example.com',
        'webmaster' => 'webmaster@example.com'
);

$CONF['encrypt'] = 'md5';

$CONF['domain_path'] = 'YES';

$CONF['domain_in_mailbox'] = 'NO';

$CONF['aliases'] = '50';
$CONF['mailboxes'] = '50';
$CONF['maxquota'] = '10240';

$CONF['quota'] = 'YES';

$CONF['quota_multiplier'] = '1048576';

$CONF['vacation'] = 'YES';

$CONF['vacation_domain'] = 'autoreply.example.com';

$CONF['user_footer_link'] = 'http://mail.example.com/';

$CONF['footer_text'] = 'Return to example.com';
$CONF['footer_link'] = 'http://mail.example.com/';
```

```
$CONF['emailcheck_resolve_domain']='NO';

$CONF['mailbox_postdeletion_script']='sudo -u vscan /usr/local/bin/postfixadmin-mailbox-postdeletion.sh';

$CONF['domain_postdeletion_script']='sudo -u vscan /usr/local/bin/postfixadmin-domain-postdeletion.sh';

$CONF['used_quotas'] = 'YES';

$CONF['new_quota_table'] = 'YES';
```

- Install sudo:

```
portmaster security/sudo devel/p5-Tie-Cache
```

- Edit the sudoers file:

```
visudo
```

  - And add the following to the bottom of the file to allow the “www” user to execute the post-deletion scripts as the “vscan” user:

```
www ALL=(vscan)  NOPASSWD: /usr/local/bin/postfixadmin-mailbox-postdeletion.sh \
                 /usr/local/bin/postfixadmin-domain-postdeletion.sh
```

- Create post-deletion directories and copy scripts:

```
mkdir -p /usr/local/virtual/deleted/{mailboxes,domains}
chown -R vscan:vscan /usr/local/virtual/deleted
chmod -R 0700 /usr/local/virtual/deleted
cp /usr/local/www/postfixadmin/ADDITIONS/postfixadmin-*deletion.sh /usr/local/bin
chmod +x /usr/local/bin/postfixadmin*
```

- Edit the postfixadmin-domain-postdeletion.sh file:

```
vi /usr/local/bin/postfixadmin-domain-postdeletion.sh
```

  - And modify the following:

```
# Change this to where you keep your virtual mail users' maildirs.
basedir=/usr/local/virtual

# Change this to where you would like deleted maildirs to reside.
trashbase=/usr/local/virtual/deleted/domains
```

- Edit the postfixadmin-mailbox-postdeletion.sh file:

```
vi /usr/local/bin/postfixadmin-mailbox-postdeletion.sh
```

  - And modify the following:

```
# Change this to where you keep your virtual mail users' maildirs.
```

```
basedir=/usr/local/virtual

# Change this to where you would like deleted maildirs to reside.
trashbase=/usr/local/virtual/deleted/mailboxes
```

- Install needed Perl ports for Vacation to work:

```
portmaster mail/p5-MIME-EncWords mail/p5-Email-Valid mail/p5-Mail-Sender devel/p5-Log-Log4perl
 devel/p5-Log-Dispatch
```

- Create Vacation user and group accounts:

```
pw groupadd vacation
pw useradd vacation -c Virtual\ Vacation -d /nonexistent -g vacation -s /sbin/nologin
```

- Create, populate and secure vacation directory:

```
mkdir /var/spool/vacation
cp /usr/local/www/postfixadmin/VIRTUAL_VACATION/vacation.pl /var/spool/vacation/
chown -R vacation:vacation /var/spool/vacation/
chmod 700 /var/spool/vacation/
chmod 750 /var/spool/vacation/vacation.pl
touch /var/log/vacation.log /var/log/vacation-debug.log
chown vacation:vacation /var/log/vacation*
```

- Edit the vacation.pl script:

```
vi /var/spool/vacation/vacation.pl
```

  - And modify the following:

```
our $db_type = 'Pg';
our $db_host = 'pg.example.com';
our $db_username = 'postfix';
our $db_password = 'SuperSecretPostfixPassword';
our $db_name = 'postfix';
our $vacation_domain = 'autoreply.example.com';
our $logfile = "/var/log/vacation.log";
our $log_level = 0;
our $log_to_file = 1;

my $sender = new Mail::Sender({%smtp_connection,TLS_allowed => 0});
```

- Edit master postfix config file for vacation filter:

```
vi /usr/local/etc/postfix/master.cf
```

  - And add this to the bottom of the file.

```
vacation  unix  -       n       n       -       -       pipe
  flags=DRhu user=vacation argv=/var/spool/vacation/vacation.pl -f ${sender} ${recipient}
```

- Edit the main postfix config file for vacation transport:

```
vi /usr/local/etc/postfix/main.cf
```

- And modify the following:

  ```
  # TRANSPORT MAP
  #
  # See the discussion in the ADDRESS_REWRITING_README document.
  transport_maps = hash:/usr/local/etc/postfix/transport
  vacation_destination_recipient_limit = 1
  ```

- Add proper lines to /usr/local/etc/postfix/transport file:

```
echo "autoreply.example.com vacation:" >> /usr/local/etc/postfix/transport
```

- Create our transport map database for Postfix:

```
postmap /usr/local/etc/postfix/transport
```

- Create a postfixadmin location block in the mail.exmaple.com nginx config file:

```
vi /usr/local/etc/nginx/conf.d/mail.example.com.conf
```

- Add the following:

  ```
  server {
    listen       80;
    listen       443 ssl;
    server_name  mail.example.com;
    root         /usr/local/www;
    access_log   /var/log/mail.example.com-access.log;
    error_log    /var/log/mail.example.com-error.log;

    # SSL Key and Cert
    ssl_certificate /usr/local/etc/ssl/mail.example.com.crt;
    ssl_certificate_key /usr/local/etc/ssl/mail.example.com.key;

    # Configure Strong SSL
    ssl_ciphers 'AES128+EECDH:AES128+EDH:!aNULL';
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_session_cache  builtin:1000  shared:SSL:10m;
    ssl_stapling on;
    ssl_stapling_verify on;
    ssl_prefer_server_ciphers on;
    ssl_dhparam /usr/local/etc/ssl/dhparams.pem;
    add_header Strict-Transport-Security max-age=63072000;
    add_header X-Frame-Options SAMEORIGIN;
    add_header X-Content-Type-Options nosniff;

    location /postfixadmin {
      root   /usr/local/www;
      index  index.php index.html index.htm;
    }

    # For all PHP requests, pass them on to PHP-FPM via FastCGI
    location ~ \.php$ {
      fastcgi_pass unix:/var/run/php-fpm.sock;
      fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
      fastcgi_param PATH_INFO $fastcgi_script_name;
      include fastcgi_params; # include extra FCGI params
  ```

```
    }
}
```

- Change ownership of Postfixadmin web directory:

```
chown -R www:www /usr/local/www/postfixadmin
```

- Restart nginx configuration:

```
service nginx configtest
service nginx restart
```

- Run the dovecot and postfix startup scripts:

```
service dovecot start
service postfix start
```

  - **NOTE**: Check your /var/log/maillog and /var/log/messages to make sure there are no errors.
  - **NOTE**: If you are receiving errors in your logs about $mydestination, be sure that *ANY* 'virtual' domain you are hosting is *NOT* listed in your /etc/hosts file.

- Visit https://mailadmin.example.com/setup.php and generate the password hash at the bottom of the page.
  1. Copy the password hash into your /usr/local/www/postfixadmin/config.inc.php file on the **$setup_password** line.
  2. Next, create a Super Admin Account using the password which created your password hash to submit the information. The username **MUST** be in email address format and the password for the Super Admin account DOES NOT need to be the same password which generated your password hash.

This guide uses superadmin@example.com as the newly created email user.

Setup SpamAssassin

> It uses a robust scoring framework and plug-ins to integrate a wide range of advanced heuristic and statistical analysis tests on email headers and body text including text analysis, Bayesian filtering, DNS blocklists, and collaborative filtering databases.

- Edit the spamassassin config file:

```
vi /usr/local/etc/mail/spamassassin/local.cf
```

  - And add the following to the bottom of the file:

```
ifplugin Mail::SpamAssassin::BayesStore::PgSQL
bayes_sql_dsn DBI:pgsql:maiadb:pg.example.com:5432
bayes_sql_username maiauser
bayes_sql_password SuperSecretPassword
auto_whitelist_factory
endif

ifplugin Mail::SpamAssassin::SQLBasedAddrList
user_awl_dsn DBI:pgsql:maiadb:pg.example.com:5432
user_awl_sql_username maiauser
user_awl_sql_password SuperSecretPassword
bayes_auto_expire 0
endif

# Change the below to reflect your correct internal and external networks.
internal_networks 192.168.0.0/24
trusted_networks 192.168.0.0/24 123.456.789.0/24
```

- Start and enable spamassassin:

```
echo 'spamd_enable="YES"' >> /etc/rc.conf
service sa-spamd start
```

- Configure RAZOR for reporting:

```
su - vscan
razor-admin -discover
razor-admin -create
razor-admin -register -l -user=username@example.com -pass=some_password
exit
```

  - **NOTE**: The above user should be an actual email address you check. The password can be any password you'd like. It's only needed by razor2 to identify and report spam.

Setup FuzzyOCR

The FuzzyOCR Plugin for SpamAssassin improves somewhat upon the standard OCR Plugin, in that it is capable of performing "fuzzy" matching of text strings. This makes it able to handle the innate inaccuracies of OCR engines, spelling mistakes, and deliberate obfuscation of words by spammers, without having to write a lot of explicit regular expression patterns to catch these variations.

- Install Tesseract via ports:

```
portmaster graphics/tesseract devel/p5-IO-All-LWP
```

- Copy FuzzyOcr files to SpamAssassin configuration directory:

```
cp /usr/local/share/examples/FuzzyOcr/FuzzyOcr.* /usr/local/etc/mail/spamassassin
```

Setup ClamAV

- Enable ClamAV at boot time:

```
echo 'clamav_freshclam_enable="YES"' >> /etc/rc.conf
echo 'clamav_clamd_enable="YES"' >> /etc/rc.conf
```

- Create the db, log and socket directories:

```
mkdir -p /var/{log,run,db}/clamav
chown -R vscan:vscan /var/{log,run,db}/clamav
```

- Download the initial ClamAV definitions:

```
freshclam
```

- Start FreshClam as well as the ClamAV daemon:

```
service clamav-freshclam start
service clamav-clamd start
```

Setup Maia Mailguard

Maia Mailguard is a web-based interface and management system based on the popular amavisd-new e-mail scanner and SpamAssassin. Written in Perl and PHP, Maia Mailguard gives end-users control over how their mail is processed by virus scanners and spam filters, while giving mail administrators the power to configure site-wide defaults and limits.

- Populate the database:

```
cd /usr/local/share/doc/maia
psql -h pg.example.com -U maiauser -W maiadb < maia-pgsql.sql
```

  - **NOTE**: I had a problem with the autolearn_status column causing problems when null values were inserted, and thus not delivering external email to the mail server. To work around this I reset the autolearn_status column default values:

```
psql -h pg.example.com -U maiauser -d maiadb
ALTER TABLE ONLY maia_mail ALTER COLUMN autolearn_status SET DEFAULT 'unavailable';
```

- Edit the maia config file:

```
vi /usr/local/etc/maia/maia.conf
```

  - And modify the following:

```
# Configure your Maia database DSN here
$dsn = 'DBI:Pg:dbname=maiadb;host=pg.example.com;port=5432';

# Your Maia database user's login name
$username = 'maiauser';

# Your Maia database user's password
$password = 'SuperSecretPassword';

# Address rewriting type [0..5] (see config.php)
$address_rewriting_type = 4;

# Authentication method (see config.php)
$auth_method = 'sql';

# Base URL to Maia's PHP scripts
$base_url = "https://mail.example.com/maia/";
```

- Run configtest.pl executable:

```
perl /usr/local/share/maia/scripts/configtest.pl
```

- Load SpamAssassin rules:

```
sa-update
su - vscan
perl /usr/local/share/maia/scripts/load-sa-rules.pl --debug
exit
```

- Edit maia config file:

```
vi /usr/local/www/maia/config.php
```

  - And modify the following

    ```
    date_default_timezone_set("America/Los_Angeles");

    $maia_sql_dsn = "pgsql://maiauser:SuperSecretPassword@tcp(pg.example.com:5432)/maiadb";

    $purifier_cache = '/usr/local/www/maia/web';

    $address_rewriting_type = 4;

    $auth_method = "pop3";
    ```

- Edit the smarty php file:

```
vi /usr/local/www/maia/smarty.php
```

  - And delete the leading slash from the “/themes” bit on line 102. So, it should look like this:

    ```
    $this->assign('template_dir', 'themes/'.$theme.'/');
    ```

- Create a maia location block in the mail.example.com nginx config file:

```
vi /usr/local/etc/nginx/conf.d/mail.example.com.conf
```

  - Add the following:

    ```
    server {
      listen       80;
      listen       443 ssl;
      server_name  mail.example.com;
      root         /usr/local/www;
      access_log   /var/log/maia.example.com-access.log;
      error_log    /var/log/maia.example.com-error.log;

      # SSL Key and Cert
      ssl_certificate /usr/local/etc/ssl/mail.example.com.crt;
      ssl_certificate_key /usr/local/etc/ssl/mail.example.com.key;

      # Configure Strong SSL
      ssl_ciphers 'AES128+EECDH:AES128+EDH:!aNULL';
      ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
      ssl_session_cache  builtin:1000  shared:SSL:10m;
      ssl_stapling on;
      ssl_stapling_verify on;
      ssl_prefer_server_ciphers on;
      ssl_dhparam /usr/local/etc/ssl/dhparams.pem;
      add_header Strict-Transport-Security max-age=63072000;
      add_header X-Frame-Options SAMEORIGIN;
      add_header X-Content-Type-Options nosniff;

      location /postfixadmin {
        root   /usr/local/www;
        index  index.php index.html index.htm;
      }
    ```

```
  location /maia {
    root   /usr/local/www;
    index  index.php index.html index.htm;
  }

  # For all PHP requests, pass them on to PHP-FPM via FastCGI
  location ~ \.php$ {
    fastcgi_pass unix:/var/run/php-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_script_name;
    include fastcgi_params; # include extra FCGI params
  }
}
```

- Restart nginx and php-fpm:

```
service nginx restart
service php-fpm restart
```

- Visit https://mail.example.com/maia/admin/configtest.php and verify everything is working.
- Create maia run directory:

```
mkdir /var/run/maia
chown vscan:vscan /var/run/maia
```

- Create maia log directory:

```
mkdir /var/log/maia
chown vscan:vscan /var/log/maia
```

- Edit maia daemon config file:

```
vi /usr/local/etc/maia/maiad.conf
```

  - And modify the following:

```
$lock_file = "/var/run/maia/maiad.lock";
$pid_file = "/var/run/maia/maiad.pid";

$mydomain = 'example.com';

$myhostname = 'mail.example.com';

@lookup_sql_dsn = ( ['DBI:Pg:dbname=maiadb;host=pg.example.com;port=5432', 'maiauser', 'SuperSecretPassword'] );

$unrar = ['rar', 'unrar'];

### http://www.clamav.net/
['ClamAV-clamd',
   \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.sock"],
   qr/\bOK$/, qr/\bFOUND$/,
   qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

- Set Maia-Mailguard to start at boot and start it now:

```
echo 'maiad_enable="YES"' >> /etc/rc.conf
service maiad start
```

- Visit https://mail.example.com/maia/
  - You should be greeted with a login screen. If so, great! Let's log in and acquire admin privileges…

- Instead of https://mail.example.com/maia/login.php (The default), visit https://maia.example.com/maia/login.php?super=register
  - And log in with any currently existing virtual user. Be sure to use a full email address to log into Maia-Mailguard. This guide uses superadmin@example.com. That user will now have admin privs via Maia (So, be careful which user you choose).

- Once logged into Maia-Mailguard as an administrator, click the **Admin** link at the top of the page (Key-shaped icon). Next, click **System Configuration**. Now adjust the **Mail size limit** to the limit set in the postfix main.cf file, in this case **25600000**.

IMPORTANT: For each domain you create using Postfixadmin or any other way you may create it, Maia needs to know about it in order to create users. This might seem like a redundant issue, but it really makes a difference and here's why. When Maia recieves mail for a user that doesn't exist, it uses the default domain's (@.) settings. This is fine. However, if it considers that mail to be spam when it is not, the user cannot retrieve that message later being as the default settings don't house mail for a non-existant user.

So, be sure to add any domain you add via PostfixAdmin to Maia-Mailguard as well.

- Edit main postfix file:

```
vi /usr/local/etc/postfix/main.cf
```

  - And modify the following:

```
# Maia-Mailguard
#
content_filter=smtp-amavis:[127.0.0.1]:10024
```

- Edit the master postfix file:

```
vi /usr/local/etc/postfix/master.cf
```

  - And modify the following:

```
smtp-amavis unix - - n - 2 smtp
  -o smtp_data_done_timeout=2400
  -o smtp_send_xforward_command=yes
  -o disable_dns_lookups=yes
  -o max_use=20
  -o smtp_tls_security_level=none
127.0.0.1:10025 inet n - n - - smtpd
  -o content_filter=
  -o local_recipient_maps=
  -o relay_recipient_maps=
  -o smtpd_restriction_classes=
  -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 mynetworks_style=host
  -o mynetworks=127.0.0.0/8
  -o strict_rfc821_envelopes=yes
  -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_address
```

```
_mappings
  -o smtp_tls_security_level=none
```

- Reload Postfix:

```
postfix reload
```

- Edit the “vscan” user’s cronjobs:

```
crontab -u vscan -e
```

  - And add the following to the vscan users’s crontab.

```
#Load new rules and store into Maia database.
30 4 * * * /usr/local/share/maia/scripts/load-sa-rules.pl > /dev/null

#Train Spam Assassin.
0 * * * * /usr/local/share/maia/scripts/process-quarantine.pl --learn --report > /dev/null

#Take a snapshot of the stats at the start of every hour.
0 * * * * /usr/local/share/maia/scripts/stats-snapshot.pl > /dev/null

#Purge mail that has not been confirmed.
0 23 * * * /usr/local/share/maia/scripts/expire-quarantine-cache.pl > /dev/null

#Send quarantine reminders.
0 15 * * * /usr/local/share/maia/scripts/send-quarantine-reminders.pl > /dev/null

#Send quarantine digests.
0 15 * * * /usr/local/share/maia/scripts/send-quarantine-digests.pl > /dev/null

#Force bayesian auto-expiry during off-peak hours.
25 2 * * * /usr/local/bin/sa-learn --sync --force-expire > /dev/null
```

- Remove BATCH setting from make.conf file:

```
sed -i.orig -e '/^BATCH=yes/d' /etc/make.conf
```

Install Roundcube

Roundcube is a browser-based multilingual IMAP client with an application-like user interface. It provides full functionality you expect from an email client, including MIME support, address book, folder manipulation, message searching and spell checking.

- Install Roundcube via ports:

```
portmaster mail/roundcube
```

  - **NOTE**: Make sure **[X]PSPELL**, **[X]SSL**, and **[X]PGSQL** are selected from the menu.

- Install PHP FileInfo and Exif:

```
portmaster sysutils/php56-fileinfo graphics/php56-exif
```

- Populate the Roundcube database:

```
cd /usr/local/www/roundcube/SQL
psql -h pg.example.com -U roundcubeuser -W roundcubedb < postgres.initial.sql
```

- Copy Roundcube configuration files and set permissions:

```
cd /usr/local/www/roundcube/config
cp config.inc.php.sample config.inc.php
cd /usr/local/www/roundcube/plugins/managesieve
cp config.inc.php.dist config.inc.php
cd /usr/local/www/roundcube/plugins/password
cp config.inc.php.dist config.inc.php
cd /usr/local/www/roundcube
find . -type f -name "config.inc.php" -exec chmod 0600 {} \; -exec chown www {} \;
```

- Edit the roundcube config file:

```
vi /usr/local/www/roundcube/config/config.inc.php
```

  - And modify the following:

```
$config['db_dsnw'] = 'pgsql://roundcubeuser:SuperSecretRoundcubePassword@pg.example.com/roundcubedb';

$config['imap_auth_type'] = LOGIN;

$config['smtp_server'] = 'tls://localhost';

$config['smtp_port'] = 587;

$config['smtp_user'] = '%u';

$config['smtp_pass'] = '%p';

$config['support_url'] = 'user@example.com';

 $config['plugins'] = array(
     'archive',
     'zipdownload',
     'managesieve',
     'password',
 );

# Add to end of the file
$config['spellcheck_engine'] = 'pspell';
$config['preview_pane'] = true;
$config['mime_types'] = '/usr/local/etc/nginx/mime.types';
$config['enable_installer'] = false;
```

- Edit the roundcube managesieve plugin config file:

```
vi /usr/local/www/roundcube/plugins/managesieve/config.inc.php
```

  - And modify the following:

```
$config['managesieve_default'] = '/usr/local/virtual/home/default.sieve';
```

- Edit roundcube password plugin config file:

```
vi /usr/local/www/roundcube/plugins/password/config.inc.php
```

  - And modify the following:

    ```
    $config['password_minimum_length'] = 8;

    $config['password_db_dsn'] = 'pgsql://postfix:SuperSecretPostfixPassword@pg.example.com/postfix';

    $config["password_query"] = "UPDATE mailbox SET password=%c WHERE username=%u";
    ```

- Secure Roundcube configuration files:

```
chmod 600 /usr/local/www/roundcube/config/*
chown www /usr/local/www/roundcube/config/*
chown -R www:www /usr/local/www/roundcube/
```

- Create a roundcube location block in the mail.example.com nginx config file:

```
vi /usr/local/etc/nginx/conf.d/mail.example.com.conf
```

  - Add the following:

    ```
    server {
      listen       80;
      listen       443 ssl;
      server_name  mail.example.com;
      root         /usr/local/www;
      access_log   /var/log/mail.example.com-access.log;
      error_log    /var/log/mail.example.com-error.log;

      # SSL Key and Cert
      ssl_certificate /usr/local/etc/ssl/mail.example.com.crt;
      ssl_certificate_key /usr/local/etc/ssl/mail.example.com.key;

      # Configure Strong SSL
      ssl_ciphers 'AES128+EECDH:AES128+EDH:!aNULL';
      ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
      ssl_session_cache  builtin:1000  shared:SSL:10m;
      ssl_stapling on;
      ssl_stapling_verify on;
      ssl_prefer_server_ciphers on;
      ssl_dhparam /usr/local/etc/ssl/dhparams.pem;
      add_header Strict-Transport-Security max-age=63072000;
      add_header X-Frame-Options SAMEORIGIN;
      add_header X-Content-Type-Options nosniff;

      location /postfixadmin {
        root   /usr/local/www;
        index  index.php index.html index.htm;
      }

      location /maia {
        root   /usr/local/www;
        index  index.php index.html index.htm;
      }

      location /roundcube {
    ```

```
    root   /usr/local/www;
    index  index.php index.html index.htm;
  }

  # For all PHP requests, pass them on to PHP-FPM via FastCGI
  location ~ \.php$ {
    fastcgi_pass unix:/var/run/php-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_script_name;
    include fastcgi_params; # include extra FCGI params
  }

}
```

- Restart nginx and php-fpm:

```
service nginx restart
service php-fpm restart
```

- Visit https://mail.example.com/roundcube/
  - Login to roundcube using your full email address and password. You should now be able to use Roundcube as a webmail client.
  - **NOTE**: If you're having any problems, be sure to check your Roundcube logs located in /usr/local/www/roundcube/logs.

Install Fetchmail

Fetchmail is a full-featured, robust, well-documented remote-mail retrieval and forwarding utility intended to be used over on-demand TCP/IP links.

- Install fetchmail:

```
portmaster mail/fetchmail
```

- Change the fetchmail config ownership:

```
chown vscan /usr/local/etc/fetchmailrc
```

- And change the fetchmail run directory to the vscan group:

```
chmod g+w /var/run/fetchmail
chgrp vscan /var/run/fetchmail
```

- Edit the global fetchmailrc config:

```
vi /usr/local/etc/fetchmailrc
```

  - And add the following:

```
set postmaster "postmaster@example.com"
set no bouncemail
poll outsidemail.example.net with protocol pop3
  user "user@example.com" pass "SuperSecretEmailPassword" nofetchall keep no rewrite mda "
/usr/local/libexec/dovecot/deliver -d user@example.com"
```

- Test fetchmail:

```
sudo -u vscan fetchmail -f /usr/local/etc/fetchmailrc
```

- Fix the fetchmail init script:

```
vi /usr/local/etc/rc.d/fetchmail:
```

    - And add the -i /var/run/fetchmail/.fetchids line to the **fetchmail_flags** definition:

```
fetchmail_flags="-f ${fetchmail_config} \
                --pidfile ${pidfile} \
                -d ${fetchmail_polling_interval} \
                -i /var/run/fetchmail/.fechids \
                ${fetchmail_logging_facility}"
```

- Start and enable fetchmail at boot:

```
echo 'fetchmail_user="vscan"' >> /etc/rc.conf
echo 'fetchmail_polling_interval="180"' >> /etc/rc.conf
echo 'fetchmail_enable="YES"' >> /etc/rc.conf
service fetchmail start
```

Install Postgrey

Greylisting is a new method of blocking significant amounts of spam at the mailserver level, but without resorting to heavyweight statistical analysis or other heuristical (and error-prone) approaches.

- Install postgrey to add greylisting to postfix:

```
portmaster mail/postgrey
```

- Start and enable postgrey at boot:

```
echo 'postgrey_enable="YES"' >> /etc/rc.conf
service postgrey start
```

- Edit the main postfix config file:

```
vi /usr/local/etc/postfix/main.cf
```

    - And modify the smtpd_recipient_restrictions parameter to add the check_policy_service option:

```
smtpd_recipient_restrictions =
  permit_mynetworks,
  permit_sasl_authenticated,
  reject_non_fqdn_hostname,
  reject_non_fqdn_sender,
  reject_non_fqdn_recipient,
  reject_unauth_destination,
  reject_unauth_pipelining,
    reject_invalid_hostname,
    reject_rbl_client bl.spamcop.net,
    reject_rbl_client sbl-xbl.spamhaus.org,
    reject_rbl_client zen.spamhaus.org,
    reject_rbl_client dnsbl.sorbs.net,
    reject_rbl_client rhsbl.sorbs.net,
    reject_rbl_client db.wpbl.info,
    reject_rbl_client cbl.abuseat.org,
    reject_rbl_client proxies.blackholes.wirehub.net,
    reject_rbl_client query.bondedsender.org,
    check_policy_service inet:127.0.0.1:10023
```

- Check the postfix config before reloading the service:

```
postfix check
```

- Finally reload the postfix service:

```
service postfix reload
```

Install DKIM

DomainKeys Identified Mail lets an organization take responsibility for a message that is in transit. The organization is a handler of the message, either as its originator or as an intermediary. Their reputation is the basis for evaluating whether to trust the message for further handling, such as delivery. Technically DKIM provides a method for validating a domain name identity that is associated with a message through cryptographic authentication.

- Install OpenDKIM:

```
portmaster mail/opendkim
```

- Then allow OpenDKIM starting at boot time and executing as opendkim user:

```
echo 'milteropendkim_enable="YES"' >> /etc/rc.conf
echo 'milteropendkim_uid="opendkim"' >> /etc/rc.conf
```

- Adding the opendkim user:

```
pw useradd -n opendkim -d /var/db/opendkim -g mail -m -s "/usr/sbin/nologin" -w no
```

- Edit Postfix configuration file:

```
vi /usr/local/etc/postfix/main.cf
```

  - And instruct postfix to use dkim milter:

```
smtpd_milters = inet:127.0.0.1:8891
non_smtpd_milters = $smtpd_milters
milter_default_action = accept
```

- Use a sample configuration file for OpenDKIM:

```
cp /usr/local/share/doc/opendkim/opendkim.conf.simple /usr/local/etc/mail/opendkim.conf
```

- Edit the configuration file:

```
vi /usr/local/etc/mail/opendkim.conf
```

  - Feel free to use the following one slightly edited to work with example.com domain:

```
LogWhy yes
Syslog yes
SyslogSuccess yes
Canonicalization relaxed/simple
Domain example.com
Selector example.com
KeyFile /var/db/opendkim/example.com.private
Socket inet:8891@localhost
ReportAddress portmaster@altservice.com
SendReports yes
```

- Now generate the keys, one will be used by opendkim to sign your messages and the other to be inserted in your DNS zone:

```
opendkim-genkey -D /var/db/opendkim -d example.com -s example.com
```

- Now insert example.com.txt content in example.com DNS zone.

```
cat /var/db/opendkim/example.com.txt
```

  - *Example output*:

```
example.com._domainkey IN TXT "v=DKIM1; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDOc
RbLGARxEFI9Ibwx79tk1kMi36rFeAT4aLu4iI3ctPUWa7y0WcuMZGCBQMMutolT8IM9e55AToqtr/W/rbKlhoeiA0r
8qJZiIX/NkjkLIXzR+9h1i47dD5zCu4u436YN0y4DgZU9bZ3D4hvoC9hSHCcCwzosSRwBpaxIMZuRGQIDAQAB" ; -
---- DKIM example.com for example.com
```

  - **NOTE**: The DNS TXT record must be put on the global DNS server providing your domain name to the Internet.

- Also add another TXT Record to your zone file:

```
_adsp._domainkey.mydomain.com IN TXT “dkim=unknown”
```

- Start the opendkim service:

```
service milter-opendkim start
```

- Restart postfix:

```
service postfix restart
```

Install DCC

Distributed Checksum Clearinghouse is an anti-spam content filter that runs on a variety of operating systems. The idea of the DCC is that if mail recipients could compare the mail they receive, they could recognize unsolicited bulk mail. A DCC server totals reports of "fuzzy" checksums of messages from clients and answers queries about the total counts for checksums of mail messages.

- Reinstall spamassassin with DCC support:

```
cd /usr/ports/mail/spamassassin
make config
```

  - Enable **[X]DCC**
  - Reinstall spamassassin:

    ```
    portmaster
    ```

- Restart spamassassin:

```
service sa-spamd restart
```

Install SPF

Briefly, the design intent of the SPF resource record (RR) is to allow a receiving MTA (Message Transfer Agent) to interrogate the Name Server (DNS) of the domain which appears in the email (the sender) and determine if the originating IP of the mail (the source) is authorized to send mail for the sender's domain.

- Install the postfix SPF policyd:

```
portmaster mail/postfix-policyd-spf-python
```

- Edit the main postfix config file:

```
vi /usr/local/etc/postfix/main.cf
```

  - And modify the smtpd_recipient_restrictions parameter to add the check_policy_service option:

    ```
    policy-spf_time_limit = 3600s

    smtpd_recipient_restrictions =
      permit_mynetworks,
      permit_sasl_authenticated,
      reject_non_fqdn_hostname,
      reject_non_fqdn_sender,
      reject_non_fqdn_recipient,
      reject_unauth_destination,
      reject_unauth_pipelining,
      reject_invalid_hostname,
      reject_rbl_client bl.spamcop.net,
      reject_rbl_client sbl-xbl.spamhaus.org,
      reject_rbl_client zen.spamhaus.org,
      reject_rbl_client dnsbl.sorbs.net,
      reject_rbl_client rhsbl.sorbs.net,
      reject_rbl_client db.wpbl.info,
      reject_rbl_client cbl.abuseat.org,
      reject_rbl_client proxies.blackholes.wirehub.net,
      reject_rbl_client query.bondedsender.org,
    ```

```
  check_policy_service inet:127.0.0.1:10023,
  check_policy_service unix:private/policy-spf
```

- Edit the master postfix config file:

```
vi /usr/local/etc/postfix/master.cf
```

  - And add the following to the bottom of the file:

```
policy-spf  unix  -       n       n       -       -       spawn
     user=nobody argv=/usr/local/bin/policyd-spf
```

- Reload postfix:

```
service postfix reload
```

Install Fail2ban

- Install py-fail2ban:

```
portmaster security/py-fail2ban
```

- Edit the ipfw action file:

```
vi /usr/local/etc/fail2ban/action.d/ipfw.conf
```

  - And modify the localhost parameter to the IP address of the server:

```
localhost = 192.168.1.100
```

- Create the local SSH file

```
vi /usr/local/etc/fail2ban/jail.d/ssh.conf
```

  - And add the following

```
[ssh-ipfw]
enabled  = true
filter   = sshd
logpath  = /var/log/auth.log
action   = ipfw
findtime  = 600
maxretry = 3
bantime  = 3600
```

- Create the dovecot service definition:

```
vi /usr/local/etc/fail2ban/jail.d/dovecot-auth.conf
```

  - And add the following:

```
[dovecot]
enabled = true
filter  = dovecot
port    = pop3,pop3s,imap,imaps
logpath = /var/log/maillog
action   = ipfw
findtime  = 600
maxretry = 3
bantime  = 3600
```

- Create the postfix service definition:

```
vi /usr/local/etc/fail2ban/jail.d/postfix-auth.conf
```

  - And add the following:

```
[postfix]
enabled = true
filter  = postfix
port    = smtp,ssmtp
logpath = /var/log/maillog
action   = ipfw
findtime  = 600
maxretry = 3
bantime  = 3600
```

- Create the postfix sasl service definition:

```
vi /usr/local/etc/fail2ban/jail.d/postfix-sasl.conf
```

  - And add the following:

```
[postfix]
enabled = true
filter  = postfix-sasl
port    = smtp,ssmtp
logpath = /var/log/maillog
action   = ipfw
findtime  = 600
maxretry = 3
bantime  = 3600
```

- Start and enable fail2ban at boot:

```
echo 'fail2ban_enable="YES"' >> /etc/rc.conf
service fail2ban start
```

- To list current banned IP:

```
ipfw list
```

Resources

- http://www.purplehat.org/?page_id=4
- http://www.maiamailguard.com/maia/wiki/Install

- https://forums.freebsd.org/threads/postgresql-postfix-nginx-php-roundcube-dovecot-spamassassin-clamav-spamd.10728/
- https://wiki.gentoo.org/wiki/Complete_Virtual_Mail_Server/Postfix_to_Database
- http://www.iredmail.org/docs/store.spamassassin.bayes.in.sql.html
- https://spamassassin.apache.org/full/3.0.x/dist/doc/Mail_SpamAssassin_Conf.html
- https://github.com/technion/maia_mailguard/issues/20
- https://www.freebsd.org/cgi/man.cgi?query=fetchmail&manpath=SuSE+Linux/i386+11.3
- http://www.development-cycle.com/2008/08/freebsd-postfix-greylisting/
- http://www.zytrax.com/tech/survival/postfix.html
- https://help.ubuntu.com/community/Postfix/SPF
- https://sites.google.com/site/ghidit/how-to-2/freebsd-9-mail-server-setup-postfix-dovecot-2-virtual-users-mysql-sasl-postfixadmin-and-others
- http://bsdwiki.com/postfixdovecotsql
- http://www.prado.it/2012/04/26/how-to-run-postfix-with-opendkim-on-freebsd-9-0/
- https://weakdh.org/sysadmin.html
- http://wiki2.dovecot.org/SSL/DovecotConfiguration

History

#1 - 09/01/2015 08:45 PM - Daniel Curtis

- Description updated

- Status changed from New to In Progress

- % Done changed from 0 to 10

#2 - 09/02/2015 09:44 PM - Daniel Curtis

- Description updated

- Priority changed from Normal to High

- % Done changed from 10 to 20

#3 - 09/03/2015 12:36 PM - Daniel Curtis

- Description updated

- % Done changed from 20 to 30

#4 - 09/03/2015 02:27 PM - Daniel Curtis

- Description updated

#5 - 09/03/2015 04:58 PM - Daniel Curtis

- Description updated

- % Done changed from 30 to 50

#6 - 09/03/2015 05:09 PM - Daniel Curtis

- Description updated

#7 - 09/04/2015 10:41 AM - Daniel Curtis

- Description updated

#8 - 09/04/2015 11:52 AM - Daniel Curtis

- Description updated

#9 - 09/04/2015 05:49 PM - Daniel Curtis

- Description updated

- % Done changed from 50 to 60

#10 - 09/05/2015 09:57 PM - Daniel Curtis

- Description updated

- % Done changed from 60 to 70

#11 - 09/06/2015 12:17 PM - Daniel Curtis

- Description updated

#12 - 09/06/2015 01:57 PM - Daniel Curtis

- Description updated

#13 - 09/06/2015 05:31 PM - Daniel Curtis

- Description updated

- % Done changed from 70 to 80

#14 - 09/06/2015 07:58 PM - Daniel Curtis

- Description updated

- % Done changed from 80 to 90

#15 - 09/07/2015 07:41 PM - Daniel Curtis

- Description updated

#16 - 09/07/2015 08:50 PM - Daniel Curtis

- Description updated

#17 - 09/08/2015 09:49 AM - Daniel Curtis

- Description updated

#18 - 09/08/2015 11:11 AM - Daniel Curtis

- Description updated

#19 - 09/08/2015 01:29 PM - Daniel Curtis

- Description updated

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

#20 - 09/08/2015 08:16 PM - Daniel Curtis

- Description updated

#21 - 09/09/2015 12:06 PM - Daniel Curtis

- Description updated

#22 - 09/10/2015 02:42 PM - Daniel Curtis

- Description updated

#23 - 09/14/2015 12:22 PM - Daniel Curtis

- Description updated

#24 - 09/15/2015 03:03 PM - Daniel Curtis

- Description updated

#25 - 09/15/2015 04:43 PM - Daniel Curtis

- Description updated

#26 - 09/15/2015 06:29 PM - Daniel Curtis

- Description updated

#27 - 09/16/2015 09:26 AM - Daniel Curtis

- Description updated

#28 - 09/16/2015 12:47 PM - Daniel Curtis

- Description updated

#29 - 09/16/2015 02:27 PM - Daniel Curtis

- Description updated

#30 - 09/16/2015 08:45 PM - Daniel Curtis

- Description updated

#31 - 09/17/2015 04:37 PM - Daniel Curtis

- Description updated

#32 - 09/17/2015 05:04 PM - Daniel Curtis

- Description updated

#33 - 09/18/2015 03:39 PM - Daniel Curtis

- Description updated

#34 - 09/21/2015 02:11 PM - Daniel Curtis

- Description updated

#35 - 09/22/2015 10:36 AM - Daniel Curtis

- Description updated

#36 - 09/22/2015 05:16 PM - Daniel Curtis

- Description updated

#37 - 09/22/2015 05:42 PM - Daniel Curtis

- Description updated

#38 - 09/22/2015 06:03 PM - Daniel Curtis

- Description updated

#39 - 09/24/2015 12:23 PM - Daniel Curtis

- Description updated

#40 - 09/24/2015 04:11 PM - Daniel Curtis

- Description updated

#41 - 09/25/2015 04:38 PM - Daniel Curtis

- Description updated

#42 - 09/28/2015 02:45 PM - Daniel Curtis

- Description updated

#43 - 09/29/2015 08:16 PM - Daniel Curtis

- Description updated

#44 - 09/30/2015 10:10 AM - Daniel Curtis

- Description updated

#45 - 10/01/2015 10:50 AM - Daniel Curtis

- Description updated

#46 - 10/18/2015 06:01 PM - Daniel Curtis

- Description updated

#47 - 10/22/2015 04:02 PM - Daniel Curtis

- Description updated

#48 - 11/27/2015 04:47 PM - Daniel Curtis

- Status changed from Resolved to Closed