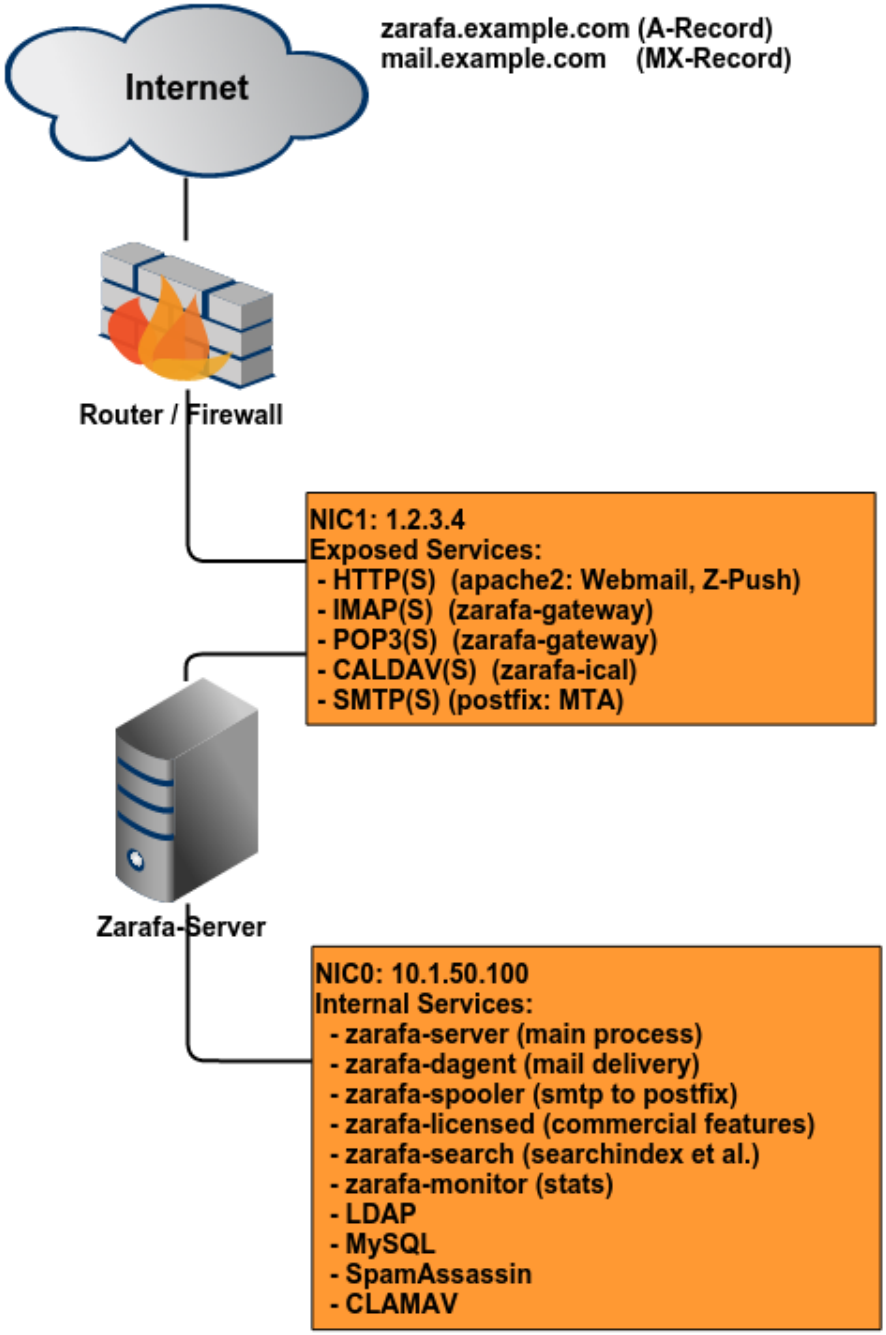
Serversetup multi-tenant Zarafa+Postfix+SASL+SpamAssassin+Clamav+openLDAP (Ubuntu 14.04 LTS)

This guide is deprecated as of 2020-01-01 in favor of a new one:

Serversetup multi-tenant Kopano+Postfix+SASL+rspamd+openLDAP (Debian 10 buster)

See further below for the old guide. Thank you for your attention.

Zarafa Groupware and dependent infrastructure setup guidelines.


Internet
zarafa.example.com (A-Record)
mail.example.com (MX-Record)
Router / Firewall
NIC1: 1.2.3.4
Exposed Services:
- HTTP(S) (apache2: Webmail, Z-Push)
- IMAP(S) (zarafa-gateway)
- POP3(S) (zarafa-gateway)
- CALDAV(S) (zarafa-ical)
- SMTP(S) (postfix: MTA)
Zarafa-Server
NIC0: 10.1.50.100
Internal Services:
- zarafa-server (main process)
- zarafa-dagent (mail delivery)
- zarafa-spooler (smtp to postfix)
- zarafa-licensed (commercial features)
- zarafa-search (searchindex et al.)
- zarafa-monitor (stats)
- LDAP
- MySQL
- SpamAssassin
- CLAMAV

Preface

This is the new and reworked setup guide for a complete virtual-user, multi-tenant, LDAP-backed Zarafa Setup based on Zarafa 7.2 and Ubuntu LTS 14.04. Dspam has been dropped in favor of spamassassin and postfix has been augmented to use DNS-based blacklisting based on multiple online blacklist providers. LDAP configuration has been expanded with the suggested optimizations taken from the Zarafa user wiki.

For LDAP Management I tried various new tools (z-admin, ldap-account-manager, admin4, etc.) but none of them could completely replace phpldapadmin with the zarafa templates I reworked for the latest phpldapadmin version, so decided to stick with it. Although ldap-account-manager supports zarafa accounts you need the paid 'Pro' version, and I'm not sure if it fully supports multi-tenant setups (correct me if I'm wrong). z-admin on the other hand looks like a nice fire-and-forget solution, but does too much behind-the-scenes configuration for my taste. I like my systems lean and clean 😎

If you want to stick with Dspam you can still use the instructions from my old guide, they still apply.

Links:

- http://www.zarafa.com/content/z-admin-web-based-administration-interface-zcp
- https://www.ldap-account-manager.org/lamcms/
- http://www.admin4.org/

Updated templates for phpldapadmin:

File	Modified
XML File create-zarafaAccount.xml	Dec 26, 2014 by David Gabriel
XML File create-zarafaCompany.xml	Dec 26, 2014 by David Gabriel
XML File modify-zarafaAccount.xml	Dec 26, 2014 by David Gabriel
XML File modify-zarafaCompany.xml	Dec 26, 2014 by David Gabriel

Download All

Throughout the guide only lines in config files that need to be changed are shown.
This guide assumes:

- general linux knowledge: basic commands and system administration know-how
- basic mysql knowledge
- basic ldap knowledge
- basic apache knowledge
- intermediate postfix / clamav / spamassassin knowledge

The Server will be directly exposed to the internet on the ports shown in the diagram above. Therefore another interface is assigned to the server that will handle internal connections that need not be seen from the internet and are only used for local or server internal communication.

- This is NOT a copy-and-paste how-to!
- The domain 'example.com' is used, replace all instances with your own domain (in ldap too, like: dc=example,dc=com).
- The external interface is assumed to have the IP 1.2.3.4. Replace with your own external interface address.
- The internal interface is assumed to have the IP 10.1.50.100. Replace with your own internal interface address.
- The password 'secret' will be used for all passwords. Replace with DIFFERENT, STRONG passwords accordingly!
- This setup assumes a local MySQL database. If you use a remote, dedicated database you will need to change the database settings and grants in various config files or the database itself.

Server

- Ubuntu 14.04 LTS 64bit http://www.ubuntu.com/download/server
- Zarafa 7.2 from http://download.zarafa.com/community
- MySQL 5.6, openLDAP 2.3, postfix, clamav, spamassassin, phpldapadmin from Ubuntu Repository
- z-push from http://z-push.org/

Certificates

For this setup two different certificates are needed. One for the Webserver (webmail.example.com) and one for the mail server (mail.example.com). It is advised to use official certificates, however for testing purposes you can create your own ones with openssl.

> The web certificates will be called 'web.crt' and 'web.key'. Replace with your own certificate filenames.
> The mail certificates will be called 'mail.crt' and 'mail.key'. Replace with your own certificate filenames.

Install Packages

> Ubuntu is very noisy and want's to configure postfix and ldap during the installation of the package. Refuse to do so or just let it create an empty configuration. You will set up these programs manually later.

Grab packages from the zarafa-site, install per their instructions. Also install postfix, spamassassin, clamav, slapd, phpldapadmin and mysql. The zarafa packages should have already pulled in apache2 and (mod)php5 as a dependency. If this is not the case please correct these dependencies manually.

```
~# aptitude install postfix postfix-ldap postfix-pcre mysql-server-5.6 slapd ldap-utils php5-ldap sasl2-bin
php5-cli php-soap phpldapadmin spamassassin clamav clamav-freshclam clamsmtp sa-compile spamc razor pyzor
```

MySQL Setup

Not much to do here, just create databases, users and permissions for zarafa:

```
~# mysql -uroot -p
mysql> create database zarafa;
mysql> grant all on zarafa.* to 'zarafa'@'localhost' identified by 'secret';
mysql> flush privileges;
```

LDAP Setup

Probably the part that generates the most head-scratching. Although the latest Ubuntu LTS server already uses the 'cn=config' format, this guide sticks with the old slapd.conf format.

> Author's personal note:
>
> Seriously, whoever came up with this online space-magic config change configuration format should have given the whole thing a good rethinking. Not only is it hard to understand and even harder to 'read', it also can break a running system at the slightest mistake. Therefore this guide sticks with the old **slapd.conf** configuration format. If you prefer the new **cn=config** format this guide is usable in most points, however you have to convert all **slapd.conf** statements to **cn=config** statement with i.e. **slaptest** and add them to **/etc/ldap/slapd.d/** directory manually.

LDAP Server Setup

First adjust the default-file:

/etc/default/slapd

```
SLAPD_CONF=/etc/ldap/slapd.conf  # this triggers the old slapd.conf behaviour
SLAPD_SERVICES="ldap://10.1.50.100:389/ ldap://127.0.0.1:389/ ldapi:///"
SLAPD_OPTIONS="-4"
```

Client config file:

/etc/ldap/ldap.conf

```
ldap_version 3
URI ldap://10.1.50.100
SIZELIMIT 0
TIMELIMIT 0
DEREF never
BASE dc=example, dc=com
```

Server config file. I included the 'nis' schema here too, some might need it for nis domain setups. Copy the zarafa-schema to **'/etc/ldap/schema'** and generate a ldap password first, however:

```
~# cp /usr/share/doc/zarafa/zarafa.schema.gz /etc/ldap/schema/
~# gunzip /etc/ldap/schema/zarafa.schema.gz
~# slappasswd
```

Now include the newly generated ldap password hash in the slapd.conf after the 'rootpw' variable:

/etc/ldap/slapd.conf

```
# Schema and objectClass definitions, depending on your
# LDAP setup
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/inetorgperson.schema
include         /etc/ldap/schema/openldap.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/misc.schema
include         /etc/ldap/schema/zarafa.schema


# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile         /var/run/slapd/slapd.pid

# List of arguments that were passed to the server
argsfile        /var/run/slapd/slapd.args

# Read slapd.conf(5) for possible values
loglevel        8192

# Where the dynamically loaded modules are stored
modulepath      /usr/lib/ldap
moduleload      back_hdb

# The maximum number of entries that is returned for a search operation
sizelimit 500

# The tool-threads parameter sets the actual amount of cpu's that is used
# for indexing.
tool-threads 1

#######################################################################
# Specific Backend Directives for hdb:
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
backend         hdb

#######################################################################
# Specific Backend Directives for 'other':
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
```

```
#backend                <other>

#######################################################################
# Specific Directives for database #1, of type hdb:
# Database specific directives apply to this databasse until another
# 'database' directive occurs
database        hdb

# The base of your directory in database #1
suffix          "dc=example,dc=com"

# rootdn directive for specifying a superuser on the database. This is needed
# for syncrepl.
rootdn          "cn=admin,dc=example,dc=com"
rootpw          {SSHA}secret


# Where the database file are physically stored for database #1
directory       "/var/lib/ldap"

# The dbconfig settings are used to generate a DB_CONFIG file the first
# time slapd starts.  They do NOT override existing an existing DB_CONFIG
# file.  You should therefore change these settings in DB_CONFIG directly
# or remove DB_CONFIG and restart slapd for changes to take effect.

# For the Debian package we use 2MB as default but be sure to update this
# value if you have plenty of RAM
dbconfig set_cachesize 0 2097152 0

# Sven Hartge reported that he had to set this value incredibly high
# to get slapd running at all. See http://bugs.debian.org/303057 for more
# information.

# Number of objects that can be locked at the same time.
dbconfig set_lk_max_objects 1500
# Number of locks (both requested and granted)
dbconfig set_lk_max_locks 1500
# Number of lockers
dbconfig set_lk_max_lockers 1500

# Save the time that the entry gets modified, for database #1
lastmod         on

# Checkpoint the BerkeleyDB database periodically in case of system
# failure and to speed slapd shutdown.
checkpoint      512 30

# Where to store the replica logs for database #1
# replogfile    /var/lib/ldap/replog

# The userPassword by default can be changed
# by the entry owning it if they are authenticated.
# Others should not be able to see it, except the
# admin entry below
# These access lines apply to database #1 only
access to attrs=userPassword,shadowLastChange
        by dn="cn=admin,dc=example,dc=com" write
        by anonymous auth
        by self write
        by * none

# Ensure read access to the base for things like
# supportedSASLMechanisms.  Without this you may
# have problems with SASL not knowing what
# mechanisms are available and the like.
# Note that this is covered by the 'access to *'
# ACL below too but if you change that as people
# are wont to do you'll still need this if you
# want SASL (and possible other things) to work
# happily.
access to dn.base="" by * read
```

```
# The admin dn has full write access, everyone else
# can read everything.
access to *
        by dn="cn=admin,dc=example,dc=com" write
        by * read


# Indexing options for database #1
index objectClass eq
# zarafa indices
index cn eq
index gidNumber eq
index mail eq
index memberUid eq
index ou eq
index uid eq
index uidNumber eq
index uniqueMember eq
index zarafaAccount eq
index zarafaAliases eq
index zarafaViewPrivilege eq
```

Now (re)start the ldap server.

```
~# service slapd restart
```

Hint: If the server throws an error change the log level to '8' or '4', this should output enough info to give you an idea what is wrong. LDAP logs to syslog in Ubuntu.

LDAP Admin Setup

Create a .ldif file to load into LDAP:

/tmp/admin.ldif

```
dn:dc=example,dc=com
objectClass: dcObject
objectClass: organization
o: example
dc: example

dn:cn=admin,dc=example,dc=com
objectClass: organizationalRole
cn: admin
```

Load the template into the LDAP database (you will be asked for the password you generated with 'slappasswd' before):

```
~# ldapadd -x -W -D cn=admin,dc=example,dc=com -f /tmp/admin.ldif
```

If something breaks it is always an option to delete the whole **/var/lib/ldap/*** folder and start over. However don't forget to set ownership and permissions on that folder correctly (**openldap:openldap**, **u+rwX**) if you do so!

Configure phpldapadmin

The config is somewhat big but most are comments and defaults that need not to be modified. Again only lines that were modified are shown.

/etc/phpldapadmin/config.php

```
/* phpLDAPadmin can encrypt the content of sensitive cookies if you set this
   to a big random string. */
$config->custom->session['blowfish'] = secret;

/* The language setting. If you set this to 'auto', phpLDAPadmin will attempt
   to determine your language automatically. Otherwise, available lanaguages
   are: 'ct', 'de', 'en', 'es', 'fr', 'it', 'nl', and 'ru'
   Localization is not complete yet, but most strings have been translated.
   Please help by writing language files. See lang/en.php for an example. */
$config->custom->appearance['language'] = 'en';

/* Our local timezone
   This is to make sure that when we ask the system for the current time, we
   get the right local time. If this is not set, all time() calculations will
   assume UTC if you have not set PHP date.timezone. */
// $config->custom->appearance['timezone'] = null;
$config->custom->appearance['timezone'] = 'America/Los_Angeles';

/* Hide the warnings for invalid objectClasses/attributes in templates. */
$config->custom->appearance['hide_template_warning'] = true;

/* A convenient name that will appear in the tree viewer and throughout
   phpLDAPadmin to identify this LDAP server to users. */
$servers->setValue('server','name','example.com');

/* Array of base DNs of your LDAP server. Leave this blank to have phpLDAPadmin
   auto-detect it for you. */
// $servers->setValue('server','base',array('dc=example,dc=com'));

/* The DN of the user for phpLDAPadmin to bind with. For anonymous binds or
   'cookie' or 'session' auth_types, LEAVE THE LOGIN_DN AND LOGIN_PASS BLANK. If
   you specify a login_attr in conjunction with a cookie or session auth_type,
   then you can also specify the bind_id/bind_pass here for searching the
   directory for users (ie, if your LDAP server does not allow anonymous binds. */
// $servers->setValue('login','bind_id','');
$servers->setValue('login','bind_id','cn=admin,dc=example,dc=com');
```

You should be able to access your server under http://example.com/phpldapadmin now. the 'User' field will be prefilled with 'cn=admin,dc=example, dc=com' and you can login with the password you generated with 'slappasswd' before.

After login create a new '**Security Object**' called '**zarafaservice**' and assign it a password. This will be the user which zarafa and postfix will use to access LDAP. If done correctly phpldapadmin should display a **dn of 'uid=zarafaservice,dc=example,dc=com'** for the user. Now adjust slapd.conf to grant the user read access to the tree. Put the entry between the 'access to attrs' and 'access to *' parameters:

/etc/ldap/slapd.conf

```
# The userPassword by default can be changed
# by the entry owning it if they are authenticated.
# Others should not be able to see it, except the
# admin entry below
# These access lines apply to database #1 only
access to attrs=userPassword,shadowLastChange
        by dn="cn=admin,dc=example,dc=com" write
        by anonymous auth
        by self write
        by * none

# !! NEW !! #
access to *
        by dn="uid=zarafaservice,dc=example,dc=com" read
# !! NEW !! #

# The admin dn has full write access, everyone else
# can read everything.
access to *
        by dn="cn=admin,dc=example,dc=com" write
        by * read
```

Install Zarafa phpldapadmin templates

Download the templates that are attached to this page and put them into the following folders:

- **create- zarafaAccount.xml** and **create-zarafaCompany.xml** into **/etc/phpldapadmin/templates/creation/**
- **modify- zarafaAccount.xml** and **modify-zarafaCompany.xml** into **/etc/phpldapadmin/templates/modification/**

Create a Company, User and Alias

With the templates in place, log into phpldapadmin again and:

- create a company (Zarafa Company)
- create a user in the newly created company (Zarafa User Account)
- edit the user, add new attribute 'zarafaAlias'
    - assign at least one email alias to the user

This user and the mailalias will be needed later for testing the functionality of the server.

Zarafa Server Setup

Be sure to read the admin guide, this will cover only what needs to be changed so zarafa will work in this specific setup. All Zarafa server configuration files are stored in '**/etc/zarafa**'.

Official docs for Zarafa 7.2 are still in the 'trunk' of the SVN repo.

Zarafa system user

As no service should run as 'root'-user we create a 'zarafa' system account first:

```
~# groupadd -g 5000 zarafa
~# useradd -c 'User for ZCP' -d '/var/lib/zarafa' -g 5000 -M -N -s '/bin/false' -u 5000 zarafa
```

Zarafa Folders

Zarafa writes / logs to various folders which now need to be owned to the newly created user. Also for the webapp to function correctly we need to give access rights to some folders to the **www-data** user.

```
~# chown -R zarafa: /var/log/zarafa /var/lib/zarafa
~# chown -R www-data: /var/lib/zarafa-webaccess/tmp /var/lib/zarafa-webapp/tmp
```

zarafa-server and ldap.conf

The option 'enable_hosted_zarafa' will switch on multi-tenancy. If you don't want this feature set it to 'false'.

/etc/zarafa/server.cfg

```
server_bind             = 10.1.50.100

# Name for identifying the server in a multi-server environment
server_name = example.com

# local admin users who can connect to any store (use this for the zarafa-dagent)
# field is SPACE separated
local_admin_users       = root zarafa

# drop privileges and run the process as this user
run_as_user             = zarafa

# drop privileges and run the process as this group
run_as_group            = zarafa

# Database engine (mysql)
database_engine         = mysql

# e-mail address of the Zarafa System user
system_email_address    = postmaster@example.com

# The user under which we connect with MySQL
mysql_user              = zarafa

# The password for the user (leave empty for no password)
mysql_password          = secret

# Database to connect to
mysql_database          = zarafa

# Where to place attachments. Value can be 'database' or 'files'
attachment_storage      = database

# enable SSL support in server
server_ssl_enabled      = yes

# Listen for SSL connections on this port
server_ssl_port         = 237

# Required Server certificate, contains the certificate and the private key parts
server_ssl_key_file     = /etc/zarafa/ssl/mail.crt

# Password of Server certificate
server_ssl_key_pass     =

# Required Certificate Authority of server
server_ssl_ca_file      = /etc/zarafa/ssl/mail.key

# Path with CA certificates, e.g. /etc/ssl/certs
server_ssl_ca_path      = /etc/ssl/certs

# Path of SSL Public keys of clients
sslkeys_path            = /etc/zarafa/ssl

# Name of the plugin that handles users
```

```
# Required, default = ldap
# Values: ldap, unix, db, ldapms (available in enterprise license)
user_plugin             = ldap

# configuration file of the user plugin, examples can be found in /usr/share/zarafa/example-config
user_plugin_config      = /etc/zarafa/ldap.cfg

# workaround for bug in search server
search_socket = file:///var/tmp/zarafa/zarafa-search
# Enable multi-tenancy environment
# When set to true it is possible to create tenants within the
# zarafa instance and assign all users and groups to particular
# tenants.
# When set to false, the normal single-tenancy environment is created.
enable_hosted_zarafa = true

# Use Indexing service for faster searching.
# Enabling this option requires the zarafa-indexer service to
# be running.
index_services_enabled = yes

# Path to the zarafa-indexer service, this option is only required
# if the server is going to make use of the indexing service.
index_services_path = file:///var/run/zarafa-indexer

# Time (in seconds) to wait for a connection to the zarafa-indexer service
# before terminating the indexed search request.
index_services_search_timeout = 10

# Allow enhanced ICS operations to speedup synchronization with cached profiles.
# default: yes
enable_enhanced_ics = yes
```

Copy the openldap configuration template to ldap.conf before you start editing.

/etc/zarafa/ldap.cfg

```
ldap_bind_user = userid=zarafaservice,dc=example,dc=com

# LDAP bind password
# Optional, default = empty (no password)
ldap_bind_passwd = secret

# Top level search base, every object should be available under this tree
ldap_search_base = dc=example,dc=com

# attribute name which is/(should: was) used in ldap_user_search_filter
ldap_object_type_attribute = objectClass
ldap_user_type_attribute_value = zarafa-user
ldap_group_type_attribute_value = zarafa-group
ldap_contact_type_attribute_value = zarafa-contact
ldap_company_type_attribute_value = organizationalUnit
ldap_addresslist_type_attribute_value = zarafa-addresslist
ldap_dynamicgroup_type_attribute_value = zarafa-dynamicgroup

# Optional, default = empty (match everything)
# For active directory, use:
#   (objectCategory=Person)
# For LDAP with posix users:
#   no need to use the search filter.
ldap_user_search_filter = (objectClass=zarafa-user)

# unique user id for find the user
# Note: contacts also use this field for uniqueness. If you change this,
# you might need to update the zarafa.schema file too, and change
# the MUST uidNumber to whatever you set here.dnl
ldap_user_unique_attribute = entryUUID

# Type of unique user id
# default: text
# For active directory, use:
#               binary
# For LDAP with posix user, use:
#               text
ldap_user_unique_attribute_type = text

# If set to bind, users are authenticated by trying to bind to the
# LDAP tree using their username + password.  Otherwise, the
# ldap_password_attribute is requested and checked.
ldap_authentication_method = bind

# Group settings
# Search for groups using this LDAP filter.
ldap_group_search_filter = (objectClass=zarafa-group)

# Company settings
# Search for companies using this LDAP filter.
ldap_company_search_filter = (objectClass=zarafa-company)
# unique company id for find the company
# Active directory: objectGUID
# LDAP: ou
ldap_company_unique_attribute = ou

# Optional, default = ou
# Active directory: ou
# LDAP: ou
ldap_companyname_attribute = ou

# Mapping from the quota attributes to a number of bytes.  Qmail-LDAP
# schema uses bytes (1), ADS uses kilobytes (1024*1024).
# We need to ajust this to 1MB in Bytes so we can use MB in phpldapadmin,
# otherwise quota won't work correctly.
ldap_quota_multiplier = 1048576
```

zarafa-dagent

This is the service which talks to postfix and delivers mails to zarafa postboxes. The dagent must be enabled in zarafa's default file. We adjust some values for spam management here that will later tie in nicely with spamassassin:

/etc/default/zarafa

```
DAGENT_ENABLED=yes
DAGENT_CONFIG=/etc/zarafa/dagent.cfg
DAGENT_OPTS="-d"
```

/etc/zarafa/dagent.cfg

```
# drop privileges and run the process as this user
run_as_user             = zarafa

# drop privileges and run the process as this group
run_as_group            = zarafa

# The following e-mail header will mark the mail as spam, so the mail
# is placed in the Junk Mail folder, and not the Inbox.
# possibly other viable spam headers:
#  spamassassin: "X-Spam-Flag: Yes"
#  dspam:        "X-DSPAM-Result: Spam"
spam_header_name = X-Spam-Status

# If the above header is found, and contains the following value
# the mail will be considered as spam.
spam_header_value = Yes,
```

zarafa-monitor

Daemon which sends quota warnings etc. Not much to adjust except the user.

/etc/zarafa/monitor.cfg

```
# drop privileges and run the process as this user
run_as_user = zarafa


# drop privileges and run the process as this group
run_as_group =
```

zarafa-gateway

This service talks pops and imaps to the outside world. Unencrypted protocols should no longer be needed. Enable them only if you need to support legacy clients.

/etc/zarafa/dagent.cfg

```
# Set this value to a name to show in the logon greeting to clients.
# Leave empty to use DNS to find this name.
server_hostname = example.com

# Whether to show the hostname in the logon greeting to clients.
server_hostname_greeting = yes

# drop privileges and run the process as this user
run_as_user             = zarafa

# drop privileges and run the process as this group
run_as_group            = zarafa

# default connection to the Zarafa server
# Please refer to the administrator manual or manpage why HTTP is used rather than the UNIX socket.
server_socket = http://10.1.50.100:236/zarafa

# enable/disable POP3, and POP3 listen port
pop3_enable     =       no
pop3_port       =       110

# enable/disable Secure POP3, and Secure POP3 listen port
pop3s_enable    =       yes
pop3s_port      =       995

# enable/disable IMAP, and IMAP listen port
imap_enable     =       no
imap_port       =       143

# enable/disable Secure IMAP, and Secure IMAP listen port
imaps_enable    =       yes
imaps_port      =       993

# File with RSA key for SSL
ssl_private_key_file    =       /etc/zarafa/ssl/mail.key

#File with certificate for SSL
ssl_certificate_file    =       /etc/zarafa/ssl/mail.crt

# Disable all plaintext authentications unless SSL/TLS is used
# must be disabled, otherwise rimap auth doesn't work
disable_plaintext_auth = no
```

zarafa-ical

This is the iCal / CALDAV Server. The preferred method is CALDAV though.

/etc/zarafa/ical.cfg

```
# whether normal connections can be made to the ical server
ical_enable = yes

# drop privileges and run the process as this user
run_as_user             = zarafa

# drop privileges and run the process as this group
run_as_group            = zarafa

# port which the ical server listens on for normal connections
ical_port = 8080

# default connection to the Zarafa server
# Please refer to the administrator manual or manpage why HTTP is used rather than the UNIX socket.
server_socket = http://10.1.50.100:236/zarafa

# wether ssl connections can be made to the ical server
icals_enable = yes

# port which the ical server listens on for ssl connections
icals_port = 8443

# File with RSA key for SSL
ssl_private_key_file = /etc/zarafa/ssl/web.key

# File with certificate for SSL
ssl_certificate_file = /etc/zarafa2/ssl/web.crt
```

zarafa-search

There seems to be a problem with the search index in 7.2. It exhibits runtime problems (or generally refuses to start up) if the pidfile and socket are in '**/var/run**' (which is a symlink to /run in Ubuntu, which is a ramdisk). A workaround seems to be to create a folder like '**/var/tmp/zarafa**' and configure server.cfg and search.cfg to use this folder instead of '**/run**'

If the search server doesn't start up be sure to clear out stale pid or lock files from '**/var/run**'.

The xapian-based search index which is utilized by Zarafa for fulltext search in the mailstores. There are multiple options for indexing mail attachments under the "ATTACHMENT INDEX SETTINGS" paragraph. As they are pretty self-explanatory and heavily dependent of the use-case, I wont give any recommendations for them. Set at least the user/group, as always:

/etc/zarafa/search.cfg

```
# run as specific user
run_as_user         = zarafa

# run as specific group
run_as_group        = zarafa

pid_file            = /var/tmp/zarafa/zarafa-search.pid

server_bind_name    = file:///var/tmp/zarafa/zarafa-search
```

Create the new folder and set permissions:

```
~# mkdir /var/tmp/zarafa
~# chown -R zarafa: /var/tmp/zarafa
```

Finally fix the the path to the pidfile in the init script. Change the PIDFILE variable at the beginning of the init script:

/etc/init.d/zarafa-search

```
PIDFILE=/var/tmp/zarafa/$NAME.pid
```

zarafa-spooler

This service delivers mail to postfix to be sent into the intertubes. Luckily nothing must be adjusted here except the user and group settings.

/etc/zarafa/spooler.cfg

```
# set our internal smtp interface for local mail
smtp_server      = 10.1.50.100

# drop privileges and run the process as this user
run_as_user      = zarafa

# drop privileges and run the process as this group
run_as_group     = zarafa

# set this if you want outlook-style meeting requests
always_send_tnef = yes
```

Z-Push Setup (ActiveSync)

Get the source from https://z-push.org and unpack to '**/usr/share/**'. Create a symlink from the directory to '**/usr/share/z-push**' and you are done. (The same company that develops zarafa also develops z-push so all the defaults fit for the zarafa deployment).

The directories '**/var/lib/z-push**' '**/var/log/z-push**' have to exist and be writeable by the webserver, so don't forget to chown them to the **www-data**-user. We also need to set up an apache alias so mobile devices that use Active Sync work correctly. The following example is later incorporated in the apache config (see below):

```
Alias /Microsoft-Server-ActiveSync /usr/share/z-push/index.php
```

```
~# cd /usr/share
~# wget http://download.z-push.org/final/2.1/z-push-2.1.3-1892.tar.gz/<...>/z-push-<latest-version>.tar.gz  #
only an example, get correct URL from z-push homepage
~# tar xzv z-push-<latest-version>.tar-gz
~# ln -s /usr/share/z-push-<version-number> /usr/share/z-push
~# mkdir /var/lib/z-push
~# mkdir /var/log/z-push
~# chown -R www-data: /var/lib/z-push
~# chown -R www-data: /var/log/z-push
```

Adjust the timezone in '**config.php**':

/usr/share/z-push/config.php

```
*  Default settings
 */
    // Defines the default time zone, change e.g. to "Europe/London" if necessary
    define('TIMEZONE', 'America/Los_Angeles');
```

Apache2 Setup

We have to enable the rewrite and ssl modules and disable the default website. Also ports.conf has to be adjusted a little, otherwise apache barfs at startup.

```
~# a2enmod alias rewrite ssl  # make sure all necessary modules are enabled
~# a2dissite default

~# rm -rf /var/www/html       # optional: delete default welcome page
```

/etc/apache2/ports.conf

```
NameVirtualHost 1.2.3.4:80
```

For webaccess we make sure **everything** gets rewritten to https.

/etc/apache2/sites-enabled/zarafa-webaccess

```
<VirtualHost 1.2.3.4:80>
  ServerName      zarafa.example.com
  ServerAdmin     webmaster@example.com

  RewriteEngine   On
  RewriteCond     %{HTTPS}        off
  RewriteRule     ^(.*)$          https://%{SERVER_NAME}%{REQUEST_URI} [L,R]

</VirtualHost>

<VirtualHost 1.2.3.4:443>

 ServerName      zarafa.example.com
 ServerAdmin     webmaster@example.com

 SSLEngine              on
 SSLCertificateFile     /etc/zarafa/ssl/web.crt
 SSLCertificateKeyFile  /etc/zarafa2/ssl/web.key

 Alias /webaccess       /usr/share/zarafa-webaccess
 Alias /z-push          /usr/share/z-push/index.php
 Alias /Microsoft-Server-ActiveSync /usr/share/z-push/index.php
  <Directory /usr/share/zarafa-webaccess/>
    DirectoryIndex index.php
    Options -Indexes +FollowSymLinks
    AllowOverride Options

    Order allow,deny
    Allow from all
  </Directory>

  # z-push php settings
  <Directory /usr/share/z-push>
    php_flag magic_quotes_gpc off
    php_flag register_globals off
    php_flag magic_quotes_runtime off
    php_flag short_open_tag on
  </Directory>

</VirtualHost>
```

Syncing LDAP data to Zarafa

Every time you change things in LDAP you have to sync those changes to the Zarafa DB. Invoke the command 'zarafa-admin --sync' to do so. The option '-l' displays all users, use this to test if the synchronization worked.

```
~# zarafa-admin --sync
~# zarafa-admin -l
```

That's it. Now you should be able to log in to the Zarafa Webaccess with the user we configured with phpopenldap before. You should also be able to log on to the Server via a smartphone - just use 'Exchange' as account type and put in your server's and user's details.

postfix, sasl, spamassassin and clamav setup

postfix must be configured to do ldap user lookups which is done via ldap-users.cf and ldap-aliases.cf. Futher main.cf and master.cf need to be adjusted to scan mail for spam and viruses and deliver them to dagent via lmtp in the end. To authenticate users who are sending mails from external addresses we will use SASL with the remote IMAP (rimap) method, which basically performs a IMAP-login to check if the supplied password is correct.

postfix

This configuration contains restrictive DNSBL postscreen filtering. This means that postfix will refuse to accept the mail in the first place and immediately send a 5** error code. This also means the mail won't ever be considered as delivered or show up in your mailqueue. Be sure to understand what this means and please read the manual!

This postfix config also contains enhanced recipient restrictions and mailq timeouts. If you do not know what they do you shouldn't use them.

main.cf

/etc/postfix/main.cf

```
smtpd_banner            = $myhostname ESMTP NO UCE
myhostname              = mail.example.com
biff                    = no
append_dot_mydomain     = no
mynetworks              = 127.0.0.0/8, 10.1.0.0/16
recipient_delimiter     = +
inet_interfaces         = all
inet_protocols          = ipv4
myorigin                = $myhostname
mydestination           = $myhostname localhost.example.com, localhost

virtual_mailbox_domains = example.com, example.net, example.org
virtual_mailbox_maps    = ldap:/etc/postfix/ldap-users.cf
virtual_alias_maps      = ldap:/etc/postfix/ldap-aliases.cf
virtual_transport       = lmtp:127.0.0.1:2003

# SASL
smtpd_sasl_auth_enable          = yes
broken_sasl_auth_clients        = yes

# TLS encryptiion
smtpd_tls_security_level        = may
smtpd_tls_auth_only             = yes
smtpd_tls_cert_file             = /etc/postfix/keys/postfix.crt
smtpd_tls_key_file              = /etc/postfix/keys/postfix.key
smtpd_tls_CAfile                = /etc/postfix/keys/postfix.pem
smtpd_tls_loglevel              = 0
smtpd_tls_received_header       = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source               = dev:/dev/urandom

### Before-220 tests (postscreen / DNSBL)
postscreen_access_list          = permit_mynetworks, cidr:/etc/postfix/postscreen_access.cidr
postscreen_dnsbl_reply_map      = pcre:/etc/postfix/postscreen_dnsbl_reply_map.pcre
postscreen_blacklist_action     = drop
postscreen_dnsbl_action         = enforce
postscreen_greet_action         = enforce
postscreen_dnsbl_threshold      = 4
postscreen_dnsbl_sites =
```

```
        zen.spamhaus.org*3
        b.barracudacentral.org*2
        bl.spameatingmonkey.net*2
        bl.spamcop.net
        dnsbl.sorbs.net
        psbl.surriel.com
        bl.mailspike.net
        swl.spamhaus.org*-4
postscreen_whitelist_interfaces = $mynetworks, static:all

### End of before-220 tests
### After-220 tests
### WARNING -- See "Tests after the 220 SMTP server greeting" in the
### Postscreen Howto and *UNDERSTAND* it *BEFORE* you enable the
### following tests! This basically enables some kind of greylisting!
#postscreen_bare_newline_action = enforce
#postscreen_bare_newline_enable = yes
#postscreen_non_smtp_command_enable = yes
#postscreen_pipelining_enable = yes
### ADDENDUM: Any one of the foregoing three *_enable settings may cause
### significant and annoying mail delays.


# CLAMAV integration via clamsmtp proxy
content_filter                  = scan:127.0.0.1:10025
receive_override_options        = no_address_mappings

# check incoming mail for 'stuff'
smtpd_recipient_restrictions =
        permit_mynetworks,
        permit_sasl_authenticated,
        reject_unknown_recipient_domain,
        reject_non_fqdn_recipient,
        reject_non_fqdn_hostname,
        reject_non_fqdn_sender,
        reject_unauth_destination,
        reject_unauth_pipelining,
        reject_invalid_hostname

smtpd_data_restrictions =
        reject_unauth_pipelining

# client restrictions
smtpd_client_restrictions =
        permit_mynetworks,
        permit_auth_destination,
        permit_sasl_authenticated

# anybody out there?
smtpd_helo_restrictions =
        permit_mynetworks,
        reject_invalid_hostname,
        reject_non_fqdn_hostname

# who may send
smtpd_sender_restrictions =
        reject_unknown_sender_domain,
        reject_non_fqdn_sender,
        permit_sasl_authenticated,
        permit_mynetworks

# mail reject codes
unknown_address_reject_code             = 554
unknown_client_reject_code              = 554
unknown_hostname_reject_code            = 554
unknown_local_recipient_reject_code     = 554
unknown_relay_recipient_reject_code     = 554
unknown_virtual_alias_reject_code       = 550
unknown_virtual_mailbox_reject_code     = 550
# deferred mail intervals
```

```
# (default: 300 seconds; before Postfix 2.4: 1000s)
# How often the queue manager scans the queue for deferred mail.
queue_run_delay                 = 900

# (default: 300 seconds; before Postfix 2.4: 1000s)
# The minimal amount of time a message won't be looked at, and the minimal amount of time to stay away from a
"dead" destination.
minimal_backoff_time            = 450

# (default: 4000 seconds)
# The maximal amount of time a message won't be looked at after a delivery failure.
maximal_backoff_time            = 1800

# (default: 5 days)
# How long a message stays in the queue before it is sent back as undeliverable. Specify 0 for mail that should
be returned immediately after the first unsuccessful delivery attempt.
maximal_queue_lifetime          = 14

# (default: 5 days, available with Postfix version 2.1 and later)
# How long a MAILER-DAEMON message stays in the queue before it is considered undeliverable. Specify 0 for mail
that should be tried only once.
bounce_queue_lifetime           = 14

# (default: 20000)
# The size of many in-memory queue manager data structures. Among others, this parameter limits the size of the
short-term, in-memory list of "dead" destinations. Destinations that don't fit the list are not added.
qmgr_message_recipient_limit    = 1000000

# max message size (15M)
message_size_limit              = 15360000
```

/etc/postfix/postscreen_dnsbl_reply_map.pcre

```
# We will be rejecting much mail which is listed in multiple DNSBLs.
# We're not proud of some of the lists we are using, thus have given
# them lower scores in postscreen_dnsbl_sites listing. So this checks
# the DNSBL name postscreen(8) gets from dnsblog(8), and if it's not
# one of our Tier 1 DNSBL sites, it changes what the sender will see:

!/^zen\.spamhaus\.org$/         multiple DNS-based blocklists
```

postfix ACL update script

This script will pull updates from zenhaus.org

/usr/local/bin/lasso-update.sh

```
#!/bin/bash
URL=http://www.spamhaus.org/drop/drop.lasso
DROPLASSO=$(mktemp)
ACLFILE=$(mktemp)
POSTSCREEN_ACCESS_FILE=/etc/postfix/postscreen_access.cidr
DATE=$(date +%Y%m%d)
wget --quiet $URL -O $DROPLASSO
if [ -e $DROPLASSO ]
then
  for network in $(cat $DROPLASSO | awk '{print $1}' |\
    grep -E "^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\/[0-9]{1,2}$")
  do
    echo -e "$network\treject" >> $ACLFILE
  done
fi
# update the cidr file with our local ip addresses
echo -e "1.2.3.4/56\tpermit" >> $ACLFILE  # put your official ip/subnet here, otherwise postscreen might reject
you immediately!
echo -e "10.1.50.0/16\tpermit" >> $ACLFILE
echo -e "127.0.0.0/8\tpermit" >> $ACLFILE

cp $ACLFILE $POSTSCREEN_ACCESS_FILE
/usr/sbin/service postfix reload
```

Put this in your crontab to run it regularly:

/etc/cron.d/sarulesupdate

```
# minute (0-59),
# |     hour (0-23),
# |     |       day of the month (1-31),
# |     |       |       month of the year (1-12),
# |     |       |       |       day of the week (0-7 with 0=7=Sunday).
# |     |       |       |       |       user
# |     |       |       |       |       |       command
10      6,12    *       *       *       root    /usr/local/bin/lasso-update.sh
```

master.cf

The master transport file. Transport decisions are made here, in our case:
internet > postscreen (DNSBL) -> smpt:postfix -> socket:spamassassin -> content_filter:clamav (via clamsmtp) -> smtp:postfix -> lmtp:dagent

> **Notice:** lmtp must not run chrooted (look at the fifth column), otherwise it won't be able to talk to dagent!

/etc/postfix/master.cf

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtpd     pass  -       -       n       -       -       smtpd
  -o content_filter=spamassassin
smtp      inet  n       -       n       -       1       postscreen
tlsproxy  unix  -       -       n       -       0       tlsproxy
dnsblog   unix  -       -       n       -       0       dnsblog
submission inet n       -       n       -       -       smtpd
  -o content_filter=spamassassin
smtps     inet  n       -       n       -       -       smtpd
 -o smtpd_tls_wrappermode=yes
 -o smtpd_sasl_auth_enable=yes
 -o content_filter=spamassassin
pickup    fifo  n       -       -       60      1       pickup
cleanup   unix  n       -       -       -       0       cleanup
qmgr      fifo  n       -       n       300     1       qmgr
tlsmgr    unix  -       -       -       1000?   1       tlsmgr
rewrite   unix  -       -       -       -       -       trivial-rewrite
bounce    unix  -       -       -       -       0       bounce
defer     unix  -       -       -       -       0       bounce
trace     unix  -       -       -       -       0       bounce
verify    unix  -       -       -       -       1       verify
flush     unix  n       -       -       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
proxywrite unix -       -       n       -       1       proxymap
smtp      unix  -       -       -       -       -       smtp
# When relaying mail as backup MX, disable fallback_relay to avoid MX loops
relay     unix  -       -       -       -       -       smtp
        -o smtp_fallback_relay=
showq     unix  n       -       -       -       -       showq
error     unix  -       -       -       -       -       error
retry     unix  -       -       -       -       -       error
discard   unix  -       -       -       -       -       discard
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       n       -       -       lmtp
anvil     unix  -       -       -       -       1       anvil
scache    unix  -       -       -       -       1       scache
scan      unix  -       -       n       -       16      smtp
        -o smtp_send_xforward_command=yes


# reinjection from spamassassin into mailflow after checks
127.0.0.1:10026 inet    n       -       n       -       -       smtpd
        -o content_filter=
        -o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
        -o smtpd_helo_restrictions=
        -o smtpd_client_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks=127.0.0.0/8
        -o smtpd_authorized_xforward_hosts=127.0.0.0/8

spamassassin unix  -       n       n       -       -       pipe
   user=debian-spamd argv=/usr/bin/spamc -e /usr/sbin/sendmail -oi -f ${sender} ${recipient}
```

ldap users and aliases

These files control ldap user and alias lookups

/etc/postfix/ldap-users.cf

```
server_host     = localhost
search_base     = dc=example,dc=com
version         = 3
bind            = yes
bind_dn         = uid=zarafaservice,dc=example,dc=com
bind_pw         = secret
scope           = sub
query_filter    = (mail=%s)
result_attribute        = mail
```

/etc/postfix/ldap-aliases.cf

```
server_host     = localhost
search_base     = dc=example,dc=com
version         = 3
bind            = yes
bind_dn         = uid=zarafaservice,dc=example,dc=com
bind_pw         = secret
scope           = sub
query_filter    = (zarafaAliases=%s)
result_attribute        = mail
```

Test LDAP connectivity

Test your setup by issuing the following commands. In both cases the email address of the user should be displayed. If this doesn't work then postfix won't be able to deliver email as it can't find email addresses or aliases in the LDAP directory!

```
~# postmap -q <insert-username-here>@example.com ldap:/etc/postfix/ldap-users.cf
~# postmap -q <insert-user-alias-here>@example.com ldap:/etc/postfix/ldap-aliases.cf
```

SASL

We have to add postfix to the sasl group and edit two files to make it work:

/etc/default/sasl

```
START=yes
DESC="SASL Authentication Daemon"
NAME="saslauthd"
MECHANISMS="rimap"
MECH_OPTIONS="127.0.0.1"
THREADS=5
OPTIONS="-c -m /var/run/saslauthd -r"
```

/etc/postfix/sasl/smtpd.conf

```
pwcheck_method: saslauthd
mech_list: plain login
```

add postfix to sasl

```
~# gpasswd -a postfix sasl
~# service postfix restart
~# service saslauthd restart
```

SpamAssassin

postfix will pass emails that passed the postscreen tests to spamassassin for various checks. The default config is pretty sane, except for some reason it runs as root per default - we want to change that. First we enable the service:

/etc/default/spamassassin

```
ENABLED=1
OPTIONS="--create-prefs --max-children 5 --helper-home-dir -u debian-spamd -g debian-spamd"
CRON=1
```

All configuration files reside in **/etc/spamassassin/**. Edit the following file:

/etc/spamassassin/local.cf

```
rewrite_header Subject *****SPAM*****
report_safe 1
trusted_networks
lock_method flock # don't set this if your / is on NFS! (pxe-boot etc.)
required_score 5.0
bayes_ignore_header X-Bogosity
bayes_ignore_header X-Spam-Flag
bayes_ignore_header X-Spam-Status
```

Create a site-specific config file and add:

/etc/spamassassin/99_examplecom.cf

```
# Enable Spam Networks
use_razor2              1
use_pyzor               1

# bayes settings
bayes_auto_learn        1
bayes_path              /var/lib/spamassassin/bayes
bayes_file_mode         0660

# Whitelist important senders
whitelist_from          *@example.com  # <-- change this to your trusted domain(s)
whitelist_from          *@example2.com

# mailspike dnsbl integration


## Spam sources
header __RCVD_IN_MSPIKE          eval:check_rbl('mspike-lastexternal', 'bl.mailspike.net.')
tflags __RCVD_IN_MSPIKE          net


## Reputation compensations
## Definitions
header __RCVD_IN_MSPIKE_Z     eval:check_rbl_sub('mspike-lastexternal', '^127\.0\.0\.2$')
describe __RCVD_IN_MSPIKE_Z   Spam wave participant
tflags __RCVD_IN_MSPIKE_Z     net
header RCVD_IN_MSPIKE_L5     eval:check_rbl_sub('mspike-lastexternal', '^127\.0\.0\.10$')
describe RCVD_IN_MSPIKE_L5   Very bad reputation (-5)
tflags RCVD_IN_MSPIKE_L5     net
header RCVD_IN_MSPIKE_L4     eval:check_rbl_sub('mspike-lastexternal', '^127\.0\.0\.11$')
describe RCVD_IN_MSPIKE_L4   Bad reputation (-4)
tflags RCVD_IN_MSPIKE_L4     net
header RCVD_IN_MSPIKE_L3     eval:check_rbl_sub('mspike-lastexternal', '^127\.0\.0\.12$')
describe RCVD_IN_MSPIKE_L3   Low reputation (-3)
tflags RCVD_IN_MSPIKE_L3     net


# *_L and *_Z may overlap each other, so account for that
meta __RCVD_IN_MSPIKE_LOW RCVD_IN_MSPIKE_L5 || RCVD_IN_MSPIKE_L4 || RCVD_IN_MSPIKE_L3
meta RCVD_IN_MSPIKE_ZBI __RCVD_IN_MSPIKE_Z && !__RCVD_IN_MSPIKE_LOW
## Scores
score RCVD_IN_MSPIKE_ZBI     4.1
score RCVD_IN_MSPIKE_L5      4.1
score RCVD_IN_MSPIKE_L4      3.5
score RCVD_IN_MSPIKE_L3      2.9


# aditional rules to hit spam
header SMF_BRACKETS_TO To:raw =~ /<<[^<>]+>>/
describe SMF_BRACKETS_TO Double-brackets around To header address
score SMF_BRACKETS_TO 1.5
```

Create the folder for the bayes databases, download an initial set of rules, test the setup, compile the rules and restart spamassassin:

```
~# mkdir -p /var/lib/spamassassin/bayes
~# chown -R debian-spamd:debian-spamd /var/lib/spamassassin
~# su - debian-spamd -c "sa-update --nogpg --channel updates.spamassassin.org --channel sought.rules.yerp.org --
channel sa.zmi.at --channel spamassassin.heinlein-support.de"
~# su - debian-spamd -c "spamassassin --lint"
~# su - debian-spamd -c "sa-compile"
~# service spamassassin restart
```

ClamAV and clamsmtp

/etc/clamav.clamd.cfg

```
LocalSocket /var/run/clamav/clamd.ctl
FixStaleSocket true
LocalSocketGroup clamav
LocalSocketMode 666
User clamav
AllowSupplementaryGroups true
ScanMail true
ScanArchive true
ArchiveBlockEncrypted false
MaxDirectoryRecursion 15
FollowDirectorySymlinks false
FollowFileSymlinks false
ReadTimeout 180
MaxThreads 12
MaxConnectionQueueLength 15
LogSyslog true
LogFacility LOG_LOCAL6
LogClean false
LogVerbose false
PidFile /var/run/clamav/clamd.pid
DatabaseDirectory /var/lib/clamav
SelfCheck 3600
Foreground false
Debug false
ScanPE true
ScanOLE2 true
ScanHTML true
DetectBrokenExecutables false
ExitOnOOM false
LeaveTemporaryFiles false
AlgorithmicDetection true
ScanELF true
IdleTimeout 30
PhishingSignatures true
PhishingScanURLs true
PhishingAlwaysBlockSSLMismatch false
PhishingAlwaysBlockCloak false
DetectPUA false
ScanPartialMessages false
HeuristicScanPrecedence false
StructuredDataDetection false
CommandReadTimeout 5
SendBufTimeout 200
MaxQueue 100
ExtendedDetectionInfo true
OLE2BlockMacros false
StreamMaxLength 25M
Bytecode true
BytecodeSecurity TrustSigned
BytecodeTimeout 60000
OfficialDatabaseOnly false
CrossFilesystems true
TCPSocket 3310
```

Now we only need to configure the clamsmtp proy to pass mails between postfix and clamav. Adjust only the following parameters:

/etc/clamsmtpd.conf

```
OutAddress:  10026
Listen:      127.0.0.1:10025
ClamAddress: /var/run/clamav/clamd.ctl
```

Tying It All Together

Now take a deep breath, you're almost there. If all went well, all configs fit and all permissions are set, after a healthy restart of the services you should have a complete ZCP setup up and running. Congratulations!

```
~# service clamav-daemon restart
~# service clamav-freshclam restart
~# service clamsmtp restart
~# service postfix restart
~# service saslauthd restart
~# service spamassassin restart

# after the following command be sure to check all zarafa processes, especially zarafa-search!
~# for i in $(ls /etc/init.d/zarafa-*); do ${i} restart; done
```

Regular Maintenance

Drop this file into '/etc/cron.d'. It's pretty self-explanatory, however you might want adjust the values and/or intervals.

/etc/cron.d/mailjobs

```
# minute (0-59),
# |     hour (0-23),
# |     |       day of the month (1-31),
# |     |       |       month of the year (1-12),
# |     |       |       |       day of the week (0-7 with 0=7=Sunday).
# |     |       |       |       |       user
# |     |       |       |       |       |       command

# sync LDAP to Zarafa regularly
*/15    *       *       *       *       root    /usr/bin/zarafa-admin --sync

# purge soft-deleted items after 30 days
3       30      *       *       *       root    /usr/bin/zarafa-admin --purge-softdelete 30

# spamassassin updates
# choose the right channels for your language!
5       22    *       *       *       root    sa-update --nogpg --channel updates.spamassassin.org --channel
sought.rules.yerp.org --channel sa.zmi.at --channel spamassassin.heinlein-support.de && service spamassassin
restart
```

Testing / Verifying the setup

- Send Mails via
  - WebAccess / WebApp
  - pop/pops
  - imap/imaps
  - postscreen checks
  - clamav checks
  - spamassassin check

- Deliver mails
  - test local delivery to username / alias
  - test group deliveries
- exchange / activesync / mobile access

This page has been viewed Unknown macro: 'tracking-info' times.

Contact

Additions, comments, critisicm? mail to: b2c[at]dest-unreachable.net