Baruwa Enterprise Edition Documentation

Release 2.0.5

Andrew Colin Kissa

September 29, 2014

CONTENTS

Baruwa Enterprise Edition is a fully fledged Mail Security solution, based on best of breed open source software packages. It provides protection from spam, viruses, phishing attempts and malware.

Baruwa Enterprise Edition works with any standard SMTP server, is highly accurate, scalable, easy to integrate as well as manage.

Automated installation and configuration management tools are provided to ensure the efficient and easy management of the System.

CHAPTER

ONE

WHAT IS BARUWA ENTERPRISE EDITION

Baruwa Enterprise Edition is a fully fledged Mail Security solution, based on best of breed open source software packages. It provides protection from spam, viruses, phishing attempts and malware.

Baruwa Enterprise Edition works with any standard SMTP server, is highly accurate, scalable, easy to integrate as well as manage.

Automated installation and configuration management tools are provided to ensure the efficient and easy management of the System.

The management interface is implemented using web 2.0 features (AJAX) where deemed fit. It has full support for i18n, enabling you to translate it into any language of your choosing. It has already been translated into to over 25 languages. Current *Languages supported*

Also included is reporting functionality with an easy to use query builder, whose results can be displayed as message lists or graphed as colorful and pretty interactive graphs.

Built in Full text search functionality allows you to find information very fast and easily. Advanced searching options available in leading web search engines are supported.

Baruwa Enterprise Edition is built on an open source core.

1.1 How does it work

It operates as an Email security gateway accepting mail from untrusted sources, running extensive checks on it and then passing the clean mail to the destination. It does not support the hosting of user mailboxes.

For incoming messages, it is configured to accept mail on behalf of your internal mail server run extensive checks on it then forward the clean mail to your internal mail server.

For outgoing messages, your internal mail server can be configured to pass all outbound messages to it for processing before being sent on to the destination. From the internal servers point of view the system is its smart host.

It can operate as a standalone all in one solution or as a cluster of servers sharing database, indexing, storage and message queue servers. The traditional concept of a cluster master is not supported, all the nodes in the cluster have equal status and can be brought into and taken out of the cluster without any special changes.

1.2 Features

- Spam, Virus, Phishing, Malware protection

- Extensive Spam Detection checks
- AJAX support for most operations
- Ultra fast full text search
- Reporting with AJAX enabled query builder
- I18n support, allows use of multiple languages
- Themes/Skins for rebranding
- Signature management / Branding
- Mail queue management and reporting
- Message delivery/relay information
- DKIM management
- Reporting graphs
- Emailed PDF reports
- Audit trails
- Archiving of old message logs
- SQLite backup prevents data loss when DB is unavailable
- MTA integration
- Multi Tenancy
- User profile aware approved/banned sender management
- IP / network addresses supported in approved/banned list manager
- SQL based MailScanner configuration management
- System status information
- IPv6 Support
- Asynchronous MailScanner logging
- Import and Export of User accounts and Domains
- AD/Exchange integration to auto populate account and group information
- Easy plug-in authentication to external authentication systems
- AD/LDAP, POP3, IMAP, SMTP, RADIUS Authentication support
- Tools for housekeeping tasks
- Easy clustering of multiple servers
- Works both with and without Javascript enabled

1.3 System Requirements

- Intel/AMD 2.0 GHZ+ 64-bit CPU
- Minimum - 2 GB RAM
- 10 GB free disk space for software and logs (SATA or SCSI for performance, and RAID/Mirroring for redundancy)

- Additional disk space for mail storage
- Centos/SL/OL Operating systems

NOTE: The amount of resources allocated to system is directly related to the amount of email the system will be processing as well as the number of users connected to the web interface.

CHAPTER

TWO

INSTALLATION

2.1 Automated Installation Prerequisites

2.1.1 Prerequisites

Minimum and Recommended Hardware

The bare minimum system requirements for all in one system are:

- 2GB RAM
- Multicore Intel/AMD 64-bit CPU
- 10 GB

The recommended system requirements for all in one system are:

- 8GB RAM
- Multicore Intel/AMD 64-bit CPU
- 40 GB

Note: The amount of resources allocated to system is directly related to the amount of email the system will be processing as well as the number of users connected to the web interface. Please scope your system resources based on the projections of email and web traffic.

Partitioning scheme

Please partition the system to provide the bulk of disk space to the /var partition. It is advisable to have the /var partition on a standalone partition with a file system that does not limit the number of files such as EXT4 and XFS.

Note: There is no need to create a /home partition for this system, as no home directories will be created. The default partition scheme does create a /home partition with the largest allocation, you need to change that.

Operating system

Baruwa Enterprise Edition should be installed on a fresh minimal installation of one of the supported operating systems (Centos/SL/OL 6.x). The platform can either be platform either be virtual or physical. If you are installing on a virtual platform make sure that the system does return a valid UUID, you can check that by running the following command.:

```
dmidecode |grep UUID
```

The minimal installation profile ensures that no unrequired software is installed. Make sure you do not install Graphical interfaces such as KDE or GNOME as these introduce large numbers of dependencies and are not required for the functioning and operation of the system.

Note: Baruwa Enterprise Edition can be installed on RHEL systems you do need however to setup a spacewalk proxy system to be able to obtain software from our spacewalk repository while still getting updates from the Redhat network which uses a similar system.

Hostname

Make sure that you set a hostname for your server and that the hostname resolves to the correct IP address on the server. The hostname should set to an actual hostname not localhost.localdomain. Make sure that the hostname command also returns a FQDN, to test this run the following command on the command line.:

```
hostname
```

Note: If you do not set the correct name above you will have several issues with the system some of which include inability to preview mail and some system services not starting for example RabbitMQ.

Selinux

Baruwa Enterprise Edition does not ship with any Selinux policy modules, you can either disable selinux or use allow2module to generate a local Selinux policy.

Network Firewall

Baruwa Enterprise Edition requires the following ports open to allow for proper functioning.

PORT	PROTOCOL	DIRECTION	DESCRIPTION
25	TCP	INBOUND/OUTBOUND	SMTP TRAFFIC
476	TCP	INBOUND	TLS SMTP TRAFFIC
587	TCP	INBOUND	SMTP SUBMISSION
80	TCP	INBOUND/OUTBOUND	WEB TRAFFIC
443	TCP	INBOUND/OUTBOUND	WEB TRAFFIC
53	TCP/UDP	OUTBOUND	DNS TRAFFIC
123	UDP	OUTBOUND	NTP TRAFFIC
2703	TCP	OUTBOUND	RAZOR TRAFFIC
24441	TCP/UDP	OUTBOUND	PYZOR TRAFFIC
873	TCP/UDP	OUTBOUND	UPDATES TRAFFIC

Enterprise subscription

Baruwa Enterprise Edition is shipped from a subscription based repository and provides stable and tested packages for Centos/RHEL/SL. Access to this repository is obtained by purchasing a server subscription. If you do not already have a subscription for the server you want to install please refer to https://www.baruwa.com to obtain a subscription for the server before proceeding.

2.2 Automated Installation

2.2.1 Step 1: Installation requirements

You need a valid Baruwa enterprise subscription, which provides you with a server entitlement as well as an activation key to activate the entitlement.

Enable the EPEL repository

The EPEL repository is a volunteer-based community effort from the Fedora project to create a repository of high-quality add-on packages for Red Hat Enterprise (RHEL) and its compatible spinoffs such as CentOS, Oracle Enterprise Linux or Scientific Linux. You can find more details on EPEL including how to add it to your host at http://fedoraproject.org/wiki/EPEL and http://fedoraproject.org/wiki/EPEL/FAQ#howtouse.

You need to enable this repo in order to access required packages:

```
rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
```

Install Spacewalk client packages

Baruwa Enterprise Edition entitlements are managed by the Baruwa Network. The Baruwa Network uses the Spacewalk server to manage entitlements. In order to access the Baruwa Enterprise Edition repository you need to install the Spacewalk client tools. These tools are provided by the Spacewalk project via a yum repository which you need to enable:

```
rpm -Uvh http://yum.spacewalkproject.org/1.9/RHEL/6/x86_64/spacewalk-client-repo-1.9-1.el6.noarch.rpm
```

Having enabled the Spacewalk repository you can now install the Spacewalk client packages:

```
yum install rhn-client-tools rhn-check rhn-setup rhnsd m2crypto yum-rhn-plugin -y
```

Install Baruwa signing keys

The packages in the Baruwa Centos/RHEL/SL enterprise repository are cryptographically signed using GPG keys. The package containing these GPG keys needs to be manually installed before continuing to the next step:

```
rpm -Uvh https://www.baruwa.com/downloads/baruwa-enterprise-release-6-2.noarch.rpm
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-BARUWA-ENTERPRISE-6
```

Activate Entitlement

The Baruwa Centos/RHEL/SL enterprise repository is available to subscribers only. To install from this repo you need to activate the entitlement for the server that you are installing.

The server entitlement `activation key` is emailed to you when you purchase a subscription. Use the `activation key` to register your server with the Baruwa Network using the command below:

```
rhnreg_ks --serverUrl=https://bn.baruwa.com/XMLRPC --activationkey=<activation-key>
```

2.2.2 Step 2: Installation

Install any available system updates:

```
yum upgrade -y
```

Install puppet:

```
yum install puppet -y
```

Download and install the puppet toaster from the baruwa.com website:

```
curl -O https://www.baruwa.com/downloads/puppet-toaster-latest.tar.bz2
tar xjvf puppet-toaster-latest.tar.bz2 -C /etc/puppet/
```

Create a puppet host manifest for your host by copying the provided sample:

```
cp /etc/puppet/manifests/toasters/baruwa/init.pp \
/etc/puppet/manifests/toasters/baruwa/$(hostname).pp
chown root:root /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
chmod 0600 /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
```

Edit the manifest file and set the options to reflect the host you are installing.

Make sure you change the following options

Note: Don't use the `@` and `:` characters in the passwords or usernames

---

Option	Description
$pgsql_password	Postgresql admin password
$baruwa_admin_user	Baruwa admin username
$baruwa_admin_email	Baruwa admin user email
$baruwa_admin_passwd	Baruwa admin user password
$baruwa_pgsql_passwd	Baruwa Postgresql password
$baruwa_timezone	Server Timezone
$baruwa_session_secret	Session encryption key
$baruwa_app_uuid	Baruwa application UUID
$baruwa_rabbitmq_passwd	Baruwa RabbitMQ password
$baruwa_quarantine_host_url	Quarantine URL
$baruwa_web_vhost	Baruwa virtual host name
$baruwa_web_serveraliases	Baruwa server aliases
$baruwa_mail_host	Mail server hostname
$baruwa_bayes_pgsql_pass	Bayes Postgresql password
$baruwa_cluster_peers	Hostnames of other nodes that are in the cluster. *Must be hostnames not IP Addresses*
$baruwa_cluster_peer_ips	IP addresses of other nodes that are in the cluster *Must be IP addresses not hostnames*
$baruwa_cluster_id	The cluster ID of this node *Must be an integer*
$baruwa_quarantine_shared	Enables and disables shared quarantine features
$baruwa_theme_path	Sets the Themes directory
$baruwa_custom_name	Sets the custom product name for rebranding
$baruwa_custom_url	Sets the url for the product
$sphinx_enable_wildcard	Enable Sphinx wildcard indexing, enabling this will use more disk space
$baruwa_dkim_selector	Sets the DKIM selector name
$openssl_country_code	SSL Certificate country code
$openssl_ca_name	SSL CA name
$openssl_province_name	SSL Certificate province
$openssl_city_name	SSL city name
$openssl_org_name	SSL organization name

Review the other settings and set accordingly.

SSL Certificates

The Baruwa web interface should ran over SSL/TLS, other services such as SMTP AUTH only work over SSL/TLS as well. So you need to either purchase a valid SSL certificate or puppet will automatically generate one non recognised SSL certificate for you using the `openssl_` options you have configured in the manifest file. This certificate that is automatically generated uses the `hostname` of the server.

Note: We have partnered with the SSLShop to bring you discounted SSL certificate pricing. `RapidSSL CA` signed certificates can be purchased at discounted pricing using the Discount coupon "BARUWA" from http://www.sslshop.co.za

---

If you have a SSL certificate that is issued by a recognised CA and would like Baruwa to use it, install it prior to running puppet:

```
mkdir -p /etc/pki/baruwa/{certs,private}
```

Create the following files

- `/etc/pki/baruwa/certs/$(hostname).pem` with the contents of your SSL certificate

- `/etc/pki/baruwa/private/$(hostname).key` with the contents of your SSL private key

If your `hostname` is different from the name you would like to use to access the web interface, you need to create a certificate/key pair for that name. Replace `baruwa.example.com` with your web name. This web name should be the same as what you have set as `$baruwa_web_vhost` in the manifest file `/etc/puppet/manifests/toasters/baruwa/$(hostname).pp`.

- `/etc/pki/baruwa/private/baruwa.example.com.key`
- `/etc/pki/baruwa/certs/baruwa.example.com.pem`

If your `hostname` is different from the mail server hostname you would like to use, then you need to create a certificate/key pair for that in the following files. Replace `baruwa.example.com` with your mail server hostname. The mail server hostname should correspond with the setting `$baruwa_mail_host` in the manifest file `/etc/puppet/manifests/toasters/baruwa/$(hostname).pp`.

- `/etc/pki/baruwa/certs/baruwa.example.com.pem`
- `/etc/pki/baruwa/private/baruwa.example.com.key`

If you have a wildcard certificate with all your names being subdomains of that domain to which the certificate is issued then you can create symlinks to each of the names for the certificates and keys.

Run Puppet

Run puppet using the manifest file that you created. This will take some time while it sets up your server. When the command finishes you will have a fully working Baruwa installation:

```
puppet -v /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
```

Note: If any of the tasks fails, rerun the above command. If you still have failures after running the command multiple times, then contact *Support*.

2.2.3 Step 3: Finalize configuration

Now that the installation and setup are complete, you need to finalize the setup by *Adding a scanning Node*, *Add an Organization*, *Adding a Domain* and *Adding an Account*.

Review the *Administrators guide* for other configuration and setup options available.

2.2.4 Step 4: Advanced options

Baruwa Enterprise Edition supports clustering, addons, additional AV engines as well as customisation using themes. If you intend on using these features read the following topics.

- *Clustering*
- *Themes*
- *Addons*
- *Additional Commercial Anti Virus Engines*

2.2.5 Step 5: Getting help

Support and assistance are available to you, refer to *Support* for details on how to get help.

2.3 Manual Installation Prerequisites

Note: Manual installations are for experienced system administrators who would like to fully customize their installations and intimately understand the various software packages used. Please use the *Automated Installation Prerequisites* if in depth customization is not what you want or you are not conversant with all the packages used to create a fully functional Mail security system.

2.3.1 Prerequisites

Minimum and Recommended Hardware

The bare minimum system requirements for all in one system are:

- 2GB RAM
- Multicore Intel/AMD 64-bit CPU
- 10 GB

The recommended system requirements for all in one system are:

- 8GB RAM
- Multicore Intel/AMD 64-bit CPU
- 40 GB

Note: The amount of resources allocated to system is directly related to the amount of email the system will be processing as well as the number of users connected to the web interface. Please scope your system resources based on the projections of email and web traffic.

Partitioning scheme

Please partition the system to provide the bulk of disk space to the /var partition. It is advisable to have the /var partition on a standalone partition with a file system that does not limit the number of files such as EXT4 and XFS.

Note: There is no need to create a /home partition for this system, as no home directories will be created. The default partition scheme does create a /home partition with the largest allocation, you need to change that.

Operating system

Baruwa Enterprise Edition should be installed on a fresh minimal installation of one of the supported operating systems (Centos/SL/OL 6.x). The platform can either be platform either be virtual or physical. If you are installing on a virtual platform make sure that the system does return a valid UUID, you can check that by running the following command.:

```
dmidecode |grep UUID
```

The minimal installation profile ensures that no unrequired software is installed. Make sure you do not install Graphical interfaces such as KDE or GNOME as these introduce large numbers of dependencies and are not required for the functioning and operation of the system.

Note: Baruwa Enterprise Edition can be installed on RHEL systems you do need however to setup a spacewalk proxy system to be able to obtain software from our spacewalk repository while still getting updates from the Redhat network which uses a similar system.

Hostname

Make sure that you set a hostname for your server and that the hostname resolves to the correct IP address on the server. The hostname should set to an actual hostname not localhost.localdomain. Make sure that the hostname command also returns a FQDN, to test this run the following command on the command line.:

```
hostname
```

Note: If you do not set the correct name above you will have several issues with the system some of which include inability to preview mail and some system services not starting for example RabbitMQ.

Selinux

Baruwa Enterprise Edition does not ship with any Selinux policy modules, you can either disable selinux or use allow2module to generate a local Selinux policy.

Network Firewall

Baruwa Enterprise Edition requires the following ports open to allow for proper functioning.

PORT	PROTOCOL	DIRECTION	DESCRIPTION
25	TCP	INBOUND/OUTBOUND	SMTP TRAFFIC
476	TCP	INBOUND	TLS SMTP TRAFFIC
587	TCP	INBOUND	SMTP SUBMISSION
80	TCP	INBOUND/OUTBOUND	WEB TRAFFIC
443	TCP	INBOUND/OUTBOUND	WEB TRAFFIC
53	TCP/UDP	OUTBOUND	DNS TRAFFIC
123	UDP	OUTBOUND	NTP TRAFFIC
2703	TCP	OUTBOUND	RAZOR TRAFFIC
24441	TCP/UDP	OUTBOUND	PYZOR TRAFFIC
873	TCP/UDP	OUTBOUND	UPDATES TRAFFIC

Enterprise subscription

Baruwa Enterprise Edition is shipped from a subscription based repository and provides stable and tested packages for Centos/RHEL/SL. Access to this repository is obtained by purchasing a server subscription. If you do not already have a subscription for the server you want to install please refer to https://www.baruwa.com to obtain a subscription for the server before proceeding.

2.4 Manual Installation

2.4.1 Step 1: Installation requirements

You need a valid Baruwa enterprise subscription, which provides you with a server entitlement as well as an activation key to activate the entitlement.

Enable the EPEL repository

The EPEL repository is a volunteer-based community effort from the Fedora project to create a repository of high-quality add-on packages for Red Hat Enterprise (RHEL) and its compatible spinoffs such as CentOS, Oracle Enterprise Linux or Scientific Linux. You can find more details on EPEL including how to add it to your host at http://fedoraproject.org/wiki/EPEL and http://fedoraproject.org/wiki/EPEL/FAQ#howtouse.

You need to enable this repo in order to access required packages:

```
rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
```

Install Spacewalk client packages

Baruwa Enterprise Edition entitlements are managed by the Baruwa Network. The Baruwa Network uses the Spacewalk server to manage entitlements. In order to access the Baruwa Enterprise Edition repository you need to install the Spacewalk client tools. These tools are provided by the Spacewalk project via a yum repository which you need to enable:

```
rpm -Uvh http://yum.spacewalkproject.org/1.9/RHEL/6/x86_64/spacewalk-client-repo-1.9-1.el6.noarch.rpm
```

Having enabled the Spacewalk repository you can now install the Spacewalk client packages:

```
yum install rhn-client-tools rhn-check rhn-setup rhnsd m2crypto yum-rhn-plugin -y
```

Install Baruwa signing keys

The packages in the Baruwa Centos/RHEL/SL enterprise repository are cryptographically signed using GPG keys. The package containing these GPG keys needs to be manually installed before continuing to the next step:

```
rpm -Uvh https://www.baruwa.com/downloads/baruwa-enterprise-release-6-2.noarch.rpm
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-BARUWA-ENTERPRISE-6
```

Activate Entitlement

The Baruwa Centos/RHEL/SL enterprise repository is available to subscribers only. To install from this repo you need to activate the entitlement for the server that you are installing.

The server entitlement `activation key` is emailed to you when you purchase a subscription. Use the `activation key` to register your server with the Baruwa Network using the command below:

```
rhnreg_ks --serverUrl=https://bn.baruwa.com/XMLRPC --activationkey=<activation-key>
```

Install Caching DNS server

You need to use a local caching DNS server to improve the performance of various checks:

```
yum install bind
chkconfig --level 3 named on
service named start
```

Then make sure that the system uses this DNS server:

```
cat > /etc/resolv.conf << 'EOF'
nameserver 127.0.0.1
EOF
```

Note: Do NOT use public DNS servers such as Google DNS, OpenDNS or your ISP's servers if you do, your DNSBL checks will get the a _BLOCKED response http://wiki.apache.org/spamassassin/DnsBlocklists#dnsbl-block for more information.

2.4.2 Step 2: Upgrade all existing packages

Install any available system updates:

```
yum upgrade -y
```

2.4.3 Step 3: Install and configure supporting packages

Step 3a: PostgreSQL

This is the database backend used by Baruwa to store data. You only have to install the server if you are going to run the database on the system system as Baruwa:

```
yum install postgresql-server postgresql-plpython -y
```

We now need to set a password on the postgresql `postgres` admin account, we use the password `strongPgP4ss` change this for your environment.:

```
chown postgres.postgres /var/lib/pgsql
echo "strongPgP4ss" > /tmp/ps
su postgres -c "/usr/bin/initdb /var/lib/pgsql/data --auth='password' --pwfile=/tmp/ps -E UTF8"
rm -rf /tmp/ps
```

You now need to configure the authentication settings on your postgresql server, edit your `pg_hba.conf` file and change the entries to the following:

```
cat > /var/lib/pgsql/data/pg_hba.conf << 'EOF'
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
local   all         all                               md5
host    all         all         127.0.0.1/32          md5
host    all         all         ::1/128               md5
EOF
```

Configure the server to use the `UTC` timezone as the default timezone:

```
sed -e "s/^#timezone = \(.*\)$/timezone = 'UTC'/" -i /var/lib/pgsql/data/postgresql.conf
```

Restart the service for the configuration changes to take effect:

```
service postgresql restart
```

With the server now started you can proceed to configuration. Here we will create a Baruwa postgresql database user account as well as a database to store Baruwa data.

We're going to assume that the database is called `baruwa`, the postgresql user is called `baruwa`, and the password is `password`.

Create the Baruwa database user:

```
psql -Upostgres postgres -c "CREATE ROLE baruwa WITH LOGIN PASSWORD 'password';"
```

Create the database:

```
createdb -U postgres -E UTF8 -O baruwa -T template1 baruwa
```

Baruwa uses functions written in the `plpgsql` and `plpythonu` procedural languages. Enable these languages in the db:

```
psql -U postgres baruwa -c "CREATE LANGUAGE plpgsql;"
psql -U postgres baruwa -c "CREATE LANGUAGE plpythonu;"
```

We're going to assume that the Bayes user is called `bayes`, and the password is `password`.

Create the Bayes database user:

```
psql -Upostgres postgres -c "CREATE ROLE bayes WITH LOGIN PASSWORD 'password';"
```

Create the SQL Auto Whitelisting Tables:

```
cat > /tmp/awl.sql << 'EOF'
CREATE TABLE awl (
  username varchar(100) NOT NULL default '',
  email varchar(255) NOT NULL default '',
  ip varchar(40) NOT NULL default '',
  count bigint NOT NULL default '0',
  totscore float NOT NULL default '0',
  signedby varchar(255) NOT NULL default '',
  timestamp timestamp with time zone default timezone('utc'::text, now()),
  PRIMARY KEY (username,email,signedby,ip)
);
SQL
psql -Ubaruwa baruwa -f /tmp/awl.sql
rm -f /tmp/awl.sql
```

Create the SQL Bayes tables, using the *Bayes SQL dump* provided, Paste the contents into `/tmp/bayes.sql` then run the following commands:

```
psql -Ubaruwa baruwa -f /tmp/bayes.sql
rm -f /tmp/bayes.sql
```

Grant the `bayes` user access to the `bayes` tables:

```
cat > /tmp/grants.sql << 'EOF'
GRANT SELECT, UPDATE, DELETE, INSERT ON TABLE bayes_token TO bayes;
GRANT SELECT, UPDATE, DELETE, INSERT ON TABLE bayes_vars TO bayes;
GRANT SELECT, DELETE, INSERT ON TABLE bayes_seen TO bayes;
GRANT SELECT, DELETE, INSERT ON TABLE bayes_expire TO bayes;
GRANT SELECT ON TABLE bayes_global_vars TO bayes;
GRANT UPDATE, SELECT, INSERT ON bayes_vars_id_seq TO bayes;
```

```
GRANT SELECT, UPDATE, DELETE, INSERT ON TABLE awl TO bayes;
EOF
psql -Upostgres baruwa -f /tmp/grants.sql
rm -f /tmp/grants.sql
```

Step 3b: RabbitMQ

The RabbitMQ server is used as the message broker to handle the processing on backend tasks such as releasing messages, reading queues and providing host status information.

Run the following commands to install and start RabbitMQ on your system.:

```
yum install rabbitmq-server -y
service rabbitmq-server start
```

Now create a virtual host and a RabbitMQ user to be used by Baruwa.

We're going to assume that the virtual host is called `baruwa`, the RabbitMQ user is called `baruwa`, and the password is `mysecretpwd`.

Create the user account, the virtual host and give the user permissions on the virtual host:

```
rabbitmqctl add_user baruwa mysecretpwd
rabbitmqctl add_vhost baruwa
rabbitmqctl set_permissions -p baruwa baruwa ".*" ".*" ".*"
```

Remove the guest user:

```
rabbitmqctl delete_user guest
```

Step 3c: Sphinx

The Sphinx search server provides fast indexed search results to queries submitted via Baruwa.

Run the following commands to install and start sphinx on your system:

```
yum install sphinx
```

Create a `/etc/sphinx/sphinx.conf` using the provided sample *sphinx.conf*

Set the required database settings:

```
sed -i -e 's:sql_host =:sql_host = 127.0.0.1:' \
        -e 's:sql_user =:sql_user = baruwa:' \
        -e 's:sql_pass =:sql_pass = password:' \
        -e 's:sql_db =:sql_db = baruwa:' /etc/sphinx/sphinx.conf
```

Start the Sphinx server:

```
service searchd restart
chkconfig --level 3 searchd on
```

Step 3d: Memcached

Memcached is used to cache data and alleviate the load on the database backend as well as store sessions:

```
yum install memcached -y
service memcached start
chkconfig --level 3 memcached on
```

Step 3e: MailScanner

MailScanner is the integrated engine that performs the various checks used to identify and classify spam and various threats.

Baruwa manages the MailScanner configuration by storing the configurations in the PostgreSQL Database. MailScanner signatures can also be managed using Baruwa for both domains and individual users.

Install MailScanner:

```
yum install mailscanner clamd clamav clamav-unofficial-sigs -y
```

Create the following configuration files based on the samples provided.

- `/etc/MailScanner/MailScanner.conf` - *MailScanner.conf*
- `/etc/exim/exim.conf` - *exim.conf*
- `/etc/exim/exim_out.conf` - *exim_out.conf*
- `/etc/exim/macros.conf` - *macros.conf*
- `/etc/exim4/trusted_configs` - *trusted_configs*
- `/etc/MailScanner/spam.assassin.prefs.conf` - *spam.assassin.prefs.conf*
- `/etc/cron.hourly/baruwa-clean-eximdb` - *baruwa-clean-eximdb*,
- `/etc/MailScanner/rules/filename.rules` - *filename.rules*,
- `/etc/MailScanner/rules/filetype.rules` - *filetype.rules*,
- `/etc/MailScanner/filetype.rules.allowall.conf` - *filetype.rules.allowall.conf*,
- `/etc/MailScanner/filename.rules.allowall.conf` - *filename.rules.allowall.conf*,
- `/etc/MailScanner/rules/content.scanning.rules` - *content.scanning.rules*
- `/etc/MailScanner/rules/scan.messages.rules` - *scan.messages.rules*
- `/etc/MailScanner/rules/nonspam.actions.rules` - *nonspam.actions.rules*
- `/etc/mail/spamassassin/baruwa.cf` - *baruwa.cf*
- `/etc/mail/spamassassin/sem.cf` - *sem.cf*
- `/etc/clamd.conf` - *clamd.conf*
- `/etc/freshclam.conf` - *freshclam.conf*

Add the ClamAV user to the exim group:

```
usermod -G exim clamav
```

Create the exim spool directories and settings files:

```
touch /etc/exim4/{non-tls-hosts,remove-headers,blocked-subjects,skip_dnsbl,skip_dkim,skip_av_checks,a
mkdir -p /var/spool/exim4.in/{db,input,msglog,scan}
chown -R exim.exim /var/spool/exim4.in
chmod 0750 -R /var/spool/exim4.in
```

Create the file command wrapper `/usr/local/bin/file-wrapper`:

```
cat > /usr/local/bin/file-wrapper << 'EOF'
#!/bin/bash
#
# Wrap the file command
/usr/bin/file -i "$1"
EOF
chmod +x /usr/local/bin/file-wrapper
```

Update the Spamassassin local configuration:

```
rm -f /etc/mail/spamassassin/local.cf
ln -s /etc/MailScanner/spam.assassin.prefs.conf /etc/mail/spamassassin/local.cf
```

Install extra Spamassassin plugins:

```
yum install spamassassin-plugin-iXhash spamassassin-plugin-decodeshorturls -y
```

Create the KAM cronjob, using the *KAM.cron* provided, Paste the contents into `/etc/cron.daily/kam` and then make it executable:

```
chmod +x /etc/cron.daily/kam
```

Enable Spamassassin rulesets updates:

```
sa-update --import /etc/mail/spamassassin/channel.d/sought.conf
sa-update --import /etc/mail/spamassassin/channel.d/spamassassin-official.conf
cat > /etc/sysconfig/update_spamassassin << 'EOF'
SAUPDATEARGS="-D --gpgkey 6C6191E3 --channel sought.rules.yerp.org --channel updates.spamassassin.org
EOF
```

Enable required Spamassassin plugins:

```
yum install perl-IP-Country re2c gcc make -y
v310file = "/etc/mail/spamassassin/v310.pre"
v320file = "/etc/mail/spamassassin/v320.pre"
initfile = "/etc/mail/spamassassin/init.pre"
sed -i -e "s/^#loadplugin Mail::SpamAssassin::Plugin::AWL/loadplugin Mail::SpamAssassin::Plugin::AWL/
sed -i -e "s/^# loadplugin Mail::SpamAssassin::Plugin::Rule2XSBody/loadplugin Mail::SpamAssassin::Plu
sed -i -e "s/^# loadplugin Mail::SpamAssassin::Plugin::Shortcircuit/loadplugin Mail::SpamAssassin::Pl
sed -i -e "s/# loadplugin Mail::SpamAssassin::Plugin::RelayCountry/loadplugin Mail::SpamAssassin::Plu
```

Install Spamassassin addons:

```
yum install perl-Razor-Agent pyzor dcc-client
mkdir /var/lib/razor
razor-admin -home=/var/lib/razor -logfile=/var/log/razor-agent.log -create
razor-admin -home=/var/lib/razor -logfile=/var/log/razor-agent.log -register
echo "razorhome = /var/lib/razor" >> /var/lib/razor/razor-agent.conf && chown exim.exim -R /var/lib/r
mkdir /var/lib/pyzor
chown root.mail /var/lib/pyzor
pyzor --homedir /var/lib/pyzor discover
chown exim /var/lib/pyzor/servers
v310file = "/etc/mail/spamassassin/v310.pre"
sed -i -e "s/^#loadplugin Mail::SpamAssassin::Plugin::DCC/loadplugin Mail::SpamAssassin::Plugin::DCC/
```

Restart the relevant services:

```
service clamd restart
service mailscanner restart
```

Make sure the services start on boot:

```
chkconfig --level 3 clamd on
chkconfig --level 3 exim off
chkconfig --level 3 dcc-client on
chkconfig --level 3 mailscanner on
```

Step 3f: SSL certificates

Note: We have partnered with the SSLShop to bring you discounted SSL certificate pricing. RapidSSL CA signed certificates can be purchased at discounted pricing using the Discount coupon "BARUWA" from http://www.sslshop.co.za

---

The Baruwa web interface should ran over SSL/TLS, other services such as SMTP AUTH only work over SSL/TLS as well. So you need to either purchase a valid SSL certificate or generate a self signed certificate. If you have an SSL certificate that is issued by a recognised CA and install it as follows:

```
mkdir -p /etc/pki/baruwa/{certs,private}
```

Create the following files

- `/etc/pki/baruwa/certs/$(hostname).pem` with the contents of your SSL certificate
- `/etc/pki/baruwa/private/$(hostname).key` with the contents of your SSL private key

If your `hostname` is different from the name you use to access the web site, you need to create a certificate/key pair for that. Replace `baruwa.example.com` with your web hostname.

- `/etc/pki/baruwa/private/baruwa.example.com.key`
- `/etc/pki/baruwa/certs/baruwa.example.com.pem`

If your `hostname` is different from the the mail server hostname, then you need to create a certificate/key pair for that in the following files. Replace `baruwa.example.com` with your mail server hostname.

- `/etc/pki/baruwa/certs/baruwa.example.com.pem`
- `/etc/pki/baruwa/private/baruwa.example.com.key`

If you have a wildcard certificate with all your names being subdomains of that domain to which the certificate is issued then you can create symlinks to each of the names for the certificates and keys.

If you do not have a CA signed signed certificate you need to generate a self signed certificate and place the private key in `/etc/pki/baruwa/private/$(hostname).key` and the certificate in `/etc/pki/baruwa/certs/$(hostname).pem`. Create an additional certificate/key pairs if your `web access name`/`mail server hostname` are different from the server `hostname`.

Step 3g: Nginx

Nginx is the web server available in Baruwa Enterprise. Install it by running:

```
yum install nginx -y
```

Create the Baruwa Nginx configuration file `/etc/nginx/conf.d/baruwa.conf` based on the provided sample *nginx.conf*.

Start the Nginx service:

```
service nginx restart
chkconfig --level 3 nginx on
```

2.4.4 Step 4: Setup Baruwa

Step 4a: Install Baruwa

With all the requirements in place you can now install Baruwa by running the following command:

```
yum install baruwa -y
```

If you are running sphinx on the same server then install the integration package:

```
yum install baruwa-sphinx -y
```

Step 4b: Create configuration files

Create the configuration file:

```
paster make-config baruwa /etc/baruwa/production.ini
```

Set the sqlalchemy database url:

```
sed -i -e 's|baruwa:@127.0.0.1:5432/baruwa|baruwa:password@127.0.0.1:5432/baruwa|' \
        /etc/baruwa/production.ini
```

Set the broker password and enable the queues:

```
sed -i -e 's:broker.password =:broker.password = mysecretpwd:' \
        -e "s:snowy.local:$(hostname):g" \
        -e 's:^#celery.queues:celery.queues:'/etc/baruwa/production.ini
```

Check the configuration file and ensure that the `baruwa.timezone` option matches the timezone configured on your server. Take time to review the other options to ensure that they are correct for your setup.

Note: Don't use the `@` and `:` characters in the passwords or usernames

---

Step 4c: Populate the database

Creation of functions written in `plpythonu` requires PostgreSQL admin user access. So we create them in this step using the `postgres` admin account:

```
psql -U postgres baruwa -f /usr/lib/python2.6/site-packages/baruwa/config/sql/admin-functions.sql
```

The creation of all database tables, addition of initial data and the creation of an admin user is taken care of via this Pylons command:

```
paster setup-app /etc/baruwa/production.ini
```

Step 4d: Create the sphinx indexes

The initial sphinx search indexes need to be created by running the command:

```
indexer --all --rotate
```

Step 4e: Start the celery daemon

Start the celeryd daemon:

```
service baruwa start
chkconfig --level 3 baruwa on
```

Step 4f: Link uwsgi configuration

Link the Baruwa configuration to the uwsgi configuration directory:

```
ln -s /etc/baruwa/production.ini /etc/uwsgi
service uwsgi restart
chkconfig --level 3 uwsgi on
```

Step 4g: Create Sudo configuration

Create a sudo file in /etc/sudoers.d as follows:

```
cat > /etc/sudoers.d/baruwa << 'EOF'
Defaults:baruwa   !requiretty, visiblepw

baruwa ALL=(exim) NOPASSWD: /usr/sbin/exim -C /etc/exim/exim_out.conf -M *, \
        /usr/sbin/exim -C /etc/exim/exim_out.conf -Mf *, \
        /usr/sbin/exim -C /etc/exim/exim_out.conf -Mrm *, \
        /usr/sbin/exim -C /etc/exim/exim_out.conf -Mg *, \
        /usr/sbin/exim -C /etc/exim/exim_out.conf -Mar *, \
        /usr/sbin/exim -C /etc/exim/exim_out.conf -qff, \
                /usr/sbin/exim -Mrm *, \
                /usr/sbin/exim -Mg *, \
                /usr/sbin/exim -Mar *

baruwa ALL = NOPASSWD: /bin/kill -s HUP *
EOF
```

Step 4h: Create Cronjobs

Create the bayes expire cronjob:

```
cat > /etc/cron.hourly/baruwa-expire-bayes < 'EOF'
#!/bin/bash
/usr/bin/sa-learn --force-expire --sync -p /etc/MailScanner/spam.assassin.prefs.conf &>/dev/null
EOF
chmod +x /etc/cron.hourly/baruwa-expire-bayes
```

Create the exim queue cleanup cronjob:

```
cat > /etc/cron.daily/baruwa-exim4 << 'EOF'
#!/bin/bash
#
# Baruwa Enterprise Edition
# Clean up stale messages from the exim queues
#
# PERIOD in seconds
PERIOD=1209600
```

```
exiqgrep -C /etc/exim/exim_out.conf -o $PERIOD -i|xargs exim -C /etc/exim/exim_out.conf -Mrm >/dev/nu
exiqgrep -o $PERIOD -i|xargs exim -Mrm >/dev/null 2>&1
EOF
```

2.4.5 Step 5: Finalize configuration

Now that the installation and setup are complete, you need to finalize the setup by *Adding a scanning Node*, *Add an Organization*, *Adding a Domain* and *Adding an Account*.

Review the *Administrators guide* for other configuration and setup options available.

2.4.6 Step 7: Advanced options

Baruwa Enterprise Edition supports clustering, addons, additional AV engines as well as customisation using themes. If you intend on using these features read the following topics.

- *Clustering*
- *Themes*
- *Addons*
- *Additional Commercial Anti Virus Engines*

2.4.7 Step 8: Getting help

Support and assistance are available to you, refer to *Support* for details on how to get help.

2.5 Sample configuration files

2.5.1 clamd.conf

```
LogFile /var/log/clamav/clamd.log
LogFileMaxSize 4M
LogTime yes
LogSyslog yes
LogFacility LOG_MAIL
PidFile /var/run/clamav/clamd.pid
TemporaryDirectory /var/tmp
DatabaseDirectory /var/lib/clamav
LocalSocket /var/run/clamav/clamd.sock
FixStaleSocket yes
TCPSocket 3310
TCPAddr 127.0.0.1
MaxConnectionQueueLength 60
StreamMaxLength 20M
MaxThreads 50
ReadTimeout 300
User clam
AllowSupplementaryGroups yes
ScanPE yes
ScanELF yes
DetectBrokenExecutables yes
```

```
ScanOLE2 yes
ScanMail yes
StructuredDataDetection no
ScanArchive yes
ArchiveBlockEncrypted no
StructuredMinCreditCardCount 5
```

2.5.2 freshclam.conf

```
DatabaseDirectory /var/lib/clamav
UpdateLogFile /var/log/clamav/freshclam.log
LogFileMaxSize 5M
LogTime yes
LogSyslog yes
DatabaseOwner clam
DatabaseMirror db.us.clamav.net
DatabaseMirror db.local.clamav.net
DatabaseMirror database.clamav.net
DatabaseMirror db.za.clamav.net
DatabaseCustomURL http://www.mailscanner.eu/scamnailer.ndb
Checks 48
SafeBrowsing yes
Bytecode yes
```

2.5.3 exim.conf

```
.include /etc/exim/macros.conf
hide pgsql_servers = PGSQL_SERVERS
primary_hostname = ms.home.topdog-software.com
domainlist local_domains = @ : localhost : localhost.localdomain
domainlist relay_sql_domains = RELAY_SQL_DOMAINS
domainlist relay_sql_smtp_domains = SMTP_SQL_DOMAINS
domainlist relay_sql_lmtp_domains = LMTP_SQL_DOMAINS
domainlist ldap_domains = LDAP_DOMAINS
domainlist smtp_callback_domains = SMTP_CALLBACK_DOMAINS
domainlist whitelisted_domains = WHITELISTED_DOMAINS
domainlist blacklisted_domains = BLACKLISTED_DOMAINS
addresslist whitelisted_addresses = WHITELISTED_ADDRESS
addresslist blacklisted_addresses = BLACKLISTED_ADDRESS
hostlist whitelisted_hosts = WHITELISTED_HOSTS
hostlist blacklisted_hosts = BLACKLISTED_HOSTS
hostlist relay_sql_hosts = RELAY_SQL_HOSTS
hostlist relay_from_hosts = localhost : localhost.localdomain
acl_smtp_rcpt = acl_check_rcpt
acl_smtp_data = acl_check_data
acl_smtp_mime = acl_check_mime
acl_smtp_connect = acl_check_connect
acl_smtp_helo   = acl_check_helo
acl_smtp_dkim = acl_check_dkim
smtp_banner = Baruwa 2.0 $tod_full
smtp_active_hostname = ${if !eq{$sender_host_address}{$received_ip_address}{${lookup dnsdb{>: ptr=$re
smtp_accept_max_per_connection = 60
smtp_accept_max = 0
smtp_load_reserve = 15
smtp_receive_timeout = 3m
```

```
smtp_accept_max_nonmail = 10
smtp_max_unknown_commands = 1
message_size_limit = 20M
spool_directory = /var/spool/exim.in
pipelining_advertise_hosts = 127.0.0.1
process_log_path = /var/spool/exim/exim-process.info
received_header_text = Received: ${if def:sender_rcvhost {from $sender_rcvhost\n\t}{${if def:sender_
av_scanner = clamd:/var/run/clamav/clamd.sock
tls_advertise_hosts = ${lookup{$sender_host_address}lsearch{/etc/exim/non-tls-hosts}{}{*}}
tls_certificate = /etc/pki/baruwa/certs/${primary_hostname}.pem
tls_privatekey = /etc/pki/baruwa/private/${primary_hostname}.key
tls_on_connect_ports = 465
tls_require_ciphers = GNUTLS_CIPHERS
daemon_smtp_ports = 25 : 465 : 587
never_users = root
rfc1413_hosts = *
rfc1413_query_timeout = 0s
ignore_bounce_errors_after = 1s
timeout_frozen_after = 1s
auth_advertise_hosts = ${if eq {$tls_cipher}{}{}{*}}
perl_startup = do '/usr/share/baruwa/exim-bcrypt.pl'
perl_at_start = true
begin acl
acl_check_rcpt:
  accept  hosts          = :
          control        = submission
  drop    message        = The sender $sender_address is banned
          hosts          = +blacklisted_hosts
  drop    message        = The domain $sender_address_domain is banned
          sender_domains = +blacklisted_domains
  drop    message        = Dictionary attack detected
          condition      = ${if >{$rcpt_fail_count}{3} {yes}{no}}
          delay          = 10m
  drop    message        = Legitimate bounces are never sent to more than one recipient.
          senders        = : postmaster@*
          condition      = ${if >{$recipients_count}{1}{true}{false}}
  drop    message        = Restricted characters in address
          domains        = +local_domains
          local_parts    = ^[.] : ^.*[@%!/|]
  drop    message        = Restricted characters in address
          domains        = !+local_domains
          local_parts    = ^[./|] : ^.*[@%!] : ^.*/\\.\\./
  accept  local_parts    = postmaster
          domains        = +local_domains : +relay_sql_domains
  drop    message        = sender verification failed
          !hosts         = 127.0.0.1
          !verify        = sender
  drop    message        = recipient verification failed
          !verify        = recipient
  accept  hosts          = +relay_from_hosts : +relay_sql_hosts
          control        = submission/sender_retain
  accept  authenticated  = *
          control        = submission/sender_retain
          set acl_m_u    = $authenticated_id
          add_header     = X-Authenticated-As: $acl_m_u
  require message        = relay not permitted
          domains        = +local_domains : +relay_sql_domains
  drop    message        = The email address does not exist
```

```
          domains        = +smtp_callback_domains
          !verify        = recipient/success_on_redirect/callout=2m,defer_ok
  drop    message        = The email address does not exist
          domains        = +ldap_domains
          condition      = ${lookup ldap{${expand:LDAP_LOOKUP}}{0}{1}}
  accept  dnslists       = wl.rbl.baruwa.net : list.dnswl.org&0.0.0.3 : hostkarma.junkemailfilter.cor
          add_header     = X-Baruwa-DNSL-Whitelisted-Host: $sender_host_address
          add_header     = X-Baruwa-DNSL-Name: ${if eq{$dnslist_domain}{}{list.dnswl.org}{$dnslist_do
          logwrite       = ${if eq{$dnslist_domain}{wl.rbl.baruwa.net}\
                           {The sender $sender_host_address is an approved sender}\
                           {The sender $sender_host_address is in a DNS whitelist at \
                           ${if eq{$dnslist_domain}{}{list.dnswl.org}{$dnslist_domain}}}}
  accept  senders        = +whitelisted_addresses
          add_header     = X-Baruwa-Whitelisted-Sender: $sender_address
          logwrite       = The sender address $sender_address is whitelisted
  accept  sender_domains = +whitelisted_domains
          add_header     = X-Baruwa-Whitelisted-Domain: $sender_address_domain
          logwrite       = The sending domain $sender_address_domain is whitelisted
  accept  hosts          = +whitelisted_hosts
          add_header     = X-Baruwa-Whitelisted-Host: $sender_host_address
          logwrite       = The sending host $sender_host_address is whitelisted
  accept  condition      = ${lookup{$sender_host_address}iplsearch{/etc/exim/skip_dnsbl}{1}{0}}
  drop    message        = The sender $dnslist_text
          dnslists       = rbl.baruwa.net=127.0.0.2 : rbl.baruwa.net=127.0.0.2/$sender_address_domain
  drop    message        = The sender $sender_host_address is in a black list http://www.spamhaus.org
          dnslists       = zen.spamhaus.org
          ratelimit      = 0 / 2h / strict / per_conn
  drop    message        = The sender $sender_host_address is in a black list at $dnslist_domain\n$dn
          dnslists       = bl.spamcop.net : cbl.abuseat.org
          ratelimit      = 0 / 2h / strict / per_conn
  drop    message        = We don't accept messages from hosts without reverse DNS records
          log_message    = The sender $sender_host_address has no reverse DNS record
          !verify        = reverse_host_lookup
          !verify        = sender/no_details/callout=2m,defer_ok
          !condition     = ${if eq{$sender_verify_failure}{}}
  deny    message        = SPF_MSG
          spf            = fail
  accept
acl_check_data:
  accept  malware       = *
          hosts         = 127.0.0.1
          condition     = ${if match \
                          {${malware_name}} \
                          {\N(\.UNOFFICIAL)$\N} \
                          {1}{0}}
          add_header    = X-Baruwa-Quarantine-Report-Bypass: ${malware_name}
  drop    malware       = *
          message       = This message contains a virus ($malware_name).
  drop    message       = This message is administratively prohibited
          hosts         = ! +relay_sql_hosts
          !authenticated  = *
          condition     = ${if and {{def:h_Reply-to:}{eq {$h_Reply-to:}{}}}{yes}{no}}
          condition     = ${lookup{$sender_host_address}iplsearch{/etc/exim/allow_empty_replyto}{0}{1
  accept
acl_check_mime:
  drop    message       = Blacklisted file extension detected
          condition     = ${if match \
                          {${lc:$mime_filename}} \
```

```
                          {\N(\.exe|\.pif|\.bat|\.scr|\.lnk|\.com)$\N} \
                          {1}{0}}
  accept
acl_check_connect:
  accept  hosts         = :
  drop    message       = The sender $sender_host_address is banned
          hosts         = +blacklisted_hosts
  accept  message       = The sending host $sender_host_address is whitelisted
          hosts         = +whitelisted_hosts
  defer   ratelimit     = 250 / 15m / strict
          message       = You can only send $sender_rate_limit msgs per $sender_rate_period
          log_message   = RATE: $sender_rate/$sender_rate_period (max $sender_rate_limit)
  accept
acl_check_helo:
  drop  message         = The sender did not present a HELO/EHLO greeting
          log_message   = remote host did not present greeting
          condition     = ${if def:sender_helo_name {false}{true}}
  drop  message         = The sender presented HELO as an IP address (See RFC2821 4.1.3)
          condition     = ${if isip{$sender_helo_name}}
  accept
acl_check_dkim:
  accept authenticated  = *
  accept hosts          = :
  accept hosts          = +whitelisted_hosts
  accept condition      = ${lookup{$sender_host_address}iplsearch{/etc/exim/skip_dkim}{1}{0}}
  deny message          = DKIM failure: $dkim_verify_reason
       dkim_status      = none:invalid
       condition        = ${if eq {$dkim_key_testing}{1} {no}{yes}}
  warn add_header       = X-DKIM: Status on $received_ip_address using Baruwa 2.0: dkim=$dkim_verify_
                          signing_identity="$dkim_cur_signer"
  accept
begin routers
split:
   driver = accept
   domains = +relay_sql_domains
   condition = ${if and {{!eq {$received_protocol}{split}}{gt {$recipients_count}{1}}}{yes}{no}}
   transport = send_to_self
   no_verify
   no_address_test
message_checks:
   driver = redirect
   allow_defer
   data = :defer: queued for message checks
   no_verify
   no_address_test
deliver_clean_smtp:
   driver = manualroute
   domains = +relay_sql_smtp_domains
   transport = remote_smtp
   route_data = ${lookup pgsql {ROUTE_QUERY}}
   no_more
deliver_clean_lmtp:
   driver = manualroute
   domains = +relay_sql_lmtp_domains
   transport = remote_lmtp
   route_data = ${lookup pgsql {ROUTE_QUERY}}
   no_more
dnslookup:
```

```
   driver = dnslookup
   domains = ! +local_domains : ! +relay_sql_domains
   transport = remote_smtp
   ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
   no_more
system_aliases:
   driver = redirect
   allow_fail
   allow_defer
   domains = @
   data = ${lookup{$local_part}lsearch{/etc/aliases}}
   file_transport = address_file
   pipe_transport = address_pipe
localuser:
   driver = accept
   check_local_user
   transport = local_delivery
   cannot_route_message = Unknown user
begin transports
send_to_self:
   driver = pipe
   batch_max = 1
   use_bsmtp
   command = /usr/sbin/exim -oMr split -bS
   user = exim
remote_smtp:
   driver = smtp
   delay_after_cutoff = false
remote_lmtp:
   driver = smtp
   protocol = lmtp
   delay_after_cutoff = false
   port = 25
local_delivery:
   driver = appendfile
   file = /var/mail/$local_part
   delivery_date_add
   envelope_to_add
   return_path_add
   group = mail
   mode = 0660
address_pipe:
   driver = pipe
   return_output
address_file:
   driver = appendfile
   delivery_date_add
   envelope_to_add
   return_path_add
begin retry
*                      *           F,2h,15m; G,16h,1h,1.5; F,14d,6h
begin rewrite
begin authenticators
PLAIN:
   driver = plaintext
   server_prompts = :
   server_condition = ${if and{ {!eq {$auth2}{}} {!eq {$auth3}{}}\
                                {bool{${perl{check_password}\
```

```
                                {${lookup pgsql {ORG_CHECK_PLAIN}{$value}}}\
                                {$auth3}}}\
                                }\
                              }\
                        {yes}{no}}
   server_set_id = $2
   server_advertise_condition = ${if def:tls_cipher }

LOGIN:
   driver = plaintext
   server_prompts = "Username:: : Password::"
   server_condition = ${if and{ {!eq {$auth1}{}} {!eq {$auth2}{}}\
                                    {bool{${perl{check_password}\
                                    {${lookup pgsql {ORG_CHECK_LOGIN}{$value}}}\
                                    {$auth2}}}}\
                                  }\
                           {yes}{no}}
   server_set_id = $1
   server_advertise_condition = ${if def:tls_cipher }
```

2.5.4 exim_out.conf

```
.include /etc/exim/macros.conf
hide pgsql_servers = PGSQL_SERVERS
primary_hostname = ms.home.topdog-software.com
domainlist local_domains = @ : localhost : localhost.localdomain
domainlist relay_sql_rand_smtp = SMTP_RAND_DOMAINS
domainlist relay_sql_nonrand_smtp = SMTP_NONRAND_DOMAINS
domainlist relay_sql_rand_lmtp = LMTP_RAND_DOMAINS
domainlist relay_sql_nonrand_lmtp = LMTP_NONRAND_DOMAINS
domainlist relay_sql_domains = RELAY_SQL_DOMAINS
hostlist   relay_from_hosts =
acl_smtp_rcpt = acl_check_rcpt
acl_smtp_data = acl_check_data
acl_smtp_mime = acl_check_mime
acl_smtp_connect = acl_check_connect
acl_smtp_helo   = acl_check_helo
smtp_banner = Baruwa 2.0 $tod_full
#disable_ipv6 = true
smtp_load_reserve = 10
tls_advertise_hosts = *
tls_certificate = /etc/pki/baruwa/certs/${primary_hostname}.pem
tls_privatekey = /etc/pki/baruwa/private/${primary_hostname}.key
tls_require_ciphers = GNUTLS_CIPHERS
daemon_smtp_ports = 25
#log_file_path=:syslog
#syslog_duplication=false
#syslog_timestamp=false
never_users = root
rfc1413_hosts = *
rfc1413_query_timeout = 0s
ignore_bounce_errors_after = 1s
timeout_frozen_after = 1s
auth_advertise_hosts =
begin acl
acl_check_rcpt:
```

```
  accept
acl_check_data:
  accept
acl_check_mime:
  accept
acl_check_connect:
  accept
acl_check_helo:
  accept
begin routers
deliver_clean_randomize:
   driver = manualroute
   domains = +relay_sql_rand_smtp
   transport = remote_smtp
   hosts_randomize = true
   route_data = ${lookup pgsql {ROUTE_QUERY}}
deliver_clean_norandomized:
   driver = manualroute
   domains = +relay_sql_nonrand_smtp
   transport = remote_smtp
   hosts_randomize = false
   route_data = ${lookup pgsql {ROUTE_QUERY}}
deliver_clean_randomize_lmtp:
   driver = manualroute
   domains = +relay_sql_rand_lmtp
   transport = remote_lmtp
   hosts_randomize = true
   route_data = ${lookup pgsql {ROUTE_QUERY}}
deliver_clean_norandomized_lmtp:
   driver = manualroute
   domains = +relay_sql_nonrand_lmtp
   transport = remote_lmtp
   hosts_randomize = false
   route_data = ${lookup pgsql {ROUTE_QUERY}}
dnslookup:
   driver = dnslookup
   domains = ! +local_domains : ! +relay_sql_domains
   transport = remote_smtp
   ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
   no_more
system_aliases:
   driver = redirect
   allow_fail
   allow_defer
   data = ${lookup{$local_part}lsearch{/etc/aliases}}
   file_transport = address_file
   pipe_transport = address_pipe
localuser:
   driver = accept
   check_local_user
   transport = local_delivery
   cannot_route_message = Unknown user
begin transports
remote_smtp:
   driver = smtp
   tpda_host_defer_action = DEFER_QUERY
   tpda_delivery_action = DELIVERY_QUERY
   delay_after_cutoff = false
```

```
   dkim_domain = ${if exists{/etc/MailScanner/baruwa/dkim/${lc:$sender_address_domain}.pem}\
                 {${lc:$sender_address_domain}}{}}
   dkim_selector = baruwa
   dkim_private_key = ${if exists{/etc/MailScanner/baruwa/dkim/${lc:$sender_address_domain}.pem}\
                       {/etc/MailScanner/baruwa/dkim/${lc:$sender_address_domain}.pem}{0}}
   tls_require_ciphers = GNUTLS_CIPHERS
remote_lmtp:
   driver = smtp
   protocol = lmtp
   port = 25
   tpda_host_defer_action = DEFER_QUERY
   tpda_delivery_action = DELIVERY_QUERY
   delay_after_cutoff = false
   dkim_domain = ${if exists{/etc/MailScanner/baruwa/dkim/${lc:$sender_address_domain}.pem}\
                 {${lc:$sender_address_domain}}{}}
   dkim_selector = baruwa
   dkim_private_key = ${if exists{/etc/MailScanner/baruwa/dkim/${lc:$sender_address_domain}.pem}\
                       {/etc/MailScanner/baruwa/dkim/${lc:$sender_address_domain}.pem}{0}}
   tls_require_ciphers = GNUTLS_CIPHERS
local_delivery:
   driver = appendfile
   file = /var/mail/$local_part
   delivery_date_add
   envelope_to_add
   return_path_add
   group = mail
   mode = 0660
address_pipe:
   driver = pipe
   return_output
address_file:
   driver = appendfile
   delivery_date_add
   envelope_to_add
   return_path_add
begin retry
*                      *           F,2h,15m; G,16h,1h,1.5; F,14d,6h
begin rewrite
begin authenticators
```

2.5.5 macros.conf

```
ROUTE_QUERY = SELECT '"<+ ' || array_to_string(ARRAY(SELECT address FROM routedata WHERE enabled='t'
SMTP_NONRAND_DOMAINS = pgsql;SELECT name FROM mtasettings WHERE name='${quote_pgsql:$domain}' \
                            AND delivery_mode=2 AND protocol=1;
SMTP_RAND_DOMAINS = pgsql;SELECT name FROM mtasettings WHERE name='${quote_pgsql:$domain}' \
                            AND delivery_mode=1 AND protocol=1;
LMTP_NONRAND_DOMAINS = pgsql;SELECT name FROM mtasettings WHERE name='${quote_pgsql:$domain}' \
                            AND protocol=2 AND delivery_mode=2;
LMTP_RAND_DOMAINS = pgsql;SELECT name FROM mtasettings WHERE name='${quote_pgsql:$domain}' \
                           AND protocol=2 AND delivery_mode=1;
SMTP_SQL_DOMAINS = pgsql;SELECT name FROM mtasettings WHERE name='${quote_pgsql:$domain}' \
                          AND protocol=1;
LMTP_SQL_DOMAINS = pgsql;SELECT name FROM mtasettings WHERE name='${quote_pgsql:$domain}' \
                          AND protocol=2;
LDAP_DOMAINS = pgsql;SELECT name FROM mtasettings WHERE name='${quote_pgsql:$domain}' \
```

```
                    AND ldap_callout='t';
SMTP_CALLBACK_DOMAINS = pgsql;SELECT name FROM mtasettings where name='${quote_pgsql:$domain}' \
                            AND smtp_callout='t';
RELAY_SQL_DOMAINS = pgsql;SELECT name FROM relaydomains WHERE name='${quote_pgsql:$domain}';
WHITELISTED_DOMAINS = pgsql;SELECT from_address FROM lists WHERE to_address='any' AND list_type=1 AND
BLACKLISTED_DOMAINS = pgsql;SELECT from_address FROM lists WHERE to_address='any' AND list_type=2 AND
WHITELISTED_ADDRESS = pgsql;SELECT from_address FROM lists WHERE to_address='any' AND list_type=1 AND
BLACKLISTED_ADDRESS = pgsql;SELECT from_address from lists WHERE to_address='any' AND list_type=2 AND
WHITELISTED_HOSTS = pgsql;SELECT from_address FROM lists WHERE to_address='any' AND list_type=1 AND
BLACKLISTED_HOSTS = pgsql;SELECT from_address FROM lists WHERE to_address='any' AND list_type=2 AND
RELAY_SQL_HOSTS = pgsql;SELECT address FROM relaysettings WHERE enabled='t' AND address='${quote_pgsq

PGSQL_SERVERS = (/tmp/.s.PGSQL.5432)/baruwa/baruwa/password

SPF_MSG = Please see http://www.openspf.org/Why?scope=${if def:sender_address_domain {mfrom}{helo}};
LDAP_LOOKUP = ${lookup pgsql {SELECT url FROM ldaplookup, ldapmaps WHERE ldaplookup.name=ldapmaps.par
DELIVERY_QUERY = ${lookup pgsql {INSERT INTO messagestatus (messageid, hostname, ipaddress, port, cor
                                 VALUES('${quote_pgsql:$message_exim_id}', '${quote_pgsql:$tpda_delive
                                 ${quote_pgsql:$tpda_delivery_port}, '${quote_pgsql:$tpda_delivery_cor
DEFER_QUERY = ${lookup pgsql {INSERT INTO messagestatus (messageid, hostname, ipaddress, port, confir
                              VALUES('${quote_pgsql:$message_exim_id}', '${quote_pgsql:$tpda_delivery_
                              ${quote_pgsql:$tpda_delivery_port}, '${quote_pgsql:$tpda_delivery_confir
                              '${quote_pgsql:$tpda_defer_errstr}')}}
DKIM_STATUS = ${lookup pgsql {SELECT name FROM maildomains,dkim_keys WHERE maildomains.id = dkim_keys
PASSWD_CHECK_LOGIN = SELECT password FROM users WHERE username='${quote_pgsql:$auth1}'
PASSWD_CHECK_PLAIN = SELECT password FROM users WHERE username='${quote_pgsql:$auth2}'
ORG_CHECK_LOGIN = SELECT password FROM relaysettings WHERE username='${quote_pgsql:$auth1}'
ORG_CHECK_PLAIN = SELECT password FROM relaysettings WHERE username='${quote_pgsql:$auth2}'

OPENSSL_CIPHERS = NONE:+VERS-TLS-ALL:+ARCFOUR-128:+RSA:+SHA256:+SHA1:+COMP-NULL
GNUTLS_CIPHERS = TLSv1.2+HIGH : TLSv1+HIGH : !SSLv2 : RC4+MEDIUM : !aNULL : !eNULL : !3DES : !MD5 :
```

2.5.6 trusted_configs

```
/etc/exim/exim.conf
/etc/exim/exim_out.conf
```

2.5.7 baruwa-clean-eximdb

```
#!/bin/bash
#
[ -x /usr/sbin/exim_tidydb ] || exit 0
/usr/sbin/exim_tidydb -t 1m /var/spool/exim.in retry &>/dev/null
exit 0
```

2.5.8 baruwa.cf

```
# Filter urls
urirhsbl                URIBL_BARUWA    uribl.baruwa.net.   A
body                    URIBL_BARUWA    eval:check_uridnsbl('URIBL_BARUWA')
describe                URIBL_BARUWA    Contains a URL listed in the Baruwa blocklist
score                   URIBL_BARUWA    15.0
tflags                  URIBL_BARUWA    net
```

```
ifplugin Mail::SpamAssassin::Plugin::Shortcircuit
  priority URIBL_BARUWA -5001
  shortcircuit URIBL_BARUWA on
endif

# BaruwaWL
header          __RCVD_IN_BARUWAWL      eval:check_rbl('BARUWAWL-lastexternal','wl.rbl.baruwa.net.')
tflags          __RCVD_IN_BARUWAWL      net nice
describe        __RCVD_IN_BARUWAWL      Sender in Baruwa Approved senders list

# HostKarma
header          __RCVD_IN_HOSTKARMA     eval:check_rbl('HOSTKARMA-lastexternal','hostkarma.junkemaili
describe        __RCVD_IN_HOSTKARMA     Sender listed in JunkEmailFilter
tflags          __RCVD_IN_HOSTKARMA     net

header          __RCVD_IN_HOSTKARMA_W   eval:check_rbl_sub('HOSTKARMA-lastexternal', '127.0.0.1')
describe        __RCVD_IN_HOSTKARMA_W   Sender listed in HOSTKARMA-WHITE
tflags          __RCVD_IN_HOSTKARMA_W   net nice

header          RCVD_IN_HOSTKARMA_BR    eval:check_rbl_sub('HOSTKARMA-lastexternal', '127.0.0.4')
describe        RCVD_IN_HOSTKARMA_BR    Sender listed in HOSTKARMA-BROWN
tflags          RCVD_IN_HOSTKARMA_BR    net
score           RCVD_IN_HOSTKARMA_BR    1.0

# Meta rules (BARUWAWL & HOSTKARMA)
meta            RCVD_IN_BARUWAWL        (__RCVD_IN_BARUWAWL && !__RCVD_IN_HOSTKARMA_W)
describe        RCVD_IN_BARUWAWL        Sender in Baruwa Approved senders list
score           RCVD_IN_BARUWAWL        -5.0

meta            RCVD_IN_HOSTKARMA_W     (!__RCVD_IN_BARUWAWL && __RCVD_IN_HOSTKARMA_W)
describe        RCVD_IN_HOSTKARMA_W     Sender listed in HOSTKARMA-WHITE
score           RCVD_IN_HOSTKARMA_W     -5.0

meta            RCVD_IN_BW_HKW          (__RCVD_IN_BARUWAWL && __RCVD_IN_HOSTKARMA_W)
describe        RCVD_IN_BW_HKW          Sender listed in HOSTKARMA-WHITE and BARUWAWL
score           RCVD_IN_BW_HKW          -5.0
```

2.5.9 KAM.cron

```
#!/bin/bash
# Insert a random delay up to this value, to spread virus updates round
# the clock. 1800 seconds = 30 minutes.
# Set this to 0 to disable it.
UPDATEMAXDELAY=600
if [ -f /etc/sysconfig/MailScanner ] ; then
        . /etc/sysconfig/MailScanner
fi
export UPDATEMAXDELAY

if [ "x$UPDATEMAXDELAY" = "x0" ]; then
  :
else
  logger -p mail.info -t KAM.cf.sh Delaying cron job up to $UPDATEMAXDELAY seconds
  perl -e "sleep int(rand($UPDATEMAXDELAY));"
fi
```

```
# JKF Fetch KAM.cf
echo Fetching KAM.cf...
reload=1
cd /etc/mail/spamassassin
#rm -f KAM.cf
#/usr/bin/wget -O KAM.cf http://www.peregrinehw.com/downloads/SpamAssassin/contrib/KAM.cf
/usr/bin/wget -N http://www.peregrinehw.com/downloads/SpamAssassin/contrib/KAM.cf
if [ "$?" = "0" ]; then
  echo It completed okay.
  if [ -r KAM.cf.backup ]; then
    if [ KAM.cf -nt KAM.cf.backup ]; then
      if ( tail -10 KAM.cf | grep -qE '^#.*(EOF|END)' ); then
        echo It succeeded, so make a backup
        cp -f KAM.cf KAM.cf.backup
      else
        echo ERROR: Could not find EOF marker in KAM.cf
        cp -f KAM.cf.backup KAM.cf
      fi
    else
      echo Remote file not newer than local copy
      reload=0
    fi
  else
    # No backup file present, so delete file if it is bad
    if ( tail -10 KAM.cf | grep -qE '^#.*(EOF|END)' ); then
      echo Success, make a backup
      cp -f KAM.cf KAM.cf.backup
    else
      echo ERROR: Could not find EOF marker in KAM.cf and no backup
      rm -f KAM.cf
      reload=0
    fi
  fi
else
  echo It failed to complete properly
  if [ -r KAM.cf.backup ]; then
    echo Restored backup of KAM.cf
    cp -f KAM.cf.backup KAM.cf
  else
    # No backup copy present, so delete bad KAM.cf
    echo ERROR: wget of KAM.cf failed and no backup
    rm -f KAM.cf
    reload=0
  fi
fi

# Reload MailScanner only if we need to.
if [ "$reload" = "1" ]; then
  echo Reloading MailScanner and SpamAssassin configuration rules
  /etc/init.d/mailscanner reload
fi
```

2.5.10 Bayes SQL dump

```
CREATE TABLE bayes_expire (
  id integer NOT NULL default '0',
  runtime integer NOT NULL default '0'
```

```
) WITHOUT OIDS;

CREATE INDEX bayes_expire_idx1 ON bayes_expire (id);

CREATE TABLE bayes_global_vars (
  variable varchar(30) NOT NULL default '',
  value varchar(200) NOT NULL default '',
  PRIMARY KEY  (variable)
) WITHOUT OIDS;

INSERT INTO bayes_global_vars VALUES ('VERSION','3');

CREATE TABLE bayes_seen (
  id integer NOT NULL default '0',
  msgid varchar(200) NOT NULL default '',
  flag character(1) NOT NULL default '',
  PRIMARY KEY  (id,msgid)
) WITHOUT OIDS;

CREATE TABLE bayes_token (
  id integer NOT NULL default '0',
  token bytea NOT NULL default '',
  spam_count integer NOT NULL default '0',
  ham_count integer NOT NULL default '0',
  atime integer NOT NULL default '0',
  PRIMARY KEY  (id,token)
) WITHOUT OIDS;

CREATE INDEX bayes_token_idx1 ON bayes_token (token);

CREATE TABLE bayes_vars (
  id serial NOT NULL,
  username varchar(200) NOT NULL default '',
  spam_count integer NOT NULL default '0',
  ham_count integer NOT NULL default '0',
  token_count integer NOT NULL default '0',
  last_expire integer NOT NULL default '0',
  last_atime_delta integer NOT NULL default '0',
  last_expire_reduce integer NOT NULL default '0',
  oldest_token_age integer NOT NULL default '2147483647',
  newest_token_age integer NOT NULL default '0',
  PRIMARY KEY  (id)
) WITHOUT OIDS;

CREATE UNIQUE INDEX bayes_vars_idx1 ON bayes_vars (username);

CREATE OR REPLACE FUNCTION greatest_int (integer, integer)
 RETURNS INTEGER
 IMMUTABLE STRICT
 AS 'SELECT CASE WHEN $1 < $2 THEN $2 ELSE $1 END;'
 LANGUAGE SQL;

CREATE OR REPLACE FUNCTION least_int (integer, integer)
 RETURNS INTEGER
 IMMUTABLE STRICT
 AS 'SELECT CASE WHEN $1 < $2 THEN $1 ELSE $2 END;'
 LANGUAGE SQL;
```

```
CREATE OR REPLACE FUNCTION put_tokens(INTEGER,
                                      BYTEA[],
                                      INTEGER,
                                      INTEGER,
                                      INTEGER)
RETURNS VOID AS 'DECLARE
  inuserid      ALIAS FOR $1;
  intokenary    ALIAS FOR $2;
  inspam_count  ALIAS FOR $3;
  inham_count   ALIAS FOR $4;
  inatime       ALIAS FOR $5;
  _token BYTEA;
  new_tokens INTEGER := 0;
BEGIN
  for i in array_lower(intokenary, 1) .. array_upper(intokenary, 1)
  LOOP
    _token := intokenary[i];
    UPDATE bayes_token
       SET spam_count = greatest_int(spam_count + inspam_count, 0),
           ham_count = greatest_int(ham_count + inham_count, 0),
           atime = greatest_int(atime, inatime)
     WHERE id = inuserid
       AND token = _token;
    IF NOT FOUND THEN
      -- we do not insert negative counts, just return true
      IF NOT (inspam_count < 0 OR inham_count < 0) THEN
        INSERT INTO bayes_token (id, token, spam_count, ham_count, atime)
        VALUES (inuserid, _token, inspam_count, inham_count, inatime);
        IF FOUND THEN
          new_tokens := new_tokens + 1;
        END IF;
      END IF;
    END IF;
  END LOOP;

  IF new_tokens > 0 AND inatime > 0 THEN
    UPDATE bayes_vars
       SET token_count = token_count + new_tokens,
           newest_token_age = greatest_int(newest_token_age, inatime),
           oldest_token_age = least_int(oldest_token_age, inatime)
     WHERE id = inuserid;
  ELSIF new_tokens > 0 AND NOT inatime > 0 THEN
    UPDATE bayes_vars
       SET token_count = token_count + new_tokens
     WHERE id = inuserid;
  ELSIF NOT new_tokens > 0 AND inatime > 0 THEN
    UPDATE bayes_vars
       SET newest_token_age = greatest_int(newest_token_age, inatime),
           oldest_token_age = least_int(oldest_token_age, inatime)
     WHERE id = inuserid;
  END IF;
  RETURN;
END;' LANGUAGE 'plpgsql';
```

2.5.11 sem.cf

```
# SEM-URI
urirhssub SEM_URI uribl.spameatingmonkey.net. A 2
body SEM_URI eval:check_uridnsbl('SEM_URI')
describe SEM_URI Contains a URI listed by SEM-URI
tflags SEM_URI net
score SEM_URI 5.0
```

2.5.12 spam.assassin.prefs.conf

```
auto_whitelist_distinguish_signed 1

#Bayes
bayes_auto_expire 1
bayes_store_module Mail::SpamAssassin::BayesStore::PgSQL
bayes_sql_dsn DBI:Pg:dbname=baruwa;host=localhost;port=5432
bayes_sql_override_username bayes
bayes_sql_username bayes
bayes_sql_password password

#AWL
auto_whitelist_factory Mail::SpamAssassin::SQLBasedAddrList
user_awl_dsn           DBI:Pg:dbname=baruwa;host=localhost;port=5432
user_awl_sql_username  bayes
user_awl_sql_password  password
user_awl_sql_table     awl

dns_available yes
bayes_ignore_header X-YOURORG-BaruwaFW
bayes_ignore_header X-YOURORG-BaruwaFW-SpamCheck
bayes_ignore_header X-YOURORG-BaruwaFW-SpamScore
bayes_ignore_header X-YOURORG-BaruwaFW-Information
lock_method flock

dcc_home /var/lib/dcc
use_pyzor 1
pyzor_options --homedir /var/lib/pyzor
use_razor2 1
razor_config /var/lib/razor/razor-agent.conf
envelope_sender_header X-BaruwaFW-From

rbl_timeout 30
razor_timeout 60
pyzor_timeout 60

meta BARUWA_PDF_411 (SUBJ_ALL_CAPS && T_FREEMAIL_DOC_PDF && FREEMAIL_REPLYTO)
meta BARUWA_PDF_ALT_411 (T_FREEMAIL_DOC_PDF && FREEMAIL_REPLYTO)

score BAYES_00 -2.0
score BAYES_05 0.1
score BAYES_20 0.3
score BAYES_40 0.5
score BAYES_50 1.5
score BAYES_60 2.5
score BAYES_80 4.0
score BAYES_95 10.0
```

```
score BAYES_99 15.0
score BAYES_999 20.0
score SPF_HELO_PASS -2.0
score RAZOR2_CF_RANGE_51_100 4.0
score RAZOR2_CF_RANGE_E8_51_100 4.9
score RAZOR2_CF_RANGE_E4_51_100 4.5
score RAZOR2_CHECK 3.5
score PYZOR_CHECK 3.5
score DNS_FROM_OPENWHOIS 0.0
score RCVD_IN_DSBL 0.0
score RCVD_IN_SORBS_WEB 5.0
score RCVD_IN_SORBS_DUL 2.01
score DKIM_VALID_AU -4.0
score RCVD_BAD_ID 1.0
score SUBJ_ALL_CAPS 1.0
score URIBL_BLACK 10.0
score URIBL_JP_SURBL 5.0
score URIBL_WS_SURBL 5.0
score RCVD_IN_BL_SPAMCOP_NET 3.0
score TVD_SPACE_RATIO 1.5
score URIBL_DBL_SPAM 5.0
score RCVD_IN_RP_RNBL 5.0
score RCVD_IN_PBL 5.0
score SHORT_URIBL 10.0
score BARUWA_PDF_411 15.0
score BARUWA_PDF_ALT_411 15.0
score KAM_MXURI 0.5
score RCVD_BAD_ID 0.1
score KAM_COUK 0.5

#ok_languages en
```

2.5.13 MailScanner.conf

```
%org-name% = BARUWA
%org-long-name% = BARUWA MAILFW
%web-site% = example.com
%etc-dir% = /etc/MailScanner
%reports-base% = /etc/MailScanner/reports
%report-dir% = /etc/MailScanner/reports/en
%rules-dir% = /etc/MailScanner/rules
%mcp-dir% = /etc/MailScanner/mcp
%spool-dir% = /var/spool/MailScanner
%signature-dir% = /etc/MailScanner/baruwa/signatures
Max Children = 4
Run As User = exim
Run As Group = exim
Queue Scan Interval = 6
Incoming Queue Dir = /var/spool/exim.in/input
Outgoing Queue Dir = /var/spool/exim/input
Incoming Work Dir = %spool-dir%/incoming
Quarantine Dir = %spool-dir%/quarantine
PID file = /var/run/MailScanner/MailScanner.pid
Restart Every = 7200
MTA = exim
Sendmail = /usr/sbin/exim -C /etc/exim/exim_out.conf
```

```
Sendmail2 = /usr/sbin/exim -C /etc/exim/exim_out.conf
Incoming Work User = exim
Incoming Work Group = clam
Incoming Work Permissions = 0640
Quarantine User = exim
Quarantine Group = baruwa
Quarantine Permissions = 0660
Max Unscanned Bytes Per Scan = 100m
Max Unsafe Bytes Per Scan = 50m
Max Unscanned Messages Per Scan = 30
Max Unsafe Messages Per Scan = 30
Max Normal Queue Size = 800
Scan Messages = %rules-dir%/scan.messages.rules
Reject Message = no
Maximum Processing Attempts = 6
Processing Attempts Database = %spool-dir%/incoming/Processing.db
Maximum Attachments Per Message = 200
Expand TNEF = yes
Use TNEF Contents = replace
Deliver Unparsable TNEF = no
TNEF Expander = /usr/bin/tnef --maxsize=100000000
TNEF Timeout = 120
File Command = /usr/local/bin/file-wrapper
File Timeout = 20
Gunzip Command = /bin/gunzip
Gunzip Timeout = 50
Unrar Command = /usr/bin/unrar
Unrar Timeout = 50
Find UU-Encoded Files = no
Maximum Message Size = messagesize.customize
Maximum Attachment Size = -1
Minimum Attachment Size = -1
Maximum Archive Depth = 4
Find Archives By Content = yes
Unpack Microsoft Documents = yes
Zip Attachments = no
Attachments Zip Filename = MessageAttachments.zip
Attachments Min Total Size To Zip = 100k
Attachment Extensions Not To Zip = .zip .rar .gz .tgz .jpg .jpeg .mpg .mpe .mpeg .mp3 .rpm .htm .html
Add Text Of Doc = no
Antiword = /usr/bin/antiword -f
Antiword Timeout = 50
Unzip Maximum Files Per Archive = 0
Unzip Maximum File Size = 50k
Unzip Filenames = *.txt *.ini *.log *.csv
Unzip MimeType = text/plain
Virus Scanning = virusscan.customize
Virus Scanners = none
Virus Scanner Timeout = 300
Deliver Disinfected Files = no
Silent Viruses = HTML-IFrame All-Viruses
Still Deliver Silent Viruses = no
Non-Forging Viruses = Joke/ OF97/ WM97/ W97M/ eicar
Spam-Virus Header = X-%org-name%-BaruwaFW-SpamVirus-Report:
Virus Names Which Are Spam = Sane*UNOFFICIAL HTML/*
Block Encrypted Messages = no
Block Unencrypted Messages = no
Allow Password-Protected Archives = yes
```

```
Check Filenames In Password-Protected Archives = yes
Allowed Sophos Error Messages =
Sophos IDE Dir = /opt/sophos-av/lib/sav
Sophos Lib Dir = /opt/sophos-av/lib
Monitors For Sophos Updates = /opt/sophos-av/lib/sav/*.ide
Monitors for ClamAV Updates = /var/lib/clamav/*.cvd
ClamAVmodule Maximum Recursion Level = 8
ClamAVmodule Maximum Files = 1000
ClamAVmodule Maximum File Size = 10000000
ClamAVmodule Maximum Compression Ratio = 250
Clamd Port = 3310
Clamd Socket = /var/run/clamav/clamd.sock
Clamd Lock File =
Clamd Use Threads = no
ClamAV Full Message Scan = yes
Fpscand Port = 10200
Dangerous Content Scanning = contentchecks.customize
Allow Partial Messages = no
Allow External Message Bodies = no
Find Phishing Fraud = yes
Also Find Numeric Phishing = yes
Use Stricter Phishing Net = yes
Highlight Phishing Fraud = yes
Phishing Safe Sites File = %etc-dir%/phishing.safe.sites.conf
Phishing Bad Sites File = %etc-dir%/phishing.bad.sites.conf
Country Sub-Domains List = %etc-dir%/country.domains.conf
Allow IFrame Tags = disarm
Allow Form Tags = disarm
Allow Script Tags = disarm
Allow WebBugs = yes
Ignored Web Bug Filenames = spacer pixel.gif pixel.png gap
Known Web Bug Servers = msgtag.com
Web Bug Replacement = https://datafeeds.baruwa.com/1x1spacer.gif
Allow Object Codebase Tags = yes
Convert Dangerous HTML To Text = no
Convert HTML To Text = no
Archives Are = zip rar ole
Allow Filenames =
Deny Filenames =
Filename Rules = %rules-dir%/filename.rules
Allow Filetypes =
Allow File MIME Types =
Deny Filetypes =
Deny File MIME Types =
Filetype Rules = %rules-dir%/filetype.rules
Archives: Allow Filenames =
Archives: Deny Filenames =
Archives: Filename Rules = %etc-dir%/archives.filename.rules.conf
Archives: Allow Filetypes =
Archives: Allow File MIME Types =
Archives: Deny Filetypes =
Archives: Deny File MIME Types =
Archives: Filetype Rules = %etc-dir%/archives.filetype.rules.conf
Default Rename Pattern = __FILENAME__.disarmed
Quarantine Infections = yes
Quarantine Silent Viruses = no
Quarantine Modified Body = no
Quarantine Whole Message = yes
```

```
Quarantine Whole Messages As Queue Files = no
Keep Spam And MCP Archive Clean = yes
Language Strings = languages.customize
Rejection Report = rejectionreport.customize
Deleted Bad Content Message Report = deletedcontentmessage.customize
Deleted Bad Filename Message Report = deletedfilenamemessage.customize
Deleted Virus Message Report = deletedvirusmessage.customize
Deleted Size Message Report = deletedsizemessage.customize
Stored Bad Content Message Report = storedcontentmessage.customize
Stored Bad Filename Message Report = storedfilenamemessage.customize
Stored Virus Message Report = storedvirusmessage.customize
Stored Size Message Report = storedsizemessage.customize
Disinfected Report = disinfectedreport.customize
Inline HTML Signature = htmlsigs.customize
Inline Text Signature = textsigs.customize
Signature Image Filename = sigimgfiles.customize
Signature Image <img> Filename = sigimgs.customize
Inline HTML Warning = inlinewarninghtml.customize
Inline Text Warning = inlinewarningtxt.customize
Sender Content Report = sendercontentreport.customize
Sender Error Report = senderrorreport.customize
Sender Bad Filename Report = senderfilenamereport.customize
Sender Virus Report = sendervirusreport.customize
Sender Size Report = sendersizereport.customize
Hide Incoming Work Dir = yes
Include Scanner Name In Reports = yes
Mail Header = X-%org-name%-BaruwaFW:
Spam Header = X-%org-name%-BaruwaFW-SpamCheck:
Spam Score Header = X-%org-name%-BaruwaFW-SpamScore:
Information Header = X-%org-name%-BaruwaFW-Information:
Add Envelope From Header = yes
Add Envelope To Header = no
Envelope From Header = X-BaruwaFW-From:
Envelope To Header = X-%org-name%-BaruwaFW-To:
ID Header = X-%org-name%-BaruwaFW-ID:
IP Protocol Version Header =
Spam Score Character = s
SpamScore Number Instead Of Stars = no
Minimum Stars If On Spam List = 0
Clean Header Value = Found to be clean
Infected Header Value = Found to be infected
Disinfected Header Value = Disinfected
Information Header Value = Please contact %org-long-name% for more information
Detailed Spam Report = yes
Include Scores In SpamAssassin Report = yes
Always Include SpamAssassin Report = no
Multiple Headers = add
Place New Headers At Top Of Message = yes
Hostname = the %org-name% ($HOSTNAME) Baruwa
Sign Messages Already Processed = no
Sign Clean Messages = signmsgs.customize
Attach Image To Signature = yes
Attach Image To HTML Message Only = yes
Allow Multiple HTML Signatures = no
Dont Sign HTML If Headers Exist =
Mark Infected Messages = yes
Mark Unscanned Messages = no
Unscanned Header Value = Not scanned: please contact %org-long-name% for details
```

```
Remove These Headers = X-Mozilla-Status: X-Mozilla-Status2:
Deliver Cleaned Messages = yes
Notify Senders = no
Notify Senders Of Viruses = no
Notify Senders Of Blocked Filenames Or Filetypes = no
Notify Senders Of Blocked Size Attachments = no
Notify Senders Of Other Blocked Content = no
Never Notify Senders Of Precedence = list bulk
Scanned Modify Subject = no
Scanned Subject Text = {Scanned}
Virus Modify Subject = no
Virus Subject Text = {Virus?}
Filename Modify Subject = no
Filename Subject Text = {Filename?}
Content Modify Subject = no
Content Subject Text = {Dangerous Content?}
Size Modify Subject = no
Size Subject Text = {Size}
Disarmed Modify Subject = no
Disarmed Subject Text = {Disarmed}
Phishing Modify Subject = yes
Phishing Subject Text = {Suspected Phishing?}
Spam Modify Subject = no
Spam Subject Text = {Spam?}
High Scoring Spam Modify Subject = no
High Scoring Spam Subject Text = {Spam?}
Warning Is Attachment = yes
Attachment Warning Filename = %org-name%-Attachment-Warning.txt
Attachment Encoding Charset = ISO-8859-1
Archive Mail =
Missing Mail Archive Is = file
Send Notices = no
Notices Include Full Headers = yes
Hide Incoming Work Dir in Notices = yes
Notice Signature = -- \n%org-name%\nEmail Security\n%website%'
Notices From = Baruwa
Notices To = postmaster
Local Postmaster = postmaster
Spam List Definitions = %etc-dir%/spam.lists.conf
Virus Scanner Definitions = %etc-dir%/virus.scanners.conf
Spam Checks = spamchecks.customize
Spam List =
Spam Domain List =
Spam Lists To Be Spam = 1
Spam Lists To Reach High Score = 3
Spam List Timeout = 10
Max Spam List Timeouts = 7
Spam List Timeouts History = 10
Is Definitely Not Spam = approvedlist.customize
Is Definitely Spam = bannedlist.customize
Definite Spam Is High Scoring = yes
Ignore Spam Whitelist If Recipients Exceed = 20
Max Spam Check Size = 1000k
Use Watermarking = no
Add Watermark = yes
Check Watermarks With No Sender = yes
Treat Invalid Watermarks With No Sender as Spam = nothing
Check Watermarks To Skip Spam Checks = yes
```

```
Watermark Secret = %org-name%-BaruwaFW-Secret
Watermark Lifetime = 604800
Watermark Header = X-%org-name%-BaruwaFW-Watermark:
Use SpamAssassin = yes
Max SpamAssassin Size = 800k
Required SpamAssassin Score = spamscore.customize
High SpamAssassin Score = highspamscore.customize
SpamAssassin Auto Whitelist = yes
SpamAssassin Timeout = 75
Max SpamAssassin Timeouts = 10
SpamAssassin Timeouts History = 30
Check SpamAssassin If On Spam List = yes
Include Binary Attachments In SpamAssassin = no
Spam Score = yes
Cache SpamAssassin Results = yes
SpamAssassin Cache Database File = %spool-dir%/incoming/SpamAssassin.cache.db
Rebuild Bayes Every = 0
Wait During Bayes Rebuild = no
Use Custom Spam Scanner = no
Max Custom Spam Scanner Size = 20k
Custom Spam Scanner Timeout = 20
Max Custom Spam Scanner Timeouts = 10
Custom Spam Scanner Timeout History = 20
Spam Actions = spamactions.customize
High Scoring Spam Actions = highspamactions.customize
Non Spam Actions = %rules-dir%/nonspam.actions.rules
SpamAssassin Rule Actions =
Sender Spam Report = senderspamreport.customize
Sender Spam List Report = senderspamrblreport.customize
Sender SpamAssassin Report = senderspamsareport.customize
Inline Spam Warning = inlinespamwarning.customize
Recipient Spam Report = recipientspamreport.customize
Enable Spam Bounce = %rules-dir%/bounce.rules
Bounce Spam As Attachment = no
Syslog Facility = mail
Log Speed = no
Log Spam = no
Log Non Spam = no
Log Delivery And Non-Delivery = no
Log Permitted Filenames = no
Log Permitted Filetypes = no
Log Permitted File MIME Types = no
Log Silent Viruses = no
Log Dangerous HTML Tags = no
Log SpamAssassin Rule Actions = no
SpamAssassin Temporary Dir = /var/spool/MailScanner/incoming/SpamAssassin-Temp
SpamAssassin User State Dir =
SpamAssassin Install Prefix =
SpamAssassin Site Rules Dir = /etc/mail/spamassassin
SpamAssassin Local Rules Dir =
SpamAssassin Local State Dir =
SpamAssassin Default Rules Dir =
DB DSN = DBI:Pg:database=baruwa
DB Username = baruwa
DB Password = password
SQL Serial Number = SELECT MAX(value) AS confserialnumber FROM configurations WHERE internal='confser
SQL Quick Peek = SELECT dbvalue(value) AS value FROM quickpeek WHERE external = ? AND (hostname = ? O
SQL Config = SELECT internal, dbvalue(value) AS value, hostname FROM quickpeek WHERE hostname=? OR ho
```

```
SQL Ruleset = SELECT row_number, ruleset AS rule FROM msrulesets WHERE name=?
SQL SpamAssassin Config =
SQL Debug = no
Sphinx Host = 127.0.0.1
Sphinx Port = 9306
MCP Checks = no
First Check = spam
MCP Required SpamAssassin Score = 1
MCP High SpamAssassin Score = 10
MCP Error Score = 1
MCP Header = X-%org-name%-BaruwaFW-MCPCheck:
Non MCP Actions = deliver
MCP Actions = deliver
High Scoring MCP Actions = deliver
Bounce MCP As Attachment = no
MCP Modify Subject = start
MCP Subject Text = {MCP?}
High Scoring MCP Modify Subject = start
High Scoring MCP Subject Text = {MCP?}
Is Definitely MCP = no
Is Definitely Not MCP = no
Definite MCP Is High Scoring = no
Always Include MCP Report = no
Detailed MCP Report = yes
Include Scores In MCP Report = no
Log MCP = no
MCP Max SpamAssassin Timeouts = 20
MCP Max SpamAssassin Size = 100k
MCP SpamAssassin Timeout = 10
MCP SpamAssassin Prefs File = %mcp-dir%/mcp.spam.assassin.prefs.conf
MCP SpamAssassin User State Dir =
MCP SpamAssassin Local Rules Dir = %mcp-dir%
MCP SpamAssassin Default Rules Dir = %mcp-dir%
MCP SpamAssassin Install Prefix = %mcp-dir%
Recipient MCP Report = %report-dir%/recipient.mcp.report.txt
Sender MCP Report = %report-dir%/sender.mcp.report.txt
Use Default Rules With Multiple Recipients = no
Read IP Address From Received Header = no
Spam Score Number Format = %d.1f
MailScanner Version Number = 4.85.1
SpamAssassin Cache Timings = 1800,300,10800,172800,600
Debug = no
Debug SpamAssassin = no
Run In Foreground = no
Always Looked Up Last = &BaruwaLog
Always Looked Up Last After Batch = no
Deliver In Background = yes
Delivery Method = batch
Split Exim Spool = no
Lockfile Dir = %spool-dir%/incoming/Locks
Custom Functions Dir = /usr/share/baruwa/CustomFunctions
Lock Type =
Syslog Socket Type =
Automatic Syntax Check = yes
Minimum Code Status = supported
```

2.5.14 filename.rules

```
From:           127.0.0.1       /etc/MailScanner/filename.rules.allowall.conf
FromOrTo:       default         /etc/MailScanner/filename.rules.conf
```

2.5.15 filetype.rules

```
From:           127.0.0.1       /etc/MailScanner/filetype.rules.allowall.conf
FromOrTo:       default         /etc/MailScanner/filetype.rules.conf
```

2.5.16 nonspam.actions.rules

```
FromOrTo:       default         deliver
```

2.5.17 scan.messages.rules

```
From:           127.0.0.1      no
FromOrTo:       default        yes
```

2.5.18 content.scanning.rules

```
FromOrTo:       *.blackberry.net        no
From:           127.0.0.1      no
FromOrTo:       default        yes
```

2.5.19 filename.rules.allowall.conf

```
allow   .*      -       -
```

2.5.20 filetype.rules.allowall.conf

```
allow   .*      -       -
```

2.5.21 nginx.conf

```
upstream baruwacluster {
    ip_hash;
    server unix:///var/run/baruwa/baruwa.sock;
        # Use this in cluster mode to connect to other servers
        #server xxx.xxx.xxx.xxx:3021;
}

server {
    listen [::]:80;
    server_name baruwa.example.com;
    access_log /var/log/nginx/baruwa-access.log combined;
```

```
    error_log /var/log/nginx/baruwa-error.log;
    charset utf-8;
    add_header X-Content-Type-Options "nosniff";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Frame-Options "SAMEORIGIN";
    add_header Strict-Transport-Security "max-age=631138519";
    root /var/empty;
    rewrite ^(.*)$ https://baruwa.example.com$1 permanent;
}

server {
    listen [::]:443;
    ssl on;
    ssl_certificate /etc/pki/baruwa/certs/baruwa.example.com.pem;
    ssl_certificate_key /etc/pki/baruwa/private/baruwa.example.com.key;
    keepalive_requests    50;
    keepalive_timeout     300 300;
    server_tokens off;
    server_name baruwa.example.com;
    access_log /var/log/nginx/baruwa-access.log combined;
    error_log /var/log/nginx/baruwa-error.log;
    charset utf-8;
    add_header X-Content-Type-Options "nosniff";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Frame-Options "SAMEORIGIN";
    add_header Strict-Transport-Security "max-age=631138519";
    root /usr/lib/python2.6/site-packages/baruwa/public;
    index index.html index.htm;
    client_max_body_size 25M;

    location /robots.txt {
      log_not_found off;
      access_log off;
      return 404;
    }

    location ~/default\/(imgs|js|css|font)/ {
      root /usr/share/baruwa/themes/assets;
      expires max;
      add_header Cache-Control "public";
      break;
    }

    location ~/.+\/(imgs|js|css|font)/ {
      root /usr/share/baruwa/themes/assets;
      expires max;
      add_header Cache-Control "public";
      break;
    }

    location ~/(imgs|js|css|font)/ {
      root /usr/lib/python2.6/site-packages/baruwa/public;
      expires max;
      add_header Cache-Control "public";
      break;
    }

    location = /favicon.ico {
```

```
      root /usr/lib/python2.6/site-packages/baruwa/public/imgs;
      expires max;
      add_header Cache-Control "public";
      break;
    }

    location = /default/favicon.ico {
      root /usr/share/baruwa/themes/assets/default/imgs;
      expires max;
      add_header Cache-Control "public";
      break;
    }

    location = ~/.+\/favicon\.ico/ {
      root /usr/share/baruwa/themes/assets/$host/imgs;
      expires max;
      add_header Cache-Control "public";
      break;
    }

    location / {
      uwsgi_pass baruwacluster;
      include uwsgi_params;
      uwsgi_param SCRIPT_NAME '';
      uwsgi_param UWSGI_SCHEME $scheme;
    }
}
```

2.5.22 sphinx.conf

```
searchd
{
    listen = 0.0.0.0:9312
    listen = 0.0.0.0:9306:mysql41
    log = /var/log/sphinx/searchd.log
    query_log  = /var/log/sphinx/query.log
    read_timeout = 5
    max_children = 30
    pid_file     = /var/run/sphinx/searchd.pid
    max_matches  = 500
    seamless_rotate = 1
    preopen_indexes = 0
    unlink_old = 1
    workers = threads
    collation_server = utf8_general_ci
    rt_flush_period = 3600
    binlog_path = /var/lib/sphinx
}

indexer
{
        mem_limit = 256M
        max_iops = 30
        max_iosize = 4M
}

source base
```

```
{
    type = pgsql
    sql_host = 127.0.0.1
    sql_user = baruwa
    sql_pass = password
    sql_db = baruwa
}

source messages : base
{
    sql_query_range = SELECT MIN(id), MAX(id) FROM messages WHERE ts < get_var('maxts')
    sql_range_step  = 10000
    sql_query_pre = SELECT set_var('maxts', NOW())
    sql_query = SELECT id, messageid, subject, CRC32(from_address) AS from_addr, CRC32(to_address) AS
                to_addr, CRC32(from_domain) AS from_dom, CRC32(to_domain) AS to_dom, headers, \
                hostname, UNIX_TIMESTAMP(timestamp) AS timestamp, isquarantined FROM messages \
                WHERE ts < get_var('maxts') AND id >= $start AND id <= $end
    sql_query_post = SELECT update_indexer_counters('messages_tmp', get_var('maxts'))
    sql_query_post_index = DELETE FROM indexer_counters WHERE tablename='messages'
        sql_query_post_index = UPDATE indexer_counters SET tablename='messages' WHERE tablename='mess
    sql_query_post_index = DELETE FROM indexer_killlist WHERE ts < (SELECT maxts FROM indexer_counter
                AND tablename='messages'
    sql_attr_uint = from_addr
    sql_attr_uint = to_addr
    sql_attr_uint = from_dom
    sql_attr_uint = to_dom
    sql_attr_timestamp = timestamp
    sql_attr_bool = isquarantined
}

source messagesdelta : base
{
    sql_query_range = SELECT MIN(id), MAX(id) FROM messages WHERE ts >= get_var('maxts') AND ts < get
    sql_range_step  = 10000
    sql_query_pre = SELECT set_var('maxts', (SELECT maxts FROM indexer_counters WHERE tablename='mess
    sql_query_pre = SELECT set_var('maxtsdelta', NOW())
    sql_query = SELECT id, messageid, subject, CRC32(from_address) AS from_addr, CRC32(to_address) AS
                to_addr, CRC32(from_domain) AS from_dom, CRC32(to_domain) AS to_dom, headers, hostnam
                UNIX_TIMESTAMP(timestamp) AS timestamp, isquarantined FROM messages \
                WHERE ts >= get_var('maxts') AND ts < get_var('maxtsdelta') \
                AND id >= $start AND id <= $end
    sql_query_killlist = SELECT id FROM messages WHERE ts >= get_var('maxts') AND ts < get_var('maxts
                UNION SELECT id FROM indexer_killlist WHERE tablename='messages'
    sql_query_post = SELECT update_indexer_counters('messages_delta_tmp', get_var('maxtsdelta'))
    sql_query_post_index = DELETE FROM indexer_counters WHERE tablename='messages_delta'
    sql_query_post_index = UPDATE indexer_counters SET tablename='messages_delta' WHERE tablename='me
    sql_attr_uint = from_addr
    sql_attr_uint = to_addr
    sql_attr_uint = from_dom
    sql_attr_uint = to_dom
    sql_attr_timestamp = timestamp
    sql_attr_bool = isquarantined
}

source archive : base
{
    sql_query_range = SELECT MIN(id), MAX(id) FROM archive WHERE ts < get_var('archive_maxts')
    sql_range_step  = 10000
```

```
    sql_query_pre = SELECT set_var('archive_maxts', NOW())
    sql_query = SELECT id, messageid, subject, CRC32(from_address) AS from_addr, CRC32(to_address) AS
                to_addr, CRC32(from_domain) AS from_dom, CRC32(to_domain) AS to_dom, \
                headers, hostname, UNIX_TIMESTAMP(timestamp) AS timestamp FROM archive \
                WHERE ts < get_var('archive_maxts') AND id >= $start AND id <= $end
    sql_query_post = SELECT update_indexer_counters('archive_tmp', get_var('archive_maxts'))
    sql_query_post_index = DELETE FROM indexer_counters WHERE tablename='archive'
    sql_query_post_index = UPDATE indexer_counters SET tablename='archive' WHERE tablename='archive_t
    sql_query_post_index = DELETE FROM indexer_killlist WHERE ts < (SELECT maxts FROM indexer_counter
                AND tablename='archive'
    sql_attr_uint = from_addr
    sql_attr_uint = to_addr
    sql_attr_uint = from_dom
    sql_attr_uint = to_dom
    sql_attr_timestamp = timestamp
}

source archivedelta : base
{
    sql_query_range = SELECT MIN(id), MAX(id) FROM archive WHERE ts >= get_var('archive_maxts') \
                AND ts < get_var('archive_maxtsdelta')
    sql_range_step  = 10000
    sql_query_pre = SELECT set_var('archive_maxts', (SELECT maxts FROM indexer_counters WHERE tablena
    sql_query_pre = SELECT set_var('archive_maxtsdelta', NOW())
    sql_query = SELECT id, messageid, subject, CRC32(from_address) AS from_addr, CRC32(to_address) AS
                to_addr, CRC32(from_domain) AS from_dom, CRC32(to_domain) AS to_dom, headers, hostnar
                UNIX_TIMESTAMP(timestamp) AS timestamp FROM archive WHERE ts >= get_var('archive_maxt
                AND ts < get_var('archive_maxtsdelta') AND id >= $start AND id <= $end
    sql_query_killlist = SELECT id FROM archive WHERE ts >= get_var('archive_maxts') AND \
                ts < get_var('archive_maxtsdelta') UNION SELECT id FROM indexer_killlist \
                WHERE tablename='archive'
    sql_query_post = SELECT update_indexer_counters('archive_delta_tmp', get_var('archive_maxtsdelta'
    sql_query_post_index = DELETE FROM indexer_counters WHERE tablename='archive_delta'
    sql_query_post_index = UPDATE indexer_counters SET tablename='archive_delta' WHERE tablename='arc
    sql_attr_uint = from_addr
    sql_attr_uint = to_addr
    sql_attr_uint = from_dom
    sql_attr_uint = to_dom
    sql_attr_timestamp = timestamp
}

source lists : base
{
    sql_query = SELECT id, from_address AS froma, to_address AS to, from_address, to_address, list_ty
    sql_attr_str2ordinal = from_address
    sql_attr_str2ordinal = to_address
    sql_attr_uint = list_type
    sql_attr_uint = user_id
}

source organizations : base
{
  sql_query = SELECT id, name FROM organizations
  sql_attr_multi = uint admins from query; \
                SELECT organization_id, user_id FROM organizations_admins
}

source domains : base
```

```
{
    sql_query = SELECT id, name, CRC32(name) AS domain_name FROM maildomains
    sql_attr_uint = domain_name
    sql_attr_multi = uint orgs from query; \
                SELECT domain_id, organization_id FROM domain_owners
                                }

source accounts : base
{
  sql_query = SELECT id, username, firstname, lastname, email, active, local FROM users
  sql_attr_uint = active
  sql_attr_uint = local
  sql_attr_multi = uint domains from query; \
                SELECT user_id, domain_id FROM domain_users
}

source auditlog : base
{
    sql_query = SELECT id, username, info, hostname, remoteip, category, timestamp FROM auditlog
    sql_attr_uint = category
    sql_attr_timestamp = timestamp
}

index lists
{
        source = lists
        path = /var/lib/sphinx/lists
        docinfo = extern
        charset_type = utf-8
        ondisk_dict = 1
}

index lists_rt
{
        type = rt
        path = /var/lib/sphinx/lists_rt
        rt_field = froma
        rt_field = to
        rt_attr_string = from_address
        rt_attr_string = to_address
        rt_attr_bigint = list_type
        rt_attr_bigint = user_id
        docinfo = extern
        charset_type = utf-8
}

index organizations
{
        source = organizations
        path = /var/lib/sphinx/organizations
        docinfo = extern
        charset_type = utf-8
        ondisk_dict = 1
}

index organizations_rt
{
        type = rt
```

```
        path = /var/lib/sphinx/organizations_rt
        rt_field = name
}

index domains
{
        source = domains
        path = /var/lib/sphinx/domains
        docinfo = extern
        charset_type = utf-8
        ondisk_dict = 1
}

index domains_rt
{
        type = rt
        path = /var/lib/sphinx/domains_rt
        rt_field = name
        rt_attr_bigint = domain_name
}

index accounts
{
        source = accounts
        path = /var/lib/sphinx/accounts
        docinfo = extern
        charset_type = utf-8
        ondisk_dict = 1
}

index accounts_rt
{
        type = rt
        path = /var/lib/sphinx/accounts_rt
        rt_field = username
        rt_field = firstname
        rt_field = lastname
        rt_field = email
        rt_attr_bigint = active
        rt_attr_bigint = local
}

index messages
{
        source = messages
        path = /var/lib/sphinx/messages
        docinfo = extern
        charset_type = utf-8
        ondisk_dict = 1
}


index messages_rt
{
        type = rt
        path = /var/lib/sphinx/messages_rt
        rt_field = messageid
        rt_field = subject
```

```
        rt_field = headers
        rt_field = hostname
        rt_attr_bigint = from_addr
        rt_attr_bigint = to_addr
        rt_attr_bigint = from_dom
        rt_attr_bigint = to_dom
        rt_attr_timestamp = timestamp
        rt_attr_uint = isquarantined
        docinfo = extern
        charset_type = utf-8
}

index archive
{
        source = archive
        path = /var/lib/sphinx/archive
        docinfo = extern
        charset_type = utf-8
        ondisk_dict = 1
}

index auditlog
{
        source = auditlog
        path = /var/lib/sphinx/auditlog
        docinfo = extern
        charset_type = utf-8
        ondisk_dict = 1
}

index auditlog_rt
{
        type = rt
        path = /var/lib/sphinx/auditlog_rt
        rt_field = username
        rt_field = info
        rt_field = remoteip
        rt_field = hostname
        rt_attr_bigint = category
        rt_attr_timestamp = timestamp
        docinfo = extern
        charset_type = utf-8
}

index archivedelta
{
        source = archivedelta
        path = /var/lib/sphinx/archivedelta
        docinfo = extern
        charset_type = utf-8
        ondisk_dict = 1
}

index messagesdelta
{
        source = messagesdelta
        path = /var/lib/sphinx/messagesdelta
        docinfo = extern
```

```
        charset_type = utf-8
        ondisk_dict = 1
}
```

CHAPTER

THREE

ADVANCED CONFIGURATION

3.1 External Authentication

Baruwa can be configured to authenticate to external authentication systems using authentication mechanisms such as LDAP, RADIUS, IMAP, POP3, SMTP, OAUTH. This is useful in cases where you have hundreds of users and cannot manually create all of them. The Baruwa user account will be automatically created the first time the user successfully authenticates to the external authentication system.

With LDAP authentication the users groups and email aliases will also be automatically added to the users Baruwa profile allowing them access to their aliased and group emails within Baruwa.

3.1.1 Supported Mechanisms

The following mechanisms are supported and can be fully configured via the web interface.

- LDAP
- RADIUS
- IMAP
- POP3
- SMTP

3.1.2 Configuration

Authentication mechanisms are setup on a per domain basis. The process is documented in the Domain management section of the admin guide under *Authentication Settings*

3.1.3 Planned Mechanisms

Future support is planned for the following

- YUBIKEY
- OAUTH

3.2 Clustering

3.2.1 Functionality available

Baruwa is capable of running in a cluster.

Full Baruwa functionality is available from any member within a Baruwa cluster and all cluster members have equal status. This allows you to provide round robin access either using a load balancer or DNS configuration. This makes the running of a cluster totally transparent to the end users.

Cluster wide as well as node status information is visible via *Global status* and *Scanner node status*

3.2.2 Requirements

Baruwa stores client session information in Memcached, so all the nodes in the cluster should be configured to use the same Memcached server.

All nodes should be configured to either use a clustered MQ broker or use the same MQ broker as the other nodes. The nodes should be aware of the other nodes queues to enable them to submit tasks to those queues.

All the nodes with in a cluster should be configured to write to a single database and index data to a single or distributed sphinx server.

The full requirements are:

- Shared Memcached server
- Shared PostgreSQL server
- Shared MQ broker or clustered broker
- Shared Sphinx server or distributed sphinx servers

The recommended setup is to have Memcached, PostgreSQL, RabbitMQ, Sphinx running on a separate server.

The firewall on the server hosting the above shared services needs to be configured to allow the following connections from the cluster nodes.

- TCP 9312, 9306 - Sphinx
- TCP 5432 - PostgreSQL or 6432 Pgbouncer
- TCP 4369 - RabbitMQ EPMD
- TCP 11211 - Memcached

3.2.3 Shared quarantine

Since version 2.0.1 Baruwa supports shared quarantines using shared storage subsystems like NFS, GlusterFS, OCFS, etc. With a shared quarantine, message operations are still possible regardless of non availability of the node that processed the message. To use a shared quarantine you need to:

- Mount the quarantine directory to the shared file subsystem
- Set the Baruwa configuration option `ms.quarantine.shared` to true
- Ensure that Exim generates unique message ids by setting the `localhost_number` option
- Ensure the `celeryd` and `exim` user ids are same for all nodes in the cluster

3.2.4 Limitations

Note: This limitation is not present when using a shared quarantine.

Quarantines are node specific, so messages quarantined on a failed node will not be accessible until the node is restored.

3.3 Themes

Themes, also known as skins, in the Baruwa Enterprise Edition are a combination of Mako Template, CSS and JS files that control the appearance of the Baruwa Web interface as well as reports and emails sent out by the system.

The theme system allows you to easily change the appearance of Baruwa, for example, to use the logo and colors of your company or institution.

There are two kinds of themes:

- `Default theme`
- `Hostname/Domain linked themes`

A `Default theme` can be used to override the built-in appearance for all hosts and domains on a server. A Default theme must be named `default` and only one default theme can be configured on a server.

`Hostname/Domain Themes` are linked to the hostname used to access the Baruwa server and the domain user accounts belong to, which means that you can virtual host various brands on the same server with different appearance and product name for each.

Using themes ensures that the changes you make survive upgrades as opposed to changes made to the built-in template and asset files shipped with Baruwa which get overwritten during an upgrade.

3.3.1 What can be customized

- Logos
- Web interface
- Emails
- Reports
- Product name
- Product url

3.3.2 Guidelines

- Themes `MUST` retain the copyright notice at the bottom.

3.3.3 Configuration

The default configuration assumes that themes are stored under the following directory `/usr/share/baruwa/themes` with the following directory structure:

```
/templates/default/
/templates/<hostname>/
/templates/<domainname>/
/assets/default/
/assets/<hostname>/
/assets/<domainname>/
```

Themes are configured by:

- Pointing the web server configuration for assets to the `default` and `site's` asset directory
- Setting the `baruwa.themes.base` to the directory containing the themes
- Setting the `baruwa.custom.name` to the custom product name
- Setting the `baruwa.custom.url` to the custom product web url

3.3.4 Creating a simple theme

To start off, you simply copy the built-in templates and assets into the a theme directory for the hostname you would like to customize for.

I will be using the hostname `spamfighter.example.com`:

```
BARUWA_PATH=$(python -c "from distutils.sysconfig import get_python_lib; print get_python_lib()")
mkdir -p /usr/share/baruwa/themes/assets/spamfighter.example.com/
mkdir -p /usr/share/baruwa/themes/templates/spamfighter.example.com/
cp -a $BARUWA_PATH/baruwa/templates/* /usr/share/baruwa/themes/templates/spamfighter.example.com/
cp -a $BARUWA_PATH/baruwa/public/* /usr/share/baruwa/themes/assets/spamfighter.example.com/
```

You can now modify the changes to the templates under `/usr/share/baruwa/themes/templates/spamfighter.example` and the CSS, JS and image files under `/usr/share/baruwa/themes/assets/spamfighter.example.com/`

In order to brand other non web interfaces such as email you need to link the themes to the domain name you want to brand.

For example to theme the domain name `example.com`:

```
ln -s /usr/share/baruwa/themes/assets/spamfighter.example.com \
/usr/share/baruwa/themes/assets/example.com
ln -s /usr/share/baruwa/themes/templates/spamfighter.example.com \
/usr/share/baruwa/themes/templates/example.com
```

3.3.5 Default theme

A default theme allows you to customize all the domains on your system using one theme. To create a default theme, simply create templates and assets directories named `default`:

```
BARUWA_PATH=$(python -c "from distutils.sysconfig import get_python_lib; print get_python_lib()")
mkdir -p /usr/share/baruwa/themes/assets/default
mkdir -p /usr/share/baruwa/themes/templates/default
cp -a $BARUWA_PATH/baruwa/templates/* /usr/share/baruwa/themes/templates/default/
cp -a $BARUWA_PATH/baruwa/public/* /usr/share/baruwa/themes/assets/default/
```

You can now modify the changes to the templates under `/usr/share/baruwa/themes/templates/default/` and the CSS, JS and image files under `/usr/share/baruwa/themes/assets/default/`

3.3.6 Creating themes from scratch

It is possible to totally redesign the Baruwa interface using a theme, this requires an understanding of the data being sent into the template files by Baruwa as well as the Mako Template language.

Theme customization services are available from the author.

3.3.7 Emails and Reports

In order to send out reports and emails that are customized using the above configurations you need to use the new generation commands.

- `paster send-quarantine-reports-ng` for quarantine reports
- `send-pdf-reports-ng` for pdf reports

3.4 Addons

3.4.1 Message Sniffer

The Message Sniffer software is designed to be installed on an email server or filtering appliance. Message Sniffer is driven by a professionally managed rulebase, available via subscription, that is continuously monitored and updated by intelligent machines and highly trained analysts. This teamwork between synthetic intelligence and extraordinary people reduces your administrative workload to a minimum and allows SNF to respond quickly (within minutes) to new threats while also predicting future hazards so they can be blocked before they arrive. Details on Message Sniffer can be found on their website at http://www.armresearch.com/Products/aboutSNF.jsp

Baruwa Enterprise Editions integrates with the Message Sniffer software.

Purchase

Message Sniffer subscriptions are available for purchase from us at discounted list prices. To purchase a Message Sniffer subscription please contact us.

Installation

The automated install system is capable of installing and configuring Message Sniffer software. In order to install Message Sniffer using the automated system, you need to contact us to purchase a subscription we will email you an AUTHENTICATION ID as well as a LICENSE ID. You should then set the following options in your manifest file.

```
$spamassassin_snf_enable = 'true'
$spamassassin_snf_licenseid = 'license id'
$spamassassin_snf_authentication = 'authentication id'
```

Then run the following command:

```
puppet -v /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
```

To install manually you need to install the software by running the following command:

```
yum install  spamassassin-plugin-snf snf snf-server -y
```

You then need to edit the following files and set the license id and authentication id.

- `/etc/snf-server/identity.xml`
- `/etc/sysconfig/snf-server`

After which you need to restart the server:

```
service snf-server restart
```

3.5 Additional Commercial Anti Virus Engines

Baruwa Enterprise Editions runs the ClamAV Anti Virus engine at SMTP time. You can however ran additional Anti Virus Engines after SMTP time with in the MailScanner Engine.

The following Commercial Anti Virus Engines are supported.

- ESet
- F-Secure
- F-Prot

3.5.1 Installation and Configuration

Follow the instructions provided by the Anti Virus vendor to install the software, then select the Anti Virus package in the settings section of the interface.

CHAPTER

FOUR

ADMINISTRATORS GUIDE

4.1 Managing Organizations

Organizations enable easy management of large number of domains, Administrators are assigned to Organizations and can manage all the domains with in the organization.

You can create smaller organizations out of bigger organizations and add specific domains from a bigger organization to allow delegation of domain management.

4.1.1 Add an Organization

1. Mouse over `Organizations`
2. Click `Add Organization`
3. Enter the name in `Organization name`
4. Select domain in `Domains` list if they already exist
5. Select admins from `Admins` list if they already exist
6. Click the `Add organization` Button

4.1.2 Update an Organization

1. Click `Organizations`
2. Select organization > Click `Edit`
3. Make changes
4. Click the `Update organization` Button

4.1.3 Delete an Organization

1. Click `Organizations`
2. Select organization > Click `Delete`
3. Check `Delete Organization domains` if you want to delete domains belonging to the organization.
4. Click the `Delete organization` Button

4.1.4 Search for an Organization

If you have a large number of organizations you can search for an organization by name.

1. Click `Organizations`
2. Enter the organization name in the search box
3. Click the `Search` Button

4.1.5 List all domains that belong to an organization

To find all domains that belong to a specific organization.

1. Click `Organizations`
2. Select organization > Click `List domains`

4.1.6 List all accounts that belong to an organization

To find all accounts that belong to a specific organization.

1. Click `Organizations`
2. Select organization > Click `List accounts`

4.1.7 Add a new domain to an organization

1. Click `Organizations`
2. Select organization > Click `Add domain`
3. Enter the domain details
4. Click `Add domain`

4.1.8 Import domains in to an organization

Domains can be imported using a CSV formatted file. To import domains in to an organization.

1. Click `Organizations`
2. Select organization > Click `Import domains`
3. Browse for the CSV file by clicking `Browse` next to the `CSV file` field
4. Check `Skip first line` if your first line contains descriptions.
5. Click the `Import` Button

4.1.9 Export an Organization's user accounts

You can export all the user accounts with in an organization.

1. Click `Organizations`
2. Click the organization name
3. Click `Export accounts`

4. Click `Download the csv file`
5. Save the file to your computer

4.1.10 View Organization details

To view the details of an organization such as number of `domains`, `admins`, `relay settings`

1. Click `Organizations`
2. Click the organization name

4.1.11 Add Outbound SMTP relay settings

Relaying of outbound mail is authenticated on a per organization basis, to enable an organization to send outbound mail through Baruwa you need to add relay settings.

Two kinds of outbound relaying are supported.

- IP address
- SMTP AUTH

Add Outbound SMTP IP Address settings

This allows the specific IP address to send outbound mail through Baruwa.

1. Click `Organizations`
2. Click the organization name
3. Click `Add relay setting`
4. Enter the IP address in the `Hostname` field
5. Ensure the `Enabled` checkbox is checked
6. Click `Add settings`

Add Outbound SMTP AUTH settings

This allows any client that supplies these credentials to send outbound mail through Baruwa.

1. Click `Organizations`
2. Click the organization name
3. Click `Add relay setting`
4. Ensure the `Enabled` checkbox is checked
5. Enter the username in the `SMTP-AUTH username` field
6. Enter the password in the `SMTP-AUTH password` field
7. Reenter the password in the `Retype Password` field
8. Click `Add settings`

4.2 Managing Domains

4.2.1 Adding a Domain

Domains can be added by either importing them using a CSV file or by adding them using the `Add domain` form.

To add a domain by import refer to *Import domains in to an organization*. To add a domain using the `Add domain` form,

1. Mouse over `Domains`
2. Click `Add a domain`
3. Enter the domain details
4. Click the `Add domain` Button

4.2.2 Updating a Domain

1. Click `Domains`
2. Select the domain > Click `Edit` under actions
3. Update the details you want to change
4. Click the `Update Domain` Button

4.2.3 Deleting a Domain

1. Click `Domains`
2. Select the domain > Click the `Domain name`
3. Click `Delete domain`
4. Click the `Delete Domain` Button

4.2.4 Exporting Domains

Domains can be exported to CSV, To export domains.

1. Click `Domains`
2. Click `Export Domains`
3. Click `Download the csv file`
4. Save the CSV file to your computer

4.2.5 Domain Settings

Each domain has a range of additional settings that you can configure. These include *Delivery Servers*, *Authentication Settings*, *Alias Domains*, *DKIM*, *Signatures*

Delivery Servers

Delivery servers are the actual mail servers hosting the email accounts where messages processed by Baruwa need to be delivered.

Multiple servers per domain are supported and they can be configured to either `load balance` or `fail over`.

In `load balance` mode mail is sent to the group of servers in a `round robin` manner while in `fail over` mail is sent to the first in the list and only to the others if the first is not available.

Adding a delivery server

1. Click `Domains`
2. Select the domain > Click the actions `settings` icon
3. Click `Add delivery server`
4. Enter server IP address or Hostname in the `Server address` field
5. Select the protocol in the `Protocol` drop down
6. Change the port in the `Port` field if your mail server does not use port 25
7. Ensure the `Enabled` checkbox is checked
8. Click the `Add server` button

Editing a delivery server

1. Click `Domains`
2. Select the domain > Click the `Domain name`
3. Scroll to the bottom
4. Select the `delivery server` > Click `Edit`
5. Make changes
6. Click the `Update server` button

Deleting a delivery server

1. Click `Domains`
2. Select the domain > Click the `Domain name`
3. Scroll to the bottom under `Delivery Servers`
4. Select the `delivery server` > Click `Delete`
5. Click the `Delete server` button

Authentication Settings

Authentication settings allow users within a domain be be authenticated to an external authentication system.

This can be used for centralized user management and to allow users to use existing authentication credentials instead of creating duplicate accounts on the Baruwa system.

The supported external authentication mechanisms include:

- `AD/LDAP`
- `SMTP`
- `POP3`
- `IMAP`
- `RADIUS`

The following mechanisms are planned but have not been implemented yet:

- `YUBIKEY`
- `OAUTH`

The AD/LDAP mechanism allows for the user details in the directory to be automatically updated to the Baruwa account created for them. These details include:

- `First name`
- `Last name`
- `Primary Email Address`
- `Alias Email Addresses`

Adding Authentication Settings

1. Click `Domains`
2. Select the domain > Click the actions `settings` icon
3. Click `Add Authentication settings`
4. Enter server IP address or Hostname in the `Server address` field
5. Select the Authentication protocol in the `Protocol` drop down
6. Enter the port in the `Port` field
7. Ensure the `Enabled` checkbox is checked
8. Enter a username map template if your usernames require translation e.g `Webmin` creates usernames like `domainowner.username` the template would be `domainowner.%(user)` For available variables see *Username map template variables*
9. Click the `Add` button

The `AD/LDAP` and `RADIUS` mechanisms require additional settings which can be added by *Adding AD/LDAP Authentication additional settings* and *Adding RADIUS Authentication additional settings*.

Username map template variables

Username map templates allow you to map Baruwa logins to complex user naming schemes such as those used by web hosting control panels for `virtual accounts`.

The following variables are available to your `username map template`:

- `%(user)s` - replaced by user part of the login
- `%(domain)s` - replaced by the domain part of the login

Adding AD/LDAP Authentication additional settings

AD/LDAP authentication requires the following additional setting.

- `Base DN` - The LDAP Directory Base DN
- `Username attribute` - The username attribute, defaults to `uid`
- `Email attribute` - The email attribute, defaults to `mail`
- `Bind DN` - The BIND DN if Directory does not allow anonymous binds
- `Bind password` - The BIND password
- `Use TLS` - Use a TLS connection
- `Search for UserDN` - Find the UserDN then Bind to that
- `Auth Search Filter` - Filter used to find the UserDN, *LDAP Search Filter Variables* are supported
- `Auth Search Scope` - Search Scope, defaults to `subtree`
- `Email Search Filter` - Filter used to find email addresses, *LDAP Search Filter Variables* are supported
- `Email Search Scope` - Search Scope, defaults to `subtree`

To Add AD/LDAP Authentication additional settings:

1. Click `Domains`
2. Select the domain > Click the `Domain name`
3. Scroll to the bottom under `Authentication Servers`
4. Select the LDAP `Authentication server` > Click `Settings`
5. Enter the required settings
6. Click the `Save settings` button

LDAP Search Filter Variables

The following variables are available for use in your LDAP search filters.

- `%n` - login (user@domain)
- `%u` - user (user part of the login)
- `%d` - domain (domain part of the login)
- `%D` - domainDN (domain DN)

Adding RADIUS Authentication additional settings

The RADIUS protocol requires a shared secret between the client and the server, the additional settings allows you to configure this.

To Add RADIUS Authentication additional settings:

1. Click `Domains`
2. Select the domain > Click the `Domain name`
3. Scroll to the bottom under `Authentication Servers`
4. Select the RADIUS `Authentication server` > Click `Settings`
5. Enter the shared secret in the `Radius secret` field
6. Click the `Save settings` button

Alias Domains

Some domains have mail addressed to the same account using different domain names, Alias domains allow users access to all their messages regardless of the domain name under a single login.

Adding an Alias Domain

1. Click `Domains`
2. Select the domain > Click the actions `settings` icon
3. Click `Add Alias Domain`
4. Enter Alias domain name in the `Domain alias name` field
5. Ensure the `Enabled` checkbox is checked
6. Click the `Add` button

DKIM

DomainKeys Identified Mail (DKIM) is a method for associating a domain name to an email message, thereby allowing a person, role, or organization to claim some responsibility for the message. The association is set up by means of a digital signature which can be validated by recipients. Wikipidia

Baruwa allows you to manage the digital signatures within the interfaces and signs any outbound messages for which DKIM is enabled.

Generate DKIM Keys

To generate DKIM keys for a domain,

1. Click `Domains`
2. Select the domain > Click the actions `settings` icon
3. Click `DKIM` > `Generate DKIM keys`
4. Select `DNS record` and add to you DNS zone

Enable DKIM signing

1. Make sure your have followed the steps in *Generate DKIM Keys*
2. Click `Domains`
3. Select the domain > Click the actions `settings` icon
4. Click `DKIM` > `Enable/Disable DKIM signing`
5. Ensure the `Enabled` checkbox is checked
6. Click the `Submit` button

Regenerate DKIM keys

1. Click `Domains`
2. Select the domain > Click the actions `settings` icon
3. Click `DKIM` > `Regenerate DKIM keys`
4. Select `DNS record` and update your DNS zone

Signatures

Baruwa can manage email signatures / disclaimers that are added to messages that are sent outbound through it. Both HTML and Text signatures are supported. HTML signatures can contain a single embedded image.

Adding Signatures/Disclaimers

1. Click `Domains`
2. Select the domain > Click the actions `settings` icon
3. Click `Signatures` > `Add signature`
4. Select `Signature type` from the drop down
5. Enter signature content
6. Ensure the `Enabled` checkbox is checked
7. Click the `Add signature` button

Importing Accounts

Accounts can be imported into a domain using a CSV file.

1. Click `Domains`
2. Select the domain > Click the actions `settings` icon
3. Click `Import accounts`
4. Browse for the CSV file by clicking `Browse` next to the `CSV file` field
5. Check `Skip first line` if your first line contains descriptions.
6. Click the `Import` Button

Exporting Accounts

Accounts can be exported from a domain to a CSV file.

1. Click `Domains`
2. Select the domain > Click the actions `settings` icon
3. Click `Export accounts`
4. Click `Download the csv file`
5. Save the file to your computer

Rulesets

Note: Domain specific rule sets are not implemented yet.

4.2.6 Searching for Domains

If you have a large number of domains you can search for a domain by name.

1. Click `Domains`
2. Enter the Domains name in the search box
3. Click the `Search` Button

4.2.7 Bulk domain management

To `enable`, `disable` or `delete` multiple domains:

1. Click `Domains`
2. Use the checkbox to select the domains
3. Select `enable` or `disable` or `delete` at the top
4. Click the `Submit` button

4.3 Managing Accounts

4.3.1 Adding an Account

Accounts can be added by either importing them using a CSV file or by adding them using the `Add Account` form.

To add an Account by import refer to *Importing Accounts*. To add a Account using the `Add Account` form:

1. Mouse over `Accounts`
2. Click `Add Account`
3. Enter the Account details
4. Click the `Create Account` button

4.3.2 Updating an Account

1. Click `Accounts`
2. Select the account > Click `Edit` under actions
3. Update the details you want to change
4. Click the `Update account` button

4.3.3 Deleting an Account

1. Click `Accounts`
2. Select the Account > Click the `Account name`
3. Click `Delete account`
4. Click the `Delete Account` button

4.3.4 Exporting Accounts

Accounts can be exported to CSV, To export accounts.

1. Click `Accounts`
2. Click `Export Accounts`
3. Click `Download the csv file`
4. Save the CSV file to your computer

4.3.5 Search for Accounts

If you have a large number of accounts you can search for an account or accounts by name.

1. Click `Accounts`
2. Enter the Accounts name in the search box
3. Click the `Search` Button

4.3.6 Add account signatures

Baruwa can manage email signatures / disclaimers that are added to messages that are sent outbound through it. Both HTML and Text signatures are supported. HTML signatures support a single embedded image.

Account specific signatures/disclaimers can be setup.

1. Click `Accounts`
2. Select the Account > Click the Username
3. Click `Add signature`
4. Select `Signature type` from the drop down
5. Enter signature content
6. Ensure the `Enabled` checkbox is checked

7. Click the `Add signature` button

4.3.7 Changing an Account password

Domain administrator and normal user account passwords can be changed using the web interface, administrator accounts can only be changed using the command line.

To change an account password:

1. Click `Accounts`
2. Select the Account > Click the Username
3. Click `Change password`
4. Enter the password in the `New Password` field
5. Reenter the password in the `Retype Password` field
6. Click the `Change password` button

4.3.8 Bulk account management

To `enable`, `disable` or `delete` multiple accounts:

1. Click `Accounts`
2. Use the checkbox to select the accounts
3. Select `enable` or `disable` or `delete` at the top
4. Click the `Submit` button

4.4 Managing Settings

4.4.1 Adding a scanning Node

In order to manage the scanner settings as well as get status information on your Baruwa servers you need to add them as scanning nodes.

1. Mouse over `Settings`
2. Click `Add scanning node`
3. Enter the Hostname in the `Hostname` field
4. Ensure the `Enabled` checkbox is checked
5. Click the `Add node` button

4.4.2 Customize Node scanner settings

You can customize scanner settings for a specific node.

1. Click `Settings`
2. Select the scanning node > Click settings under actions
3. Make the changes

4. Click the `Save settings` button

4.4.3 Customize the Global scanner settings

These settings apply to all scanners that are managed from within this interface.

1. Mouse over `Settings`
2. Click `MailScanner settings`
3. Make the changes
4. Click the `Save settings` button

4.5 System Status

System status gives you a dash board view of your Baruwa system or cluster.

The following information is provided:

- Global status
- Scanner node status
- Mail Queues
- Audit logs

4.5.1 Global status

The global status dashboard gives you the status information for the whole of your Baruwa system/cluster at a glance.

Day's processed message totals

- Number of messages processed
- Number of messages found to be clean
- Number of messages found to be High scoring spam
- Number of messages found to be Low scoring spam
- Number of messages found to be Virus infected
- Number of messages found to be Policy blocked
- Number of messages in the Inbound queues
- Number of messages in the Outbound queues

Graph of Day's processed message totals

A graphical view of the above information in a PIE chart graph.

Scanner node status

The status of all the scanning nodes in this Baruwa cluster.

4.5.2 Scanner node status

Provides the status of a specific scanning node, and allows you to pull additional information via select commands.

The following status information is provided.

- Day's stats for the specific node
- Node Hardware status (`CPU`, `Memory`, `Disk`, `Network`)
- System Network stats
- System software status (`Scanners`, `MTA`, `Anti Virus engine`)

4.5.3 Mail Queues

The status of both the `inbound` and `outbound` mail queues is provided. The following actions can be performed on messages that are in the queues:

- Delivery
- Bounce
- Hold
- Delete
- Preview

Details on how to carry our the above actions can be found in the user guide's *Processing queued messages* section.

4.5.4 Audit logs

Audit logs are provided for the interactions that users have with the system. The following information is recorded.

- Date and Time
- Username
- Interaction information
- Baruwa Node hostname or IP address
- Users IP address
- Category

Interactions are classified under the following categories

- Read
- Create
- Auth
- Update

The Audit logs can be exported in both PDF and CSV formats for offline usage.

The Audit logs are searchable, all full text search options are supported. Tips on searching are available on the *Baruwa Search Tips and Tricks* page.

4.6 Command line Reference

Custom paster commands are provided to enable scripting of house keeping tasks such as quarantine management and Database maintenance.

4.6.1 Command options and help

These commands may take options to get details on the supported options run:

```
paster baruwa
paster COMMAND_NAME -h or paster help COMMAND_NAME
```

4.6.2 Quarantine management

```
paster prune-quarantine /etc/baruwa/production.ini
```

Deletes quarantined files older than `ms.quarantine.days_to_keep`. This is set in the `/etc/baruwa/production.ini` file

4.6.3 Quarantine reports

```
paster send-quarantine-reports-ng /etc/baruwa/production.ini
```

Generates an email report of the quarantined messages. This command allows you to specify the number of days the report should cover as well as the maximum number of messages to return. The following switches allow you to specify periods.

- `-o NUM_DAYS, --newer-than=NUM_DAYS` Report on messages this number of days back
- `-m MAX_MSGS, --max-records=MAX_MSGS` Maximum number of messages to return
- `-i ORG_ID, --org-id=ORG_ID` Process only this organization's accounts
- `-e EXCLUDE_ORG, --excluded-org=EXCLUDE_ORG` Exclude this organization's accounts

4.6.4 Database maintenance

```
paster prune-database /etc/baruwa/production.ini
```

Deletes records older than 30 days from the messages table of the database, and archives them to the archive table. It deletes records older than 90 days from the archives table. These defaults can be configured in the configuration file as the following options:

- `baruwa.messages.keep.days`
- `baruwa.archive.keep.days`

The following options allow you to specify the periods of the records that need to be processed.

- `-d --days` records older than this number are deleted from messages
- `-a --adays` records older than this number are deleted from archives

4.6.5 Spamassassin rule description updates

```
paster update-sa-rules /etc/baruwa/production.ini
```

Updates the Spamassassin rule descriptions in the database.

4.6.6 PDF reports

```
paster send-pdf-reports-ng /etc/baruwa/production.ini
```

Sends PDF reports by email. This command allows you to specify the report type [domain, user], report period [daily, weekly, monthly] and the number of days to report on. The following switches allow you to specify the options.

- `-t REPORT_TYPE, --report-type=REPORT_TYPE` Report type [user, domain]
- `-p REPORT_PERIOD, --report-period=REPORT_PERIOD` Report period [daily, weekly, monthly]
- `-d NUMBER_OF_DAYS, --number-of-days=NUMBER_OF_DAYS` Restrict to number of days
- `-i ORG_ID, --org-id=ORG_ID` Process only this organization's accounts
- `-e EXCLUDE_ORG, --excluded-org=EXCLUDE_ORG` Exclude this organization's accounts

4.6.7 Mail queue Stats updates

```
paster update-queue-stats /etc/baruwa/production.ini
```

Query the inbound and outbound queues and write stats to the database.

4.6.8 Delta search index updates

```
paster update-delta-index --index messages --realtime /etc/baruwa/production.ini
paster update-delta-index --index archive /etc/baruwa/production.ini
```

The `messages` and `archive` index have deltas to ensure that indexing is efficient the above commands merge the delta index with the main index and remove id's from the realtime index that have been indexed to disk indexes.

The `messages` index has a real time index while `archive` does not.

4.6.9 Create an administrator account

```
paster create-admin-user -u USERNAME -p PASSWORD -e EMAIL -t TIMEZONE /etc/baruwa/production.ini
```

Create an administrator account

4.6.10 Change user password

```
paster change-user-password --username USERNAME /etc/baruwa/production.ini
```

Changes an accounts password, This is the only way to change an administrator account's password as it cannot be changed via the web interface.

4.6.11 Generate list of top spammers

```
paster send-top-spammer-list -e EMAIL [-m -s SPAMSCORE -p REPORT_PERIOD -d] /etc/baruwa/production.ir
```

Generates a list of top spammers and emails or displays it.

- `-e EMAIL, --email=EMAIL` Email address to send data to
- `-m, --include-message-count` Include the number messages received
- `-d, --dry-run` Print to stdout do not send email
- `-n NUM, --messages-sent=NUM` Return senders with message counts equal to or greater than
- `-s SPAMSCORE, --spam-score-threshold=SPAMSCORE` Count messages with spam scores equal to or greater than
- `-p REPORT_PERIOD, --report-period=REPORT_PERIOD` Report period [daily, weekly, monthly]

4.6.12 Generates list of clean senders

```
paster send-whitelist-data -e EMAIL [-m -s SPAMSCORE -p REPORT_PERIOD -d] /etc/baruwa/production.ini
```

Generates a list of top ham senders for whitelisting.

- `-e EMAIL, --email=EMAIL` Email address to send data to
- `-m, --include-message-count` Include the number messages received
- `-d, --dry-run` Print to stdout do not send email
- `-n NUM, --messages-sent=NUM` Return senders with message counts equal to or greater than
- `-s SPAMSCORE, --spam-score-threshold=SPAMSCORE` Count messages with spam scores equal to or greater than
- `-p REPORT_PERIOD, --report-period=REPORT_PERIOD` Report period [daily, weekly, monthly]

4.7 Languages supported

The following languages are currently supported. Adding a new language is a simple task which can be done using the online translation service: Transifex which is used to manage our translations.

- English
- French
- German
- Greek
- Catalan
- Chinese
- Dutch
- Bulgarian
- Czech
- Danish

- Hindi
- Indonesian
- Italian
- Norwegian
- Polish
- Portuguese
- Russian
- Spanish
- Swedish
- Thai
- Turkish
- Japanese
- Romanian
- Arabic
- Hebrew
- Finnish
- Korean
- Latvian
- Ukrainian
- Urdu
- Vietnamese
- Persian
- Afrikaans
- Burmese
- Hungarian
- Slovak
- Swahili

4.8 FAQ's

Answers to many common questions.

4.8.1 I think I've found a security problem! What should I do?

Answer: `Email security@baruwa.com`

If you think you've found a security vulnerability with Baruwa, please send a message to security@baruwa.com. Do NOT post a bug report to our issue tracking system or disclose the issue on our mailing lists.

4.8.2 Can a user have multiple email addresses on a single account ?

Answer: `Yes`

You can add alias addresses to a users account. Domains using Active Directory authentication will have these auto populated from the groups and addresses in active directory.

Alias domain addresses are also auto created the first time a user logs in.

4.8.3 Can users use their current mail password to login to Baruwa ?

Answer: `Yes`

Setup external authentication with either POP3, IMAP, SMTP, LDAP and RADIUS / RSA SecurID.

4.8.4 Are there any restrictions on username format ?

Answer: `No`

However users that authenticate to external systems will have their email address automatically configured as their username locally.

4.8.5 Which operating systems are supported ?

Answer: `CentOS,RHEL,SL,OL`

Baruwa Enterprise supports both RPM and DEB based operating systems. The following operating systems are fully supported.

- RHEL 6
- CentOS 6
- Scientific Linux 6
- Oracle Linux 6

4.8.6 Which MTA does Baruwa Enterprise use ?

Answer: `Exim`

Baruwa Enterprise uses a customized version of the Exim MTA

4.9 Upgrading

4.9.1 2.0.5

Upgrade Type

- Enhancement
- Bug fix

Backward compatibility

This release introduces a backwards incompatible database schema change. The relaysettings table has been modified to support the relay settings description.

New dependencies

None

New configuration options

- `baruwa.memcached.host` - Sets the address of the memcached server, this used for the distributed locking in a cluster.

Upgrading

Review the changelog for version *2.0.5* and read the updated documentation before you proceed with the upgrade.

Backup your current system:

```
tar cjvf /usr/local/src/baruwa-configs.tar.bz2 /etc/baruwa
```

Modify the relaysettings table, you will need to supply the Baruwa PostgreSQL password:

```
psql -Ubaruwa baruwa
baruwa=> ALTER TABLE relaysettings ADD column description varchar(255);
```

Download and install the updated puppet toaster:

```
curl -O https://www.baruwa.com/downloads/puppet-toaster-latest.tar.bz2
tar xjvf puppet-toaster-latest.tar.bz2 -C /etc/puppet/
```

Perform the upgrade:

```
yum upgrade -y
cp /etc/puppet/manifests/toasters/baruwa/$(hostname).pp /etc/puppet/manifests/toasters/baruwa/$(hostn
/etc/puppet/bin/update-puppet-config.pl -oldconfig /etc/puppet/manifests/toasters/baruwa/$(hostname).
-newconfig /etc/puppet/manifests/toasters/baruwa/init.pp > /etc/puppet/manifests/toasters/baruwa/$(ho
puppet -v /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
service mailscanner restart
service uwsgi restart
service baruwa restart
```

4.9.2 2.0.4

Upgrade Type

- Enhancement
- Bug fix

Backward compatibility

This release introduces a backwards incompatible database schema change. The `quickpeek` database view has been modified to better order the options returned.

New dependencies

None

New configuration options

None

Upgrading

Review the changelog for version *2.0.4* and read the updated documentation before you proceed with the upgrade.

Backup your current system:

```
tar cjvf /usr/local/src/baruwa-configs.tar.bz2 /etc/baruwa
```

Download and install the updated puppet toaster:

```
curl -O https://www.baruwa.com/downloads/puppet-toaster-latest.tar.bz2
tar xjvf puppet-toaster-latest.tar.bz2 -C /etc/puppet/
```

Perform the upgrade:

```
yum upgrade -y
paster setup-app /etc/baruwa/production.ini
puppet -v /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
service mailscanner restart
service uwsgi restart
service baruwa restart
```

4.9.3 2.0.3

Upgrade Type

- Enhancement
- Bug fix

Backward compatibility

This release does not introduce any backwards incompatible changes.

New dependencies

None

New configuration options

- `baruwa.dkim.selector` - Sets the DKIM selector name `default: baruwa`

Upgrading

Review the changelog for version *2.0.3* and read the updated documentation before you proceed with the upgrade.

Backup your current system:

```
tar cjvf /usr/local/src/baruwa-configs.tar.bz2 /etc/baruwa
```

Download and install the updated puppet toaster:

```
curl -O https://www.baruwa.com/downloads/puppet-toaster-latest.tar.bz2
tar xjvf puppet-toaster-latest.tar.bz2 -C /etc/puppet/
```

Perform the upgrade:

```
yum upgrade -y
rm -rf /var/lib/baruwa/data/cache/*
rm -rf /var/lib/baruwa/data/sessions/*
rm -rf /var/lib/baruwa/data/templates/*
service uwsgi restart
service baruwa restart
puppet -v /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
```

4.9.4 2.0.2

Upgrade Type

- Enhancement
- Bug fix

Backward compatibility

This release introduces a backwards incompatible database schema change. The `UNIQUE INDEX` on the message-id field has been dropped to allow for duplicate message-id's to be supported. Duplicate message-id's may occur in high volume environments.

The template variables for the `messages/preview.html` and the `status/preview.html` templates have changed. The changes allow for the support of alternative message format display as well as displaying correctly formatted HTML messages. If you have customized your templates, you will need to review the new variable format and update your customized templates.

New dependencies

- cssutils
- pyzmail

New configuration options

None.

Upgrading

Review the changelog for version *2.0.2* and read the updated documentation before you proceed with the upgrade.

Backup your current system:

```
tar cjvf /usr/local/src/baruwa-configs.tar.bz2 /etc/baruwa
```

Download and install the updated puppet toaster:

```
curl -O https://www.baruwa.com/downloads/puppet-toaster-latest.tar.bz2
tar xjvf puppet-toaster-latest.tar.bz2 -C /etc/puppet/
```

Modify the message-id index, you will need to supply the Baruwa PostgreSQL password:

```
psql -Ubaruwa baruwa
baruwa=> DROP INDEX ix_messages_messageid;
baruwa=> CREATE INDEX ix_messages_messageid ON messages(messageid);
```

Perform the upgrade:

```
yum upgrade -y
rm -rf /var/lib/baruwa/data/cache/*
rm -rf /var/lib/baruwa/data/sessions/*
rm -rf /var/lib/baruwa/data/templates/*
service uwsgi restart
service baruwa restart
puppet -v /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
```

If you had customized your interface, then update the changed templates to use the new variables.

4.9.5 2.0.1

Upgrade Type

- Security [Severity: Medium]
- Bug fix
- Enhancement

Backward compatibility

This release does not introduce any backwards incompatible changes.

New dependencies

- sqlparse

New configuration options

- `ms.quarantine.shared` - Enables and disables shared quarantine features `default: disabled`
- `baruwa.themes.base` - Sets the directory containing themes `default: /usr/share/baruwa/themes`
- `baruwa.custom.name` - Sets the custom product name for rebranding `default: Baruwa Hosted`
- `baruwa.custom.url` - Sets the url for the product `default: http://www.baruwa.net/`

Upgrading

Baruwa Enterprise Edition has switched from using the certificate authenticated repository to a Spacewalk managed entitlement system. In order to access the new system you need to install the Spacewalk client tools and obtain an activation key for your server entitlement.

Review the changelog for version *2.0.1* and read the updated documentation before you proceed with the upgrade.

Backup your current system:

```
tar cjvf /usr/local/src/baruwa-configs.tar.bz2 /etc/baruwa
tar cjvf /usr/local/src/baruwa-software.tar.bz2 /usr/lib/python2.6/site-packages/baruwa
```

When ready to perform the upgrade, have your activation key handy then run the following commands, replace `<activation-key>` with your actual activation key:

```
rpm -Uvh https://www.baruwa.com/downloads/baruwa-enterprise-release-6-2.noarch.rpm
rpm -Uvh http://yum.spacewalkproject.org/1.9/RHEL/6/x86_64/spacewalk-client-repo-1.9-1.el6.noarch.rpm
yum install rhn-client-tools rhn-check rhn-setup rhnsd m2crypto yum-rhn-plugin -y
rhnreg_ks --serverUrl=http://bn.baruwa.com/XMLRPC --activationkey=<activation-key>
```

Download and install the updated puppet toaster:

```
curl -O https://www.baruwa.com/downloads/puppet-toaster-latest.tar.bz2
tar xjvf puppet-toaster-latest.tar.bz2 -C /etc/puppet/
```

Review the new options available to the puppet manifest and add to your previous manifest, then run:

```
yum upgrade -y
rm -rf /var/lib/baruwa/data/cache/*
rm -rf /var/lib/baruwa/data/sessions/*
rm -rf /var/lib/baruwa/data/templates/*
service uwsgi restart
service baruwa restart
puppet -v /etc/puppet/manifests/toasters/baruwa/$(hostname).pp
```

If you had customized your interface, then follow the theming guidelines to create a theme that will not be overridden by your next update.

4.10 Changelog

4.10.1 2.0.5

- Implemented distributed locking to enable only one cluster member to execute commands within the cluster.
- Implemented standalone search index update script for use within clusters.

- Fixed issues with LDAP attributes not being updated.
- Fixed the prune database command
- Added support for domain aliases in rulesets
- Improvements to the caching system
- Added support for the Esets and F-Secure AV engines
- Improved the display formatting of DKIM keys
- Added a description to relay settings
- Prevent normal users from downloading prohibited or infected attachments
- Various fixes and minor improvements
- Point data feeds to datafeeds.baruwa.com
- Updated documentation

4.10.2 2.0.4

- Moved the sphinx configuration options to MailScanner.conf, Sphinx configuration options moved from the BS.pm module into the MailScanner.conf file to simplify updating the module.
- Improved the ConfigSQL view with better ordering.
- Implemented deletion of default settings from ConfigSQL, Make sure that options are deleted from the ConfigSQL database when updated to the default value. Previously the values were left in the database.
- Implemented validation of MailScanner ConfigSQL options
- Implemented online help for Scanner settings
- Updated the forms to display online help
- Updated CSS to display help popups

4.10.3 2.0.3

- Fixed unicode encode error in spamassassin rules update command.
- Implemented locking to update delta command to ensure only one instance runs.
- Fixed quarantine clean command date format exception.
- Replaced old commands with their new generation versions.
- Fixed issue with fake charsets causing exceptions.
- Prevented cron.d file from being overwritten during update.
- Made improvements to authentication and authorization subsystems.
- Fixed prune quarantine command issue where customized cleanup days options were not being honored for the messages and archive tables.
- Fixed display of bayesian auto learn status, Bayes auto learn status was displayed incorrectly on the message detail page when bayes learning was disabled by the engine.
- Fixed sphinx indexing cronjobs.
- Fixed issue with incorrect attachments being downloaded when messages contain an embedded image.

- Fixed Spam rules display, preventing the "required score" from displaying as a rule.
- Fixed MailScanner config spamactions option which was not being picked up correctly.
- Fixed delivery status information, which incorrectly displayed as quarantined messages that had been deleted.
- Implemented Default theme support, which allows for global overriding of built-in appearance.
- Fixed branding issue where the logo was not being replaced with the theme version. Closes issue #19
- Implemented a configurable DKIM selector. Closes issue #17. A new option `baruwa.dkim.selector` introduced to allow configuration of the DKIM selector.
- Fixed Error when adding address to approved/banned senders using an alias domain. Closes issue #20
- Made default settings match supplied mailscanner configuration file. Closes issue #17.
- Fixed Information Header Value not applying. Closes issue #13
- Implemented the Blue lagoon theme as base template, this is built using responsive design which scales to display on all device sizes.
- Updated the translations.
- Updated the documentation.

4.10.4 2.0.2

- Fixed taskid session checks, which caused an exception when the session attribute did not exist.
- Fixed issue with headers which can not be decoded leading to exceptions
- Fixed issue with empty values breaking quarantine messages due to attempt to concat strings with None values.
- Added checks to prevent the creation of duplicate user accounts from external authentication mechanisms due to the case being different.
- Fixed the deletion of relay settings, which was causing an exception.
- Fixed accounts navigation issue, when paging using AJAX.
- Added support for custom logos in PDF reports, fixes issue #14.
- Fixed incorrect memory usage percentages in the status page.
- Improve daily totals calculation, it now supports users timezone settings.
- Fixed an exception with the Psutil backend which was not being caught.
- Added organization filters to the quarantine and pdf reports commands.
- Improvements to lost password handing, restrict requests to local users and fix the reset url.
- Added a top spammers generation command which can be used to export data to external or internal blacklists.
- Added a top clean senders generation command which can be used to export data to external or internal whitelists.
- Improvements to display all dates and times in users own timezone.
- Implemented JSON data exports to support JSON driven charts and graphs.
- Improvements to the search functions error handling.
- Improvements to the external authentication modules.
- Improvements to the message preview functionality, now able to display both the text and HTML alternatives of an email. HTML messages formatted correctly using embedded CSS styles which are sanitized.

- Added support for duplicate message id's which are generated on high mail volume installations.
- Various minor code cleanups and fixes.
- Updates to the documentation.

4.10.5 2.0.1

- Fixed domains information leak when logged in as domain admin. Domain admins were able to see domains belonging to other users in the drop down menu under edit or delete accounts.
- Added support for theming and customization. Included are support for Interface, email, reports customization as well as productization with a custom name.
- Added support for shared quarantines on shared storage which allows messages to be accessed even when the node that processed them is offline.
- Implemented full cluster functionality for all components
- Improvements to Active Directory / LDAP including support for address verification of alias domain accounts, import of aliases from LDAP servers that use the mail attribute such as OpenLDAP, fix case sensitivity issue with Active Directory servers.
- Fixed MailScanner SQL config keyword issue.
- Fixed duplicates of account listings when user belonged to more than one domain
- Fixed various issues that caused quarantine reports not to be sent to some user accounts.
- Fixed auto user logout when they delete their account.
- Improve the predicate matching system for authorization of actions.
- Fixed previewing of embedded images in emails.
- Fixed the searching of archives when did not display the actual messages found.
- Fixed signature processing on the nodes after configuration in the interface.
- Added experimental PDF reporting command with theme support
- Added experimental Quarantine reporting command with theme support
- Fix to various cronjobs like the ones pruning database tables.
- Disabled NJABL
- Updated translations

4.10.6 2.0.0

- Initial release

CHAPTER

FIVE

USER GUIDE

5.1 Signing In and Signing Out

5.1.1 Signing In

To signin in to Baruwa, you enter your `username` and `password` and select the language to use if the auto detected language is not the one you prefer to use.

If you are signing in using external authentication such as your AD/LDAP or IMAP credentials then you need to provide the full username with the domain part included.

Your session will automatically timeout after 8 hours and you will have to login again.

5.1.2 Signing Out

To sign out click the `Logout` link on the top right corner of your screen.

Your session will automatically timeout after 8 hours and you will have to login again.

5.2 Changing Your Password

You can change your password if your account is setup to use local (internal) authentication.

If your account uses external authentication then use the system hosting your account credentials to change them.

5.2.1 Change a Known Password

While logged in.

1. Go to the `Account` page.
2. Click `Change Password`.
3. Enter your new password twice then your old password.
4. Click the `Change Password` button.

5.2.2 Reset a Forgotten Password

At the login page.

1. Click `Forgotten password ?`
2. Enter your email address, Click the `Reset my password` Button
3. Check your email, follow the instructions in the email

5.3 Personalizing Your Account

You can personalize various settings of your account using the account page.

5.3.1 Account names

You can change the First and Last name used to address you in any correspondence from Baruwa.

1. Go to the `Account` page
2. Click `Update Account`
3. Enter `First name` and `Last name`
4. Click the `Update account` button

5.3.2 Change Your Default Time Zone

By default your account uses the time zone setup for your domain by your domain administrator.

This option allows you change the time zone, All times in the Baruwa interface will be displayed in this time zone.

1. Go to the the `Account` page
2. Click `Update Account`
3. In the `Timezone` drop-down menu select the time zone you want to use.
4. Click the `Update account` button

5.3.3 Enable or Disable reports

You can enable or disable reports using this option. Reports include your `daily quarantine report` and a `monthly usage report`.

1. Go to the the `Account` page
2. Click `Update Account`
3. In the `Send reports` checkbox, select to enable, deselect to disable
4. Click the `Update account` button

5.3.4 Enable or Disable Spam Checks

You can choose to enable or disable Spam checks on messages destined to your account.

1. Go to the the `Account` page
2. Click `Update Account`
3. In the `Enable spam checks` checkbox, select to enable, deselect to disable
4. Click the `Update account` button

5.3.5 Customize Spam scores

You can customize the scores at which messages are determined to be either Spam or definite Spam.

Note:

- The `Spam High score` must be higher than the `Spam low score`
- Setting `0.0` makes Baruwa use the Domain or system defaults.

---

1. Go to the the `Account` page
2. Click `Update Account`
3. In the `Spam low score` or `Spam high score` input, enter the score
4. Click the `Update account` button

5.3.6 Add Email signatures/Disclaimers

Baruwa can manage email signatures / disclaimers that are added to messages that are sent outbound through it. Both HTML and Text signatures are supported. HTML signatures support a single embedded image.

A WYSIWG Editor is used to setup the HTML signatures and it allows you to upload images that you can embed in your HTML signature.

1. Go to the the `Account` page
2. Click `Add signature`
3. Select `Signature type` from the drop down
4. Enter signature content
5. Ensure the `Enabled` checkbox is checked
6. Click the `Add signature` button

5.4 Messages

5.4.1 Most Recent Messages

When you login the default view you see is the most recent messages for your account. By default the latest 50 messages are shown.

If you want to change the number of recent messages displayed you can use the drop down select `Show: items per page` to do that.

The selected number will be displayed during your current session, when you logout the number will reset to 50.

5.4.2 Full message listing

If you want to see more then the most recent messages you should,

1. Mouse over `Messages`
2. Click `Full message list`
3. Use the pagination links to see more messages.

5.4.3 Quarantine

If you want to see only quarantined messages,

1. Mouse over `Messages`
2. Click `Quarantine`
3. Use the pagination links to see more messages.

You can carry out message operations on several messages from within this view. Refer to *Bulk Message Operations* for details.

5.4.4 Archived messages

If you want to see older archived messages,

1. Mouse over `Messages`
2. Click `Archive`
3. Use the pagination links to see more messages.

5.4.5 Message Details

If you want to see the details of any specific message click the link to the message.

The following information is available.

- Message ID
- From Address
- To Address
- Subject
- Received date and time (Displayed in your timezone)
- Received by server (The server that received the message)
- Received from (The server that sent the message)
- Received via (Servers that processed this message, includes country information)
- Size

- Message headers
- Quarantined
- Virus infected
- Prohibited file
- Other infection
- Spam checks information (Spam check results and rules used to make determination)
- Delivery information (Status of mail delivery to final destination)

If the message is quarantined you are able to `preview`, `release`, `learn` or `delete` the message. Refer to *Message operations* on how to do this.

You are also able to add the sender to an authorized or banned sender list from with this view using `email address`, `domain name` or `IP address`. Refer to *To add the sender to a list* on how to do this.

5.4.6 Message operations

The Baruwa interface allows you to `preview`, `release`, `learn` or `delete` quarantined messages and authorize or ban senders of messages using `email address`, `domain name` or `IP address`.

Previewing a quarantined message

To preview a quarantined message,

1. Click the message link
2. Click `Preview message`
3. Click `Attachments to download any attachments`
4. Click `Display images` to display any remote images (This is not advisable)

Releasing a quarantined message

To release a quarantined message,

1. Click the message link
2. Click `Release message`
3. Check `Release` checkbox
4. Enter `Alt recipients` if you want to send the message to another email address
5. Click the `Submit` Button

Bayesian learning a message

You can update the Bayes system by teaching it if a message is Spam or Not Spam.

1. Click the message link
2. Go to the bottom of the page
3. Check `Bayesian Learn` checkbox

4. Select `Spam` or `Clean` from the drop down
5. Click the `Submit` Button

Deleting a quarantined message

You can delete a message from the quarantine.

1. Click the message link
2. Go to the bottom of the page
3. Check `Delete` checkbox
4. Click the `Submit` Button

5.4.7 To add the sender to a list

1. Click `Add sender to list`
2. Select the type of list you want to add them to using the `List type` drop down
3. Check `Add to aliases as well` if you want it to apply to your aliases as well
4. Check `Use IP address` to use the `IP address`
5. Check `Use Domain` to list the whole domain
6. Click the `Add to list` button

5.4.8 Bulk Message Operations

It is possible to carry out message operations (`release`, `learn` or `delete`) on multiple messages at ago.

To do this.

1. Select the messages using the check box
2. Select the operations (`release`, `learn` or `delete`) at the top
3. Click the `Process` button
4. View the operations results

5.4.9 Filters

Message filters are available on the *Full message listing*, *Quarantine* and *Archived messages* pages.

Refer to *Manage Filters* on how to manage these filters.

5.5 Approved and Banned Sender Lists

Baruwa supports the use of Approved and Banned sender lists.

Addresses on your approved sender list will skip all spam checks allowing their emails to always get delivered to you.

Addresses on your banned sender list will have their messages to you rejected.

5.5.1 Adding addresses to lists

1. Mouse over `Lists`
2. Click `Add to List`
3. Enter the address can be an `Email Address`, `Domain Name` or `IP address`
4. Select the list type from the `List type` drop down menu
5. Check `Add to aliases as well` if you want it added to your aliases
6. Click the `Add to list` button

5.5.2 Deleting addresses from lists

1. Mouse over `Lists`
2. Click either `Approved senders` or `Banned senders`
3. Find the address
4. Click the red `x` under the action column

5.6 Reports

The reports view allows you to run a set of predefined reports. The following reports are available.

5.6.1 Available reports

- Top Senders by Quantity
- Top Senders by Volume
- Top Sender Domains by Quantity
- Top Sender Domains by Volume
- Spam Score Distribution
- Top Mail hosts
- Top Recipients by Quantity
- Top Recipients by Volume
- Message Totals

You can use `filters` to filter the results available in your report. These `filters` can be saved for later reuse. Refer to *Manage Filters* for details.

Reports are exportable, and can be exported as PDF or CSV. Refer to *Export report* for details on how to export a report.

5.6.2 Export report

Export report to PDF

1. Click report link
2. Click `Download PDF`

Export report to CSV

1. Click report link
2. Click `Download CSV`

5.6.3 Manage Filters

A filter rule consists of one message property and one condition. If the message matches the property and condition it is selected.

Filter properties

The following properties are available to filter messages on.

- Message ID
- Message size
- From Address
- From Domain
- To Address
- To Domain
- Subject
- Received from
- Was scanned
- Is Spam
- Is Definite spam
- Is RBL listed
- Is approved sender
- Is banned sender
- Spam score
- Spam report
- Is virus infected
- Is name infected
- Is other infected
- Date

- Time
- Headers
- Is quarantined
- Processed by host

Filter conditions

Different properties support different conditions. The conditions supported by a specific property will automatically be selected when you select the property.

The following conditions are available.

- is equal to
- is not equal to
- is greater than
- is less than
- contains
- does not contain
- matches regex
- does not match regex
- is null
- is not null
- is true
- is false

Setting Up Filter Rules

1. Go to the `Reports` page Or within the *Full message listing*, *Quarantine* and *Archived messages* pages.
2. Select the property from the first drop down menu
3. Select the condition
4. Enter condition text if the condition requires one
5. Click `Add filter`

Saving Filter Rules

1. Go to the `Reports` page
2. Select the filter rule under `Active Filter(s)`
3. Click `Save`

Deleting a saved Filter Rule

1. Go to the `Reports` page
2. Select the filter rule under `Saved Filter(s)`
3. Click `Delete`

5.7 Mail queues

Messages that are yet to be processed are kept in the `inbound queue`, messages that have been processed but are yet to be delivered are kept in the `outbound queue`.

The status of both the `inbound` and `outbound` mail queues is provided. The following actions can be performed on messages that are in the queues:

- Delivery
- Bounce
- Hold
- Delete
- Preview

You can access these mail queues by clicking the numbers next to `In:` and `Out:` at the top of your screen

5.7.1 Processing queued messages

Deliver a message in the outbound queue

Delivery only applies to messages that have already been processed by Baruwa, that is why only messages in the outbound queue can be delivered.

To deliver a message:

1. Click the number next to `Out:` at the top of your screen
2. Select the message
3. Scroll to the bottom of the screen
4. Select `Deliver`
5. Click the `Process` button

Note: Delivery is only possible if the destination server is up and accepting mail.

Delete a queued message

1. Click the number next to `In:` or `Out:` at the top of your screen
2. Select the message
3. Scroll to the bottom of the screen
4. Select `Delete`

5. Click the `Process` button

Bounce a queued message

1. Click the number next to `In:` or `Out:` at the top of your screen
2. Select the message
3. Scroll to the bottom of the screen
4. Select `Bounce`
5. Click the `Process` button

Hold a queued message

1. Click the number next to `Out:` at the top of your screen
2. Select the message
3. Scroll to the bottom of the screen
4. Select `Hold`
5. Click the `Process` button

Preview a queued message

1. Click the number next to `In:` or `Out:` at the top of your screen
2. Select the message
3. Click `Preview message`

5.8 Baruwa Search Tips and Tricks

Baruwa supports many of the search tricks you use in popular web search engines.

5.8.1 Search with an exact phrase

To search for an exact phrase enclose the phrase in quotes `"Blocked message"`

5.8.2 Search for one or other

Use the pipe character `|` to separate the phrases `"Barrack Obama" | "Mike Tyson"`

5.8.3 Search using a wildcard

Use the star character `*` For example `boy*` will match `boy`, `boyfriend`

5.8.4 Search using the negate operator

`shaken !stirred` or `shaken -stirred` will match phrases with shaken but not `shaken stirred`

5.8.5 Search using grouping

`(red | green | blue) car` will match `red car`, `green car` or `blue car`

5.8.6 Search Specific fields

Note: It is also possible to limit your search to specific fields, the field operators will be provided later.

CHAPTER

SIX

SUPPORT

6.1 Free support

Email only support is available via the Enterprise edition support email address `enterprise-support (AT) baruwa.com`.

A mailing list also exists where you can discuss Enterprise edition related issues as well as ask for help and advise from fellow subscribers. The developers subscribe to and actively monitor this list.

6.2 Paid for support

Paid for support and consultancy services are available. All hands on or On device support which includes troubleshooting, investigation and resolution is only provided under paid for support.

Prepayment of an initial support fee is required before any hands on support tasks are carried out.

To request for paid support or to obtain our rate card, please email `enterprise (AT) baruwa.com`.