Bytemark Symbiosis

COLLABORATORS			
	TITLE : Bytemark Symbiosis		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Patrick J. Cherry, Steve Kemp, David Edwards, and James Carter	9th April 2019	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
2009:1112	2009-11-12	Initial release.	PJC
2010:0427	2010-04-26	Renamed the project, and updated the documentation to match.	SKX
2012:0302	2012-03-02	Rewritten for the Squeeze release>	PJC
2012:0305	2012-03-05	Updated release notes for the Squeeze release.	PJC
2012:0420	2012-04-20	Added information about PostgreSQL backups, apache2 logging, and fixed the erroneous reference to exim_rewrite_scan in the release notes.	PJC
2014:1118	2014-11-18	Updated for Wheezy release.	JFC
2015:0909	2016-02-29	Updated for Jessie release.	PJC
2018:0621	2018-06-21	Updated for Stretch release.	AL

Contents

Chapter 1

What's new since the last release

The current release is based on Debian 9.0, code-name Stretch. Since the last release of Symbiosis, the following features have been implemented.

- **symbiosis-httpd-configure** now includes a **--diff-only** option, allowing potential changes between Apache configurations to be compared without disrupting active domains.
- #12 SSL hooks have been implemented in `/etc/symbiosis/ssl-hooks.d`. These are triggered when SSL certificates are updated, meaning other running services can be notified to act accordingly. As an example, Apache configurations will now be regenerated and the Apache service will be reloaded automatically when new SSL certificates are added using **symbiosis-ssl**.
- **symbiosis-monit** checks are now launched using systemd timers, rather than as cron tasks.
- Skeleton directories are automatically created for new domains when a top level directory is created within /**srv** and owned by the **admin** user.
- The default Apache logger, written in Ruby, has been replaced with a new Golang version for improved performance.
- A new "message of the day" has been added, with an appropriate Stretch theme!

Additionally, a number of bugs have been fixed, including:

- #51 Website stats are now disabled by default. They can be enabled with the creation of a **config/stats** file on a per-website basis.
- #76 The **50-reject-www-data** firewall rule has been removed from the default firewall configuration.
- #107 The `.well-known` directory path for domains is now excluded from rewrites by default, meaning it should always be accessible for verification with Let's Encrypt.
- #113 **symbiosis-httpd-logger** no longer uses the **--sync** flag by default, improving performance for servers which host a large amount of websites.
- #114 Exim no longer returns a **Warning: purging the environment** message when it's restarted.
- #115 The **admin** user is now added to the **www-data** group by default on
- #120 A number of frequently used packages are now installed by default.

- #123 PHP variables **post_max_size** and **upload_max_filesize** now default to 64MB each. These can be overridden in the /**etc**/**php**/**7.0**/**apache2**/**conf.d**/**00-symbiosis-httpd.ini** configuration file.
- #126 A new MySQL user, **admin@localhost**, is created for new installations of Symbiosis Stretch, to be used with phpMyAdmin following the inclusion of MariaDB which uses unix socket auth for **root@localhost** by default. Servers which have been upgraded from Symbiosis Jessie are unaffected.

Thank you to all those who reported those issues.

Chapter 2

Installing and administering Symbiosis

Symbiosis will install well on a freshly-installed Debian 9.0 system. Currently it is only available for *i386* and *amd64* architectures, running on the Linux kernel.

It is designed to be as friendly as possible for beginners, whilst maintaining flexibility for more experienced systems administrators. Later in this chapter we'll spell out a few basics to bear in mind when working with a system running Symbiosis.

Installing Symbiosis running on Debian 9.0 (Stretch)

Installing on a fresh Stretch system is relatively simple. First, add the Bytemark package signing key:

```
curl -sSL https://secure.bytemark.co.uk/key/repositories-2014.key | sudo apt-key  ↩
    add -
```

Second, add the following to `/etc/apt/sources.list.d/symbiosis.list`:

```
#
# Bytemark Symbiosis Packages
#
deb     http://symbiosis.bytemark.co.uk/stretch/ ./
deb-src http://symbiosis.bytemark.co.uk/stretch/ ./
```

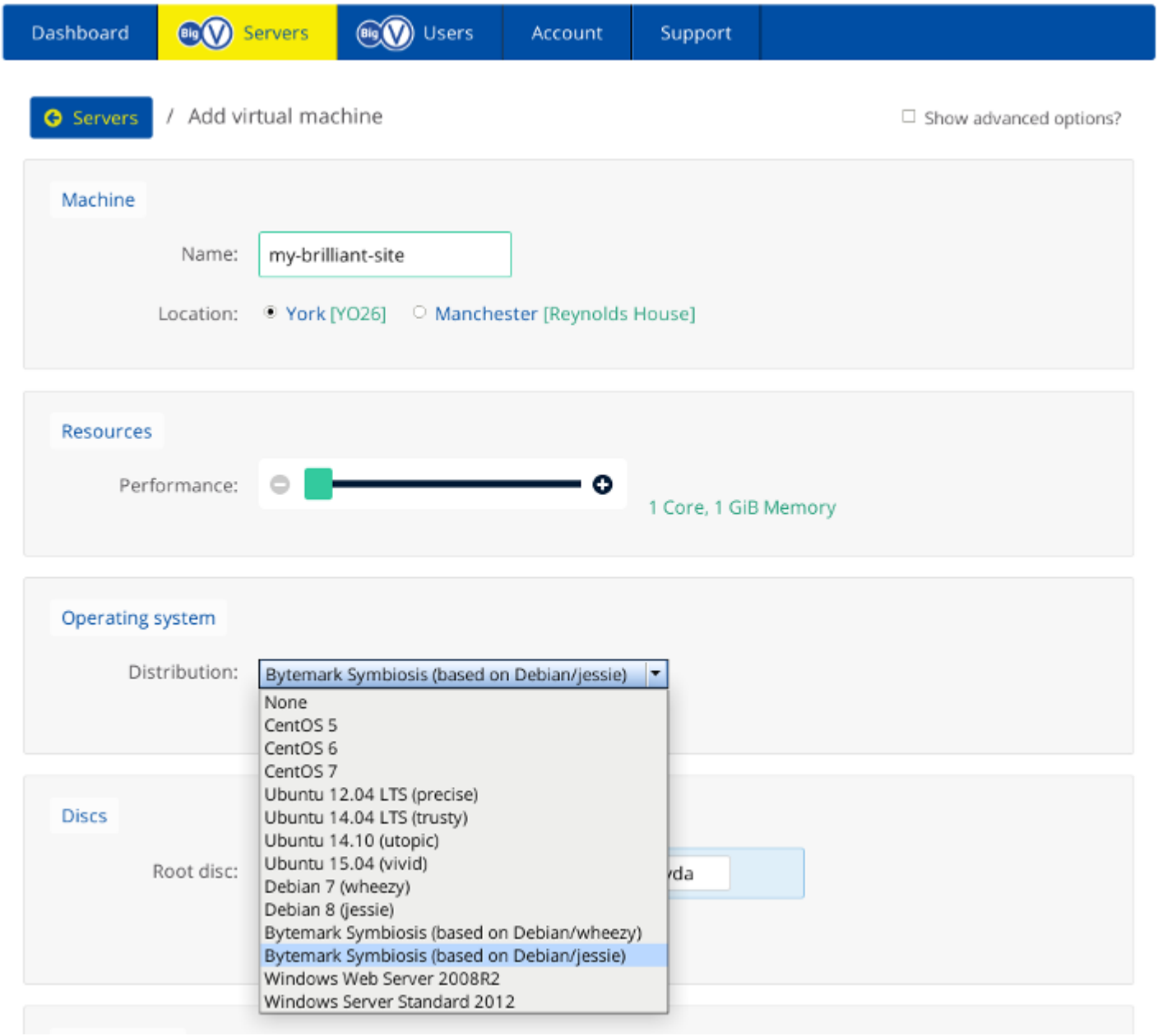
Once that is in your sources, run:

```
apt-get update
apt-get install --install-recommends bytemark-symbiosis
```

At the end of this process, you should have a fully functioning Symbiosis system with all of the features documented here available to you for use.

Installing Symbiosis Stretch using the Cloud Server control panel

Users of Bytemark's Cloud Servers can get a fresh install of Symbiosis by simply selecting the image at VM install time. The panel is accessed at https://panel.bytemark.co.uk. Once logged in, using the `Add a cloud server` button, a machine can be installed within a few seconds. Select the Stretch Symbiosis image as per the screenshot below.



Upgrading from Symbiosis 8 (Jessie) to Symbiosis 9 (Stretch)

Debian have comprehensive release notes, of which chapter 4 covers the recommended upgrade procedure. We have provided a shorter version for this, which is immediately below:

The first thing to do is make sure that you have backups. These should be kept in /var/backups/localhost, and they should be up to date.

Note
Any modifications you may have made to Symbiosis scripts will likely be lost during the upgrade, so you should be prepared to reapply these changes after the upgrade.

Next, alter /etc/apt/sources.list. Change all instances of the word `jessie` to `stretch`. If you have backports, you can remove them, and any entries for Jessie LTS should also be removed. Then change the Symbiosis repository lines to match those shown in the previous section.

You can then proceed with the upgrade by running:

```
apt-get update
apt-get dist-upgrade
```

Following the upgrade, to use PHP7 you will need to disable and enable the appropriate Apache modules:

```
a2dismod php5
a2enmod php7.0
systemctl restart apache2
```

You can swap back at any time by disabling the php7.0 module and enabling the php5 module instead.

The updated version of Roundcube relies on PHP7.0 by default. If you would like to continue using PHP5, you will need to install the `php-net-idna2` package by running:

```
apt-get install php-net-idna2
```

As this is an upgrade for **all** the software on the system, a large number of questions may be asked about configuration files during the upgrade. Some of these will relate to packages Symbiosis has installed as dependencies, and the answers to these questions are given below.

Questions asked during the upgrade

Q: *Configuring roundcube-core: Configure database for roundcube with dbconfig-common?*

A: Yes

Q: *Configuring roundcube-core: Database type to be used by roundcube*

A: mysql

Q: *Configuring roundcube-core: Host running the server for roundcube*

A: localhost

Q: *Configuring roundcube-core: Password of the database's administrative user*

A: Enter your MySQL root password

Q: *Configuring roundcube-core: MySQL application password for roundcube*

A: User preference (Leave blank to generate a random password)

Q: *Configuring phpmyadmin: Configure database for phpmyadmin with dbconfig-common?*

A: Yes

Q: *Configuring phpmyadmin: Host running the MySQL server for phpmyadmin*

A: localhost

Q: *Configuring phpmyadmin: Web server to reconfigure automatically*

A: apache2

Q: *Configuring libc6:amd64: Restart services during package upgrades without asking?*

A: Yes

Q: *Configuration file `/etc/sysctl.conf` modified since installation*

A: Install the package maintainer's version

Q: *Configuring openssh-server*

A: Install the package maintainer's version

Q: *Configuring grub-pc*

A: Keep the local version currently installed

Q: *Configuration file `/etc/ntp.conf` modified since installation*

A: Keep your currently-installed version

Q: *Configuration file `/etc/phpmyadmin/config.inc.php` modified since installation*

A: Install the package maintainer's version

Q: *Configuring phpmyadmin: Perform upgrade on database for phpmyadmin with dbconfig-common?*

A: Yes

Q: *Configuring unattended-upgrades*

A: Install the package maintainer's version

That should be everything; you may have been asked other questions if you have installed extra packages on your system - answer them as you see fit.

Release notes

This release of Symbiosis includes a number of new features that are summarised in Chapter 1.

Apache HTTPd upgrade from 2.4.10 to 2.4.25

Apache has undergone a significant version change in Stretch. If you've made any custom Apache config changes, you may need to look at the docs. As of 2.4.17, HTTP2 is supported. Further information is available here.

PHP upgrade from 5.6 to 7.0

The version of PHP included with Symbiosis Stretch has been upgraded to 7.0 from 5.6. This new version is enabled by default on both new installations, and following a dist-upgrade. Further information on the changes between 5.6 and 7.0 is available here.

Packages installed by Symbiosis

Each component that makes up Symbiosis is separately packaged as follows. Each package can be installed individually if needed.

bytemark-symbiosis Meta-package that pulls in the core requirements for a Symbiosis system, and as well as recommending all packages needed for a complete Symbiosis system.

symbiosis-backup Organises and configures backup2l to backup vital parts of the system, and rsync them to a remote location.

symbiosis-common Contains the core libraries that Symbiosis uses to operate.

symbiosis-dns Adds automatic DNS generation and upload to the system. Ties in with the Bytemark DNS service.

symbiosis-email Configures Exim and Dovecot for use with Symbiosis.

symbiosis-email-activesync - Provides email access using the Microsoft Exchange ActiveSync protocol

symbiosis-firewall Maintains the iptables and ip6tables firewalls, as well as providing automatic blacklisting and whitelisting.

symbiosis-ftpd Configures pure-ftpd to work with Symbiosis.

symbiosis-httpd Configures the Apache web server.

symbiosis-key Adds the Bytemark Symbiosis key to apt.

symbiosis-monit Provides service monitoring.

symbiosis-mysql Brings in MariaDB version 10.1, and configures it to bind to all interfaces, not just localhost, for remote access. MariaDB is a fork of MySQL, and is designed to be a drop-in replacement.

symbiosis-pam Brings in two PAM dependencies to make the system more secure — one checks passwords and warns when they are weak, the other sets per-user temporary directories.

symbiosis-phpmyadmin Brings in phpMyAdmin, and configures it to use HTTP authentication.

symbiosis-webmail Adds webmail functionality, using Roundcube. Includes one additional package:

symbiosis-webmail-roundcube Provide webmail access to Symbiosis using Roundcube

symbiosis-xmpp Adds an XMPP server, which can be used to chat to people on the global XMPP network.

Systems administration and Symbiosis

Symbiosis is an attempt to encourage best practice at all times in systems administration, whilst keeping things as simple as possible, and free of surprises. As a result there are a few general rules to bear in mind when tinkering with your system.

Use of `root`, and other users

As far as possible Symbiosis will discourage you from using `root` when logging in and configuring the system. This primarily applies to

- Anything in the `/srv/` directory
- The firewall configuration in `/etc/symbiosis/firewall`

For example, if a directory in `/srv` is owned by a system user or group, i.e. one with a UID/GID less than 1000, then it will not show up to various tasks, including, but not limited to,

- Email and FTP logins
- Cron tasks in `config/crontab`

- Apache logging to `public/logs/`
- Mail delivery to mailboxes.

In short, try not to use `root` if at all possible.

However it is perfectly possible to configure separate domains in `/srv/` to be owned by different users, as long as they are non-system users, i.e. ones with user IDs greater than 1000. All programs will respect these permissions.

Customising configurations

Lots of configuration on the system is automatically generated to make Symbiosis work as it does. In previous releases of Symbiosis this meant that files would get overwritten without notice. However as of the Squeeze release in February 2012 configuration files are handled more conservatively.

Two things to watch out for. If a configuration file has

```
# DO NOT EDIT THIS FILE - CHANGES WILL BE OVERWRITTEN
```

written in it, then there is a high chance that any changes will be overwritten. It has to be the exact wording and spacing above for overwriting to take place, so if that sentence is removed from the configuration then it **will not** get overwritten.

Similarly many files are generated from templates, for example DNS and apache snippets. These will now have a checksum at the bottom of the file.

```
# Checksum MD5 586732ff59e60115d0ec1c4905c72773
```

This checksum allows Symbiosis scripts to establish if the template used to generate the snippet has changed, if the data used in the generation has changed, or if the file itself has been edited. For example if an IP address is changed by editing `config/ip`, then that would allow the apache snippet for that domain can be updated, as can the DNS snippet.

This also means that sysadmins can edit the templates, and allow them to regenerate, or edit the snippets themselves safe in the knowledge that their changes will not get overwritten.

Other configuration styles

The Backup2l, Dovecot, and Exim configuration files are generated not with a template, but with a collection of snippets, which are joined and checked using a Makefile. This allows extra configuration snippets to be added in to the configuration.

If it is deemed necessary, sysadmins can add extra snippets to these configurations. The basic procedure is to read the configuration file, and decide where the extra directives need to go. This is made easier by the fact that through the configuration files comments are added showing where each part comes from.

```
# ------------------------------------------------------------------------
# /etc/exim4/symbiosis.d/10-acl/40-acl-check-mail/00-header
# ------------------------------------------------------------------------

# ACL that is used after the MAIL command
acl_check_mail:

# ------------------------------------------------------------------------
# /etc/exim4/symbiosis.d/10-acl/40-acl-check-mail/90-default
# ------------------------------------------------------------------------
```

```
# Allow anything not already denied to connect
  accept
```

In this example, if an extra directive were required in this ACL, then a file could be created in `/etc/exim4/symbiosis.d/10-acl/40-acl-check-mail/`, maybe with the filename `10-do-stuff`. To create the new configuration, we'd then need to run make in `/etc/exim4/`. This would regenerate `/etc/exim4/exim4.conf`, and perform a basic syntax check. If happy with the new configuration, then exim4 could be restarted.

The equivalent Dovecot configuration is in `/etc/dovecot/symbiosis.d/` which generates `/etc/dovecot/dovecot.conf`. The Backup2l configuration is in `/etc/symbiosis/backup.d/conf.d/`, which generates `/etc/symbiosis/backup.d/backup2l.conf`.

Chapter 3

Website Configuration

This is a detailed break down of all the configuration options and files available when configuring website hosting for a domain.

Throughout this chapter, as with the rest of this documentation, the domain `my-brilliant-site.com` is used as an example.

All configuration for the domain `my-brilliant-site.com` will be performed inside the `/srv/my-brilliant-site.com/` directory.

The Bytemark Symbiosis project uses the popular Apache HTTPD software for serving your websites, and this comes complete with PHP7 along with many of the most popular PHP extensions.

Getting started

All the files required for a website for the domain **my-brilliant-site.com** are kept in `/srv/my-brilliant-site.com/public/htdocs/`.

- If this directory does not exist, a **404 Not Found** error will be returned.
- If this directory exists, but is empty, then a default page is shown.
- The index file can be written in *HTML* or *PHP*, and should be called `index.html` or `index.php` respectively.
- Once this directory is present, both http://my-brilliant-site.com and http://www.my-brilliant-site.com will show the same content, i.e. there is no need to name the site with a **www** prefix.
- If different content is required for http://www.my-brilliant-site.com then that should be put in `/srv/www.my-brilliant-site.com/public/htdocs/`.

CGI scripts

If you wish to use CGI scripts for your domain, then simply copy them to a directory named `cgi-bin/` beneath the `public/` directory. They must all be marked as executable. This means setting the permissions to **755**. In FileZilla, right click the file and select File Permissions... from the menu. The file should have **Execute** set for the owner, group, and public permissions.

For example, for **my-brilliant-site.com** the scripts would live in `/srv/my-brilliant-site.com/public/cgi-bin/`.

Any **executable** files in that directory will now be treated as CGI scripts for your domain. For example if you created the file `/srv/my-brilliant-site.com/public/cgi-bin/test.cgi` This would be referred to as: http://my-brilliant-site.com/cgi-bin/test.cgi

Statistics

Each hosted website can have visitor statistics automatically generated and accessible at http://my-brilliant-site.com/-stats/. These statistics will be updated once per day, and the raw access logs will be made available as `/srv/my-brilliant-site.com/public/logs/`.

As of the Stretch release of Symbiosis, these daily statistics are disabled by default. If you wish to continue using them, you'll need to enable them explicitly with the creation of a `stats` file in the website's configuration directory. For example, for **my-brilliant-site.com**, the `stats` file should exist at `/srv/my-brilliant-site.com/config/stats`.

If you had previously disabled stats with the creation of a file `config/no-stats`, this should be removed automatically following a dist-upgrade.

It is also possible to customise the statistics generated by editing the file `config/webalizer.conf`. This file is documented at the Webalizer project website.

If there are many sites on the same machine, then it is possible to customise all the sites' Webalizer configurations by editing the template that is available at `/etc/symbiosis/apache.d/webalizer.conf.erb`. Configuration files will be updated when the statistics are next generated, but only for sites whose configurations either do not exist, or have not been edited by hand.

Testing new websites

You can view new websites before any DNS changes are made.

For example, if the virtual machine **example.default.bytemark.uk0.bigv.io** is hosting **www.my-brilliant-site.com**, i.e. the directory `/srv/my-brilliant-site.com/public/htdocs/` has been created, then the website can immediately be viewed at http://my-brilliant-site.com.testing.example.default.bytemark.uk0.bigv.io.

There are some important things to note though: - There is no **www** part added to the domain name — it is just the directory name prepended to **.testing.example.default.bytemark.uk0.bigv.io**. - This testing alias isn't guaranteed to work in all cases, for complex site setups it might not work entirely as expected. - The testing alias only allows the testing of websites. Therefore FTP logins, email delivery, or checking is explicitly unsupported.

Displaying the same content under two domains

In this scenario, you have registered two domains for example **my-brilliant-site.com** and **my-brilliant-site.co.uk**, but you want the same content to be served at both addresses. There is no need to create two separate directory structures, you can just set up one directory structure and then create a soft link (aka symbolic link or symlink) to the second.

1. Once the my-brilliant-site.com directory structure has been completed, log on to your machine as admin over SSH.

2. Run the command **ln -s /srv/my-brilliant-site.com /srv/my-brilliant-site.co.uk**

This creats a symbolic link of `my-brilliant-site.co.uk` pointing at `my-brilliant-site.com`. Now browsing to my-brilliant-site.co.uk will show the same content that appears at my-brilliant-site.com.

Redirecting to the preferred website domain

If a document tree were created in `/srv/my-brilliant-site.com/public/` then that site would be available under two hostnames:

- http://my-brilliant-site.com/
- http://www.my-brilliant-site.com/

There are people who prefer to use only a single name, and to automatically redirect visitors using the *wrong* name to using the preferred name. This can easily be achieved by using Apache's mod_rewrite facility.

If you prefer all visitors see the www-based site you could create the file `/srv/my-brilliant-site.com/public/htdocs/.htaccess` with the following contents:

```
RewriteEngine on
RewriteCond %{HTTP_HOST} !^www.*$ [NC]
RewriteRule ^(.*)$ http://www.%{HTTP_HOST}/$1 [R=301,L]
```

This examines each incoming request, and if the hostname doesn't begin with "www." then it is prepended to the request and a redirect is issued.

Custom Apache configuration

It is perfectly possible to alter the way Symbiosis configures Apache, either for an individual domain, or for all domains hosted on the server.

Symbiosis hosts sites on a server in one of two ways, based on the IP address that site has configured. If it uses one of the server's primary IP addresses, then it is assumed that the site is hosted using the "mass-hosting" configuration. If the site has a secondary IP assigned then Symbiosis generates an individual snippet for that site, and Apache is configured to use that snippet when dealing with HTTP requests for that domain. Both configuration techniques are configured using a template, which allows the server's administrator to fiddle with, and tweak the configuration.

In `/etc/symbiosis/apache.d/` there are a number of templates that are used to generate configuration snippets for both the mass-hosting, as well as individual sites.

Logging

By default, access requests for each site on a machine will go to `public/logs/access.log`. If the site has SSL enabled, the request logs will go to `public/logs/ssl_access.log`. These logs get rotated once a day, and compressed after two days.

The error logs for a site will go to one of two places, depending on how the site is configured. If the site has its own SSL certificate, or otherwise has its own IP address, then the error logs will go to `public/logs/error.log`, or

`public/logs/ssl_error.log`. Otherwise the error logs will go to `/var/log/apache2/zz-mass-hosting.error.log`.

Finally, if a request is received for a domain that is not present on the box, then it is logged to `zz-mass-hosting.access.log` if it received on the primary IP of the machine. If the request comes on any other IP then it is logged to `other_vhosts_access.log`. Both of these last two files are located in `/var/log/apache2` .

Web configuration layout

Here is an example configuration layout for the domain `my-brilliant-site.com`, all of which is contained under `/srv/my-brilliant-site.com/`.

`config/stats` If this file exists, statistics will be generated for this domain.

`config/ssl-only` If this file exists, traffic will be redirected to the SSL version of the website. This will also configure your site to use Strict Transport Security (HSTS). Ensure your website functions as expected over HTTPS **before** adding this file. Once HSTS is enabled, browsers are issued a forced redirect to HTTPS until HSTS expires (default age of 6 months) — even if the **ssl-only** file is removed!

`config/webalizer.conf` This is the Webalizer configuration file for this domain.

`public/cgi-bin/` This is the directory which may be used to hold CGI scripts for your domain.

`public/htdocs/` This is the directory from which content is served for the URLs http://my-brilliant-site.com/ *and* http://www.my-brilliant-site.com/. If this directory does not exist visitors will be shown an error page.

`public/htdocs/stats/` This directory will be automatically created, if it isn't already present, and updated with statistics referring to the number of visitors to your website.

`public/logs/access.log` This file contains the Apache webserver access log for the domain. It will be archived daily, and removed after 30 days.

`public/logs/ssl_access.log` This file contains the Apache webserver access log for the domain when accessed over SSL.

`public/logs/error.log` This file contains the Apache webserver error log for the domain, if the domain has been configured to run under its own IP address. It will be archived daily, and removed after 30 days. If the site does not have its own IP address, then errors are logged to `/var/log/apache2/zz-mass-hosting.error.log`.

`public/logs/ssl_error.log` This file contains the Apache webserver error log for the domain when accessed over SSL, if the domain has been configured with its own IP address.

Chapter 4

SSL configuration

Secure Sockets Layer is a technique used to encrypt communication between two machines on a network. It uses a system of public and private keys to encrypt and decrypt the connection — the public key is used by the sender to encrypt, and the private key is used by the receiver to decrypt. This protocol is used not only for transactions involving a web server and browser, but also by the email servers and their clients.

In addition to the public key encryption, there is a system of trust that validates that the certificate presented actually belongs to the server that is presenting it. This system involved having the certificate signed by a trusted authority. Web browsers and email clients tend to come with a selection of certificates from trusted authorities pre-installed, which allows them to verify a previously unseen certificate as valid.

SSL providers

Until recently, having a certificate signed by a trusted authority involved having varying degrees of identity checks made, and paying a fee. Vendors that are able to sell you a certificate include Rapid SSL and Comodo.

In December 2015, a new, free, automatic SSL certificate service started issuing certificates. This servers is called LetsEncrypt, and it is supported by a number of big names on the internet, including Facebook and Mozilla. This service has now been successfully providing domain-verified certificates for a few months, and is used by Symbiosis to generate trusted certificates for all the domains on a machine.

Symbiosis can also generate self-signed certificates on occasions where it has not been possible to use LetsEncrypt.

SSL provider configuration

Symbiosis has the idea of "providers" to generate SSL keys, requests, and certificates. By default Symbiosis will use the **LetsEncrypt** provider, but you can specify another provider, e.g. *SelfSigned" by putting the word "selfsigned" in `config/ssl-provider` for the domain in question.

If you wish to use LetsEncrypt, put "letsencrypt" in `config/ssl-provider`.

If you wish to disable automatic certificate provisioning entirely, you can put the word **false** into `config/ssl-provider`.

Generating certificates with symbiosis-ssl

Symbiosis uses a command ‘symbiosis-ssl` to manage domains’ certificates. It is run on a daily basis to check and replace certificates that are due to expire in the next 10 days, or are otherwise missing.

This command can also be used to verify sites' certificates.

```
$ symbiosis-ssl --verbose
* Examining certificates for app.my-brilliant-site.com
  Current SSL set 0: signed by /C=US/O=Let's Encrypt/CN=Let's Encrypt Authority X1 ↩
     , expires 2016-05-25 23:00:00 UTC
* Examining certificates for my-brilliant-site.com
  Current SSL set 0: signed by /C=US/O=Let's Encrypt/CN=Let's Encrypt Authority X1 ↩
     , expires 2016-05-25 10:32:00 UTC
* Examining certificates for example.default.bytemark.uk0.bigv.io
  Current SSL set 4: signed by /C=US/O=Let's Encrypt/CN=Let's Encrypt Authority X1 ↩
     , expires 2016-05-29 10:26:00 UTC
```

This command shows the current set in place for each site. A set is a collection of files, consisting of

- the private key (`ssl.key`),
- certificate request (`ssl.csr`),
- signed certificate (`ssl.crt`),
- intermediate bundle (if required) (`ssl.bundle`),
- a combined private key, and certificate, (and bundle) (`ssl.combined`).

Each set is kept in its own directory in `config/ssl/sets`, and the current set is symlinked to `config/sets/current`.

In the above output the host **example.default.bytemark.uk0.bigv.io** is currently using set 4. That means its directory layout will look like:

```
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/current -> /srv/example. ↩
   default.bytemark.uk0.bigv.io/config/ssl/sets/4
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/letsencrypt/account_key
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/0/ssl.combined
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/0/ssl.crt
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/0/ssl.csr
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/0/ssl.key
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/1 # (with a complete set ↩
    of ssl.key etc)
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/2 #
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/3 #
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/4/ssl.bundle
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/4/ssl.combined
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/4/ssl.crt
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/4/ssl.csr
/srv/example.default.bytemark.uk0.bigv.io/config/ssl/sets/4/ssl.key
```

Generating certificate signing requests

If you wish to use a different certificate than the one Symbiosis has generated for your domain, you can use the certificate signing request that was created for the certificate that should already be in place. You can then go to your provider with this request, and ask them to generate a certificate.

The CSR generated will be for the "bare" domain, with all aliases as Subject Alternative Names.

If you wish to inspect the CSR, you can run

```
openssl req -in ssl.csr -text -noout
```

Applying a third party certificate

To add a third party SSL certificate for a website, upload your ssl.crt, ssl.key, and ssl.bundle files from the certificate provider to your server. For a domain `example.com`, these should be located at:

```
/srv/domain.com/config/ssl.crt
/srv/domain.com/config/ssl.key
/srv/domain.com/config/ssl.bundle
```

Ensure ssl-provider is set to false, by running: echo "false" > /srv/domain.com/config/ssl-provider

Remove the SSL `current` symlink if present, as this will otherwise take precedence: rm /srv/domain.com/config/ssl/current

Reconfigure the Apache vhost to activate the certificate: sudo symbiosis-httpd-configure -v domain.com

SSL Configuration layout

Here is an example configuration layout for the domain `my-brilliant-site.com`, all of which is contained under `/srv/my-brilliant-site.com/`.

`config/ssl-provider` The SSL provider to be used for this domain. If this is set to **false** then no certificates will be generated for the domain.

`config/ssl.crt` Legacy SSL certificate.

`config/ssl.key` Legacy SSL key.

`config/ssl.bundle` Legacy SSL certificate chain or bundle.

`config/ssl.combined` Legacy combined SSL certificate, chain, and key.

`config/ssl/` This is the SSL configuration directory

`config/ssl/letsencrypt/` This is the LetsEncrypt configuration directory.

`config/ssl/sets/` This is the directory that contains all the various SSL certificate and key sets.

`config/ssl/current` This is a symbolic link to the current SSL certificate and key set

Self-signed SSL provider configuration

These files can go in the domain's `config/` directory, or in `/etc/symbiosis/`, if you want to set host-wide defaults.

`ssl/selfsigned/rsa_key_size` The size of RSA key generated for both the SSL certificates, as well as the account. Defaults to 2048.

`ssl/selfsigned/lifetime` The length of time in days the certificate should be valid for. Defaults to 365.

LetsEncrypt SSL provider configuration

These files can go in the domain's `config/` directory, or in `/etc/symbiosis/` if you want to set host-wide defaults.

`ssl/letsencrypt/rsa_key_size` The size of RSA key generated for both the SSL certificates, as well as the account. Defaults to 2048.

`ssl/letsencrypt/email` The email address associated with the account. Defaults to root@example.default.bytemark where example.default.bytemark.uk0.bigv.io is the hostname of the machine.

`ssl/letsencrypt/endpoint` The LetsEncrypt endpoint to use. Defaults to https://acme-v01.api.letsencrypt.org/-directory.

`ssl/letsencrypt/docroot` The document root for this domain. Defaults to `/srv/domain.com/public/htdocs`. This is required for the LetsEncrypt domain verification to take place.

`ssl/letsencrypt/account_key` The private RSA key for this LetsEncrypt account. A new one is generated if not present.

Chapter 5

Email Configuration

This is a detailed break-down of all the configuration options and files available when configuring how email is handled for a domain. Throughout this chapter, the domain `my-brilliant-site.com` is used as an example. Thus all the configuration for `my-brilliant-site.com` will be inside the `/srv/my-brilliant-site.com/` directory.

Port Configuration

The mail servers have been set up with standard port assignments as follows. These are all the standard ports for the protocols.

Service	Port	Encryption
POP3	110	TLS (using STARTTLS)
IMAP	143	TLS (using STARTTLS)
SMTP	25 or 587	TLS (using STARTTLS)
POP3	995	TLS (on connect)
IMAP	993	TLS (on connect)
SMTP	465	TLS (on connect)
Sieve	4190	TLS (using STARTTLS)

Accepting email for a domain

In order for a domain to be configured to accept email, one of two things must be present. Either the domain must have a `mailboxes/` directory present, or one of the files `config/default_forward` or `config/aliases` must be present.

For example, if the domain **my-brilliant-site.com** would like to host mail normally, i.e. one mailbox per user hosted on the same machine, then the directory `/srv/my-brilliant-site.com/mailboxes/` should be created. Then in there, one directory per user should be created. If **bob@my-brilliant-site.com** would like to receive mail, then `/srv/my-brilliant-site.com/mailboxes/bob/` should be created.

Once this directory has been created, this mailbox can be accessed via IMAP/POP3, or Roundcube webmail. For example, the mailboxes for **mybrilliant-site.com** would be accessible at **mybrilliant-site.com/webmail**.

Assuming that this is the only directory inside `/srv/my-brilliant-site.com/mailboxes/` then only mail addressed to **bob@my-brilliant-site.com** will be accepted. Any other mail addressed to **my-brilliant-site.com** will be rejected.

If you would like to accept all mail for **my-brilliant-site.com**, regardless of who it is addressed to, then create the file `/srv/my-brilliant-site.com/config/default_forward`. The contents of this file should be a single email address, or a comma-separated list of email addresses. For example, to forward all mail to **bob@my-brilliant-site.com**, unless the recipient's mailbox already exists, then `/srv/my-brilliant-site.com/config/default_forward` should contain `bob@my-brilliant-site.com`.

If you would like the domain **nomail.my-brilliant-site.com** not to receive any mail at all, then remove the directory `/srv/nomail.my-brilliant-site.com/mailboxes/` and ensure that the file `/srv/nomail.my-brilliant-site.com/config/default_forward` does not exist.

Email for Unix users.

Before you start this section

1. Both a unix user and a `normal` Symbiosis email user can be set up to receive email to the same address. The `normal` user will always take precedence over the unix user and have their mail delivered to their inbox first, so take care when using this feature!

A new feature in this release is the ability to have unix users with email accounts based in their home directories. These will receive emails for the host name of the machine, which you can find out by running `hostname` on the command line. The result of this will display the domain in the email address the system users will get, eg, the part after the @. The other half will be dictated by their username, eg, "admin" or "my-user".

To start with, we create a .password file in, eg, /home/my-user/. Initially this can contain the password in plain text.

Once the password file is in place, the new user will be able to login and collect email. Logins over SMTP, IMAP, and POP3 will all work identically to a normal email user, with the same ports and SSL/TLS requirements. The principal difference is the username is just their bare username, without a domain. E.g. for a user **clare**, her login for SMTP/IMAP/POP3 etc, is just **clare**.

Unix users' Maildir directories will reside in /home/my-user/Maildir by default. This allows these users to use system mail readers such as mutt in order to read and send email, obviating the need to use IMAP and SMTP.

These users are also able to control the following files:

- .forward - to control forwarding of emails.
- .vacation - to set a vacation (holiday) message.
- .sieve - to set up Sieve filters.

Password files

The password for a mailbox should be set by the contents of a file named `password` inside a user's mailbox directory. The contents of this file may be in plain text, or encrypted. If plain text is used, the system will automatically encrypt the password.

To encrypt all email passwords on the system, you can run

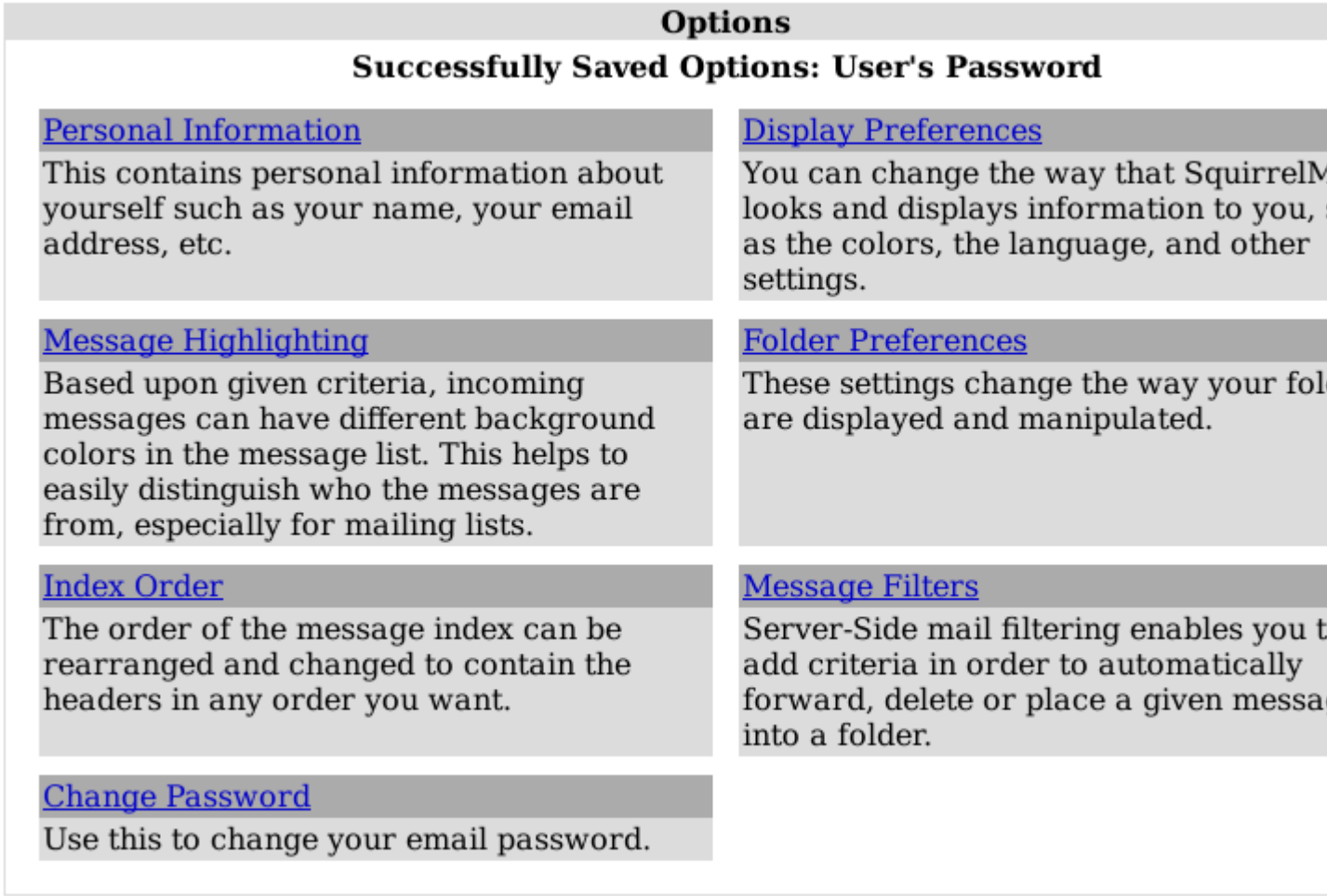
```
`symbiosis-email-encrypt-passwords --verbose`
```

The **--verbose** flag is there to provide more output.

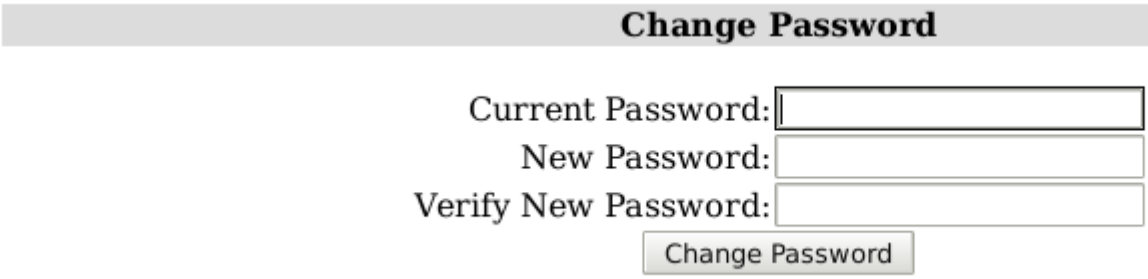
Allowing users to change their own password

Users are able to change their own passwords through the webmail system, which by default is Roundcube.

To change a password in Roundcube, log in and select Options. This will being up the Options page :



From there, select Change Password, where you must enter your current password, and enter a new one. Passwords which are ether too weak or too short will be rejected by the system.



Suffixes

All email addresses can be used with a suffix. This allows people to filter their email by the To: address. The separator between the local part and suffix is the + sign.

For example, Bob signs up to a shopping site at http://example.com. He might use bob+example@my-brilliant-site.com his email address when signing up, such that he can filter all email from that shop.

Enforcing mailbox size with quotas

Symbiosis can enforce users' mailbox size with quotas. This will prevent mail from being delivered to a user if their mailbox grows too large.

A default quota for each individual mailboxes in a domain can be specified in `config/mailbox-quota`. A per-mailbox quota can be defined in a file named `quota` which resides in a user's mailbox directory.

These files both have the same format, which is just a number of bytes over which mail should not be delivered. This number can have a suffix of `k`, `M`, or, `G` which represent kilobytes, megabytes, and gigabytes, or `ki`, `Mi`, or `Gi` to represent kibibytes, mebibytes, and gibibytes, respectively.

For example, to limit the size of each mailbox for the domain **my-brilliant-site.com** to 200MB, i.e. 200,000,000 bytes, put `200M` in `/srv/my-brilliant-site.com/config/mailbox-quota`.

To grant **bob@my-brilliant-site.com** a 1GiB quota, i.e. 1,073,741,824 bytes, put `1Gi` in `/srv/my-brilliant-site.com/mailboxes/bob/quota`.

Quotas in a user's mailbox directory take precedence over the default quota.

Server-side filtering using Sieve

Sieve is a standard language that users can employ to filter their email on the server. Additionally using any one of a number of clients, users can edit their filtering rules without needing shell access to the server.

Each user can create a number of scripts in a directory called `sieve.d/`, with the current script being kept in a file called `sieve`.

Only one of these scripts can be active at a given time for each user; add to an existing file rather than creating a new one if you require extra filters.

Forward files

There are two methods of forwarding email. The first is a per-mailbox forwarding service, and the second is a per-domain service. For the per-user service, a file named `forward` should be put in a user's mailbox directory. The per-domain service uses the same file format as the per-user service, but the file should be uploaded to `config/default_forward` instead.

For example, **bob@my-brilliant-site.com** would set up a file called `/srv/my-brilliant-site.com/mailboxes/bob/forward`.

If all the mail for **my-brilliant-site.com** needed to be forwarded elsewhere, then the file would be called `/srv/my-brilliant-site.com/config/default_forward`.

Both of these files can be interpreted in two ways. Firstly they can be a comma separated list of email addresses. For example, if Bob wanted to forward his email onto Charlie and Dave, his forward file might read

```
charlie@example.com, dave@example.com
```

The second way these files are interpreted is as an Exim filter file. The full specification is documented at the Exim project site.

Here are some examples of what is possible.

To forward mail on, but keep a copy

```
# Exim filter
unseen deliver charlie@example.org
unseen deliver dave@example.com
```

To rewrite all mail for a domain to **example.com**. This is probably best used in `config/default_forward`.

```
# Exim filter
deliver $local_part@example.com
```

The Exim documentation has further examples of what is possible.

Vacation messages

It is possible to set a vacation message for a user by putting a message in file called `vacation` in the user's mailbox directory.

For example, for **bob@my-brilliant-site.com**, the message would go in `/srv/my-brilliant-site.com/mailboxes/bob/vacation`. On Bob's return, the people who received vacation messages are logged to `/srv/my-brilliant-site.com/mailboxes/bob/vacation.log`. Once he's read it, that file, along with `/srv/my-brilliant-site.com/mailboxes/bob/vacation` and `/srv/my-brilliant-site.com/mailboxe bob/vacation.db` should **all** be removed.

Important
Vacation messages can irritate other email users by replying to mailing lists, email bounces, and so on. Every effort is made to stop this from happening, but it is by no means fool-proof.

Email alias lists

Each domain can have a list of aliases. This is just a file that contains a list of local parts, and a list of places they should be sent on to. This file should be in the `config/` directory and is named `aliases`.

For example, **my-brilliant-site.com** has a list of dummy addresses that should be sent on to Bob. So the aliases file would be kept at `/srv/my-brilliant-site.com/config/aliases` and contains the following.

```
webmaster   bob@my-brilliant-site.com
chairman    charlie@example.com
staff       bob@my-brilliant-site.com, charlie@example.com, dave@example.com
```

This ensures that webmaster@my-brilliant-site.com is sent to bob@my-brilliant-site.com; chairman@my-brilliant-site.com is sent to charlie@example.com; staff@my-brilliant-site.com is sent to bob@my-brilliant-site.com, charlie@example.com, and dave@example.com.

Spam and virus scanning

Symbiosis comes with SpamAssassin and ClamAV installed to protect your email users against spam and virus in their inbox. To enable these features, simply create the files `config/antispam` or `config/antivirus` as appropriate. This will configure that domain to reject email if it is considered to be spam, or if it contains a virus.

If you'd rather accept all email and simply tag it as spam, put the word **tag** in `config/antispam`. This will also cause the email to be delivered into the `Spam/` folder for that user.

Customising SpamAssassin

The configuration for SpamAssassin for the **admin** user is kept in `/srv/.spamassassin/user_prefs`. Here you can adjust what score is needed to reject spam, and which tests are used during scanning. This file will only appear after a mail has been received with spam detection turned on, but one can be created and configured before this occurs.

The file contains comments and instructions, and further tips can be found on the SpamAssassin wiki.

In brief, to cause **more mail** to be rejected, you need to reduce the threshold score. Therefore change the line reading `#required_score 5` should be changed to `required_score 4`. Notice that the `#` has been removed at the start of the line to un-comment it.

Similarly if mail is being rejected, you can increase the score.

Further instructions can be found on the SpamAssassin wiki.

There is no facility to train the SpamAssassin Bayesian learner yet.

Filtering mail using headers

Headers are added to messages when spam or virus scanning is enabled. These can be used by email clients to filter email, for example in to spam or quarantine folders.

With spam scanning enabled, any email that **is** accepted has the following headers added

- `X-Spam-Score`
- `X-Spam-Bar`
- `X-Spam-Status`

The score is determined by SpamAssassin, and is the basis for acceptance or rejection. The higher the score, the more certain SpamAssassin is that the message is unwanted. The default threshold for rejection is 5.

The bar is a length of pluses or minuses that provide an easy-to-parse representation of the score. A positive score is given pluses, a negative score minuses. For example a score of 5.6 would be represented as `++++++`; a score of -2.2 would be represented as `--`.

The status is always either `innocent` or `spam`, depending on the score.

When virus scanning is enabled, the header `X-Anti-Virus` is added to messages that have been scanned. This is set to either `infected` or `clean`.

The content of an email can be changed if it's marked as spam by SpamAssassin. For example, you may wish to prepend an email's subject with **SPAM** to highlight it in your inbox. To do this, append the following code block to the end of the **/etc/exim4/system_filter** file:

```
if $h_X-Spam-Status: contains "spam"
then
    headers add "Old-Subject: $h_subject"
    headers remove "Subject"
    headers add "Subject: *** SPAM *** $h_old-subject"
    headers remove "Old-Subject"
endif
```

Using real-time blacklists from Spamhaus

There are three lists from Spamhaus that can be used to reject email based on the sender's IP address, namely

The Spamhaus Block List (SBL) a list of addresses from which Spamhaus does not recommend receiving email.

The Exploits Block List (XBL) a list of hijacked computers infected by third party exploits and viruses.

The Policy Block List (PBL) a list of addresses that should not be sending unauthenticated email at all.

These lists are combined to form the Zen list.

The following instructions will enable use of these lists on our example domain **my-brilliant-site.com**.

1. Connect to your machine using FileZilla
2. On the remote directory tree, navigate to `/srv/my-brilliant-site.com/config/`.
3. In this directory, create another directory called `blacklists/`. This is done by clicking the right mouse button on the `config/` directory, and selecting Create directory from the menu that pops up.
4. On your local machine create a file called `zen.spamhaus.org`. This is just an empty file.
5. Once this is done, navigate to the `blacklists` directory on the remote file system, and select `zen.spamhaus.org` from the local file system, and upload it. Make sure that the remote file has the correct name, i.e. no extra `.txt` extension.

That is all that is needed to start using the Spamhaus Zen blacklist. If you'd rather use a combination of lists create one or more of the following files:

- `sbl.spamhaus.org` to enable the SBL list
- `xbl.spamhaus.org` to enable the XBL list
- `pbl.spamhaus.org` to enable the PBL list
- `sbl-xbl.spamhaus.org` to enable the combined SBL and XBL list
- `zen.spamhaus.org` to enable the combined SBL, XBL, and PBL list

Manually blocking incoming mail from specific sources

While publicly maintained blacklists like spamhaus are much easier to rely on and lower maintenance, at some point you might find occasion to block specific email senders. Symbiosis allows blocking based on these criteria:

- Hostname of sender, which is matched against the reverse DNS of the sender's IP. Example entry: `*.bad-domain.com`
- IP of sender, which can be a single IP or a range specified in CIDR notation (be wary of blocking too much if you use this option). Example entry: `192.168.0.1`
- Address of sender. This option may specify wildcard records, eg "*@example.com" will catch all emails from that domain. Please note this works on the "envelope from" rather than the "from" address. Example entry: `bad_sender@example.com`.

To block with one of these criteria, you can use:

- `/etc/exim4/blacklist/by_hostname` for eg `.bad-domain.com`
- `/etc/exim4/blacklist/by_ip` for `192.168.0.1`
- `/etc/exim4/blacklist/by_sender` for `bad_sender@example.com`

Each entry to these files should be on a new line.

It is also possible to explicitly allow email from senders that would otherwise be blacklisted by adding entries in similarly named files under `/etc/exim4/whitelist`.

Rate limiting outbound email

Symbiois can now easily be configured to limit the number of outbound emails, either per-user, or per-domain. You might want to rate limit to prevent anyone taking advantage of your server, maybe using it to send out SPAM. The limit can be configured on a per-user basis with the following file :

`/srv/domain.com/mailboxes/bob/ratelimit`

and on a per-domain basis with the following file :

`/srv/domain.com/config/mailbox-ratelimit`

In both cases, the contents of the file should be a number, which represents the allowed limit per day. If the file is left blank, the default of 100 is applied. Senders who breach this limit will be sent an error email to explain why their message has not been sent, and discarded.

Setting an outbound IP for all email from a domain

If the domain has a `config/ip` file in place then the IP in that file will be used for outgoing email. This can be useful if your machine has a number of IP addresses and you're suffering from deliverability issues. config/

Blocking email

Symbiosis allows you to blacklist email senders by:

- email address (specific and wildcarded)
- by IP (specific and a range)
- by hostname (also specific or wildcarded).

Execute the following actions as **root** user rather than **admin**.

To block a specific email address, simply add the address to the file **/etc/exim4/blacklist/by_sender**.

The **by_sender** list accepts email addresses like **malefactor@example.com** or wildcarded ones like ***@example.com** to block a whole domain. Note that this blacklist is matched against the *Envelope Sender* address, rather than the *From* address.

To block by IP, add the IP address to **/etc/exim4/blacklist/by_ip**

The **by_ip** list accepts IP addresses like **192.168.66.6** or ranges like **10.66.6.0/24**. This is used to blacklist by the IP address of the connecting machine.

Finally the **by_hostname** list accepts hostnames like **bad.example.com**, or wildcarded like ***.example.com**. This is used to blacklist against the reverse DNS of the IP of the host connecting.

Enabling SNI for Exim and Dovecot

If you have multiple active domains with email hosted on your server, enabling SNI will allow you to use a unique SSL certificate for each. This will help avoid the "certificate mismatch" errors you may see when attempting to login to one of your email accounts.

To configure SNI support for Exim, please see this guide.

And to configure SNI support for Dovecot, please see this guide.

Configuration layout

Here is an example configuration layout for the domain `my-brilliant-site.com`. All the following files are kept in `/srv/my-brilliant-site.com/`.

`mailboxes/` This is where individual mailboxes are defined. If this directory does not exist, then mail will not be accepted for `my-brilliant-site.com`, unless a default forwarding address or filter has been set up.

`mailboxes/bob/` Mail will be accepted for the email address bob@my-brilliant-site.com.

`mailboxes/bob/Maildir/` This is where the email for bob@my-brilliant-site.com will be delivered. It will be **created automatically** upon receipt of the first message to that address.

`mailboxes/bob/password` File containing the password for bob@my-brilliant-site.com allowing him to collect his email over IMAP/POP3, and relay email using SMTP. His username is the same as his email address. See Section 5.4 for more information.

mailboxes/bob/quota File containing the quota for a user. The quota should a number of bytes. This can be followed by one of `k`, `M`, or `G` to specify kibibytes, mebibytes, or gibibytes respectively. For example `100M` would be 100 mebibytes, or 104857600 bytes. See Section 5.7 for more information.

mailboxes/bob/forward File containing either a comma-separated list of addresses, or an Exim filter. All mail addressed to bob@my-brilliant-site.com will be forwarded to the list of addresses, or processed by the filter. See Section 5.9 for more information.

mailboxes/bob/vacation File containing a vacation message for Bob. See Section 5.10 for more information.

mailboxes/bob/sieve File containing a Sieve filter. This can be edited by the user without shell access to the server. See Section 5.8 for more information.

config/aliases This file contains a list of aliases for a domain. The format is the local username followed by one or more spaces, and then comma separated list of email addresses which should receive the mail. See Section 5.11 for more information.

config/default_forward File containing either a comma-separated list of addresses, or an Exim filter. All mail addressed to the domain my-brilliant-site.com for local parts without directories under mailboxes will be forwarded to this address or processed by this filter. See Section 5.9 for more information.

config/antispam If this file is present, then all email for the domain my-brilliant-site.com will be scanned by SpamAssassin to determine whether it is spam. If it is spam, it will be rejected. If that file begins with the word tag, mail will never be rejected, just tagged as usual.

config/mailbox-quota If this file is present, then all mailboxes for this domain will have their quota determined by this file. The quota should a number of bytes. This can be followed by one of `k`, `M`, or `G` to specify kilobytes, megabytes, or gigabytes respectively. For example `100M` would be 100 megabytes, or 100,000,000 bytes. See Section 5.7 for more information.

config/antivirus If this file is present, then all email for the domain my-brilliant-site.com will be scanned for viruses by ClamAV. If a message is determined to contain a virus, it will be rejected. If that file begins with the word tag, mail will never be rejected, just tagged.

config/blacklists/sbl.spamhaus.org Reject mail for this domain if the sending machine's IP is listed in the Spamhaus Block List.

config/blacklists/xbl.spamhaus.org Reject mail for this domain if the sending machine's IP is listed in the Spamhaus Exploits Block List.

config/blacklists/pbl.spamhaus.org Reject mail for this domain if the sending machine's IP is listed in the Spamhaus Policy Block List.

config/blacklists/sbl-xbl.spamhaus.org Reject mail for this domain if the sending machine's IP is listed in either the Spamhaus or the Exploits block lists.

config/blacklists/zen.spamhaus.org Reject mail for this domain if the sending machine's IP is listed in the Spamhaus Zen Block List, which is a combination of the Spamhaus, Exploits, and Policy block lists.

Chapter 6

XMPP Reference

XMPP is a protocol that supports both private instant messages, and group instant messages. The server also supports features such as roster management, for keeping track of contacts and showing who is and is not online. Here is a broad overview of what the symbiosis-xmpp package supports:

- Federation - this is where users of your XMPP server may communicate with users of any other correctly configured XMPP server, with their own, locally hosted account. This will also allow a user to connect to multiple user chats (see below) on the local server or remote servers, should they wish to do so.
- Roster (contact list) management - before receiving messages from a new contact, each user must add the other to their contact list. The server will then remember contacts, such that they will be known on all a user's XMPP clients.
- Private messages - a user is able to speak to any of their online contacts
- Multiple user chat (MUC) - this feature enables users to communicate in groups, rather than one on one. These chats will often be named after the intended subject of discussion, eg "office" or "managers". If you wish, you can host a MUC that anyone else can connect to and use to chat. Think of this as being a cross between an instant message and a mailing list.

Symbiosis uses Prosody as its XMPP server.

To configure your domain to start using XMPP, create the file `config/xmpp`.

Adjusting the XMPP configuration

Domains' XMPP configuration is kept in `/etc/prosody/config.d` with each domain having its own snippet. Feel free to edit these snippets as you see fit, as once edited they will never get overwritten automatically.

If you do wish to restore the configuration to the default, you can run **symbiosis-xmpp-configure --force --verbose**.

Symbiosis uses a configuration template for the XMPP server. This is kept in `/etc/symbiosis/xmpp.d/prosody.template.erb`. This is the place to adjust things for all domains running on the server.

Once you've adjusted that to your liking, you can run **symbiosis-xmpp-configure --verbose** to apply your changes.

Chapter 7

FTP configuration

FTP users can be authenticated in two ways: on a per-domain basis, or on a per-user-per-domain basis. It is possible to enable other forms of authentication too.

Per-domain authentication

Basic per-domain authentication is controlled by the `config/ftp-password` file. This file contains the plain-text or hashed password for the FTP user whose username is the domain name. This user is limited to accessing the `public` directory for that domain.

For example, `/srv/my-brilliant-site.com/config/ftp-password` contains the password for the FTP user **my-brilliant-site.com**, and that user will be limited to accessing `/srv/my-brilliant-site.com/public`.

Multi-user authentication

This authentication method is controlled by the `config/ftp-users` file. This file contains more than just the password. Each line in the file represents a different user, and contains the username, password, base directory, and quota. Comments in the file start with **#**.

```
# username:password:directory:quota
bab:babs password:/path/to/base:10M
```

The directory and quota fields are optional. If the password field is empty, the user will not be able to log in.

In the above example, if that file was kept at `/srv/my-brilliant-site.com/config/ftp-users` then the user **babs@my-brilliant-site** would be able to log in with the password **babs password**. She'd be limited to the accessing files and directories below `/path/to/base`, and uploads to that that directory would be prohibited if it contains more than 10 Megabytes of data.

Other forms of authentication

It is possible to use the other forms of authentication provided by Pure-FTPd. The Pure-FTPd manual gives a good run down of all the various ways to do it. Here the two most common ways have been documented.

PureDB authentication

To enable authentication for virtual users, but would rather not use the Symbiosis method, you can create a Pure FTPd authentication DB, and use that. To tell the server to authenticate against it, you can run the following commands, as root.

```
echo /etc/pure-ftpd/pureftpd.pdb > /etc/pure-ftpd/conf/PureDB
touch /etc/pure-ftpd/pureftpd.pdb
ln -s /etc/pure-ftpd/conf/PureDB /etc/pure-ftpd/auth/50puredb
service pure-ftpd restart
```

Then you can use the pure-pw command to add new users. For example to add the user **foo**, you can run:

```
pure-pw useradd foo -u 1000 -g 1000 -d /path/to/home -m
```

It will prompt you for the password, and then rebuild the password file `/etc/pure-ftpd/pureftpd.pdb` automatically.

Pam authentication

If you would like to add normal PAM authentication, then you can run the following commands as root.

```
echo 1 > /etc/pure-ftpd/conf/PAMAuthentication
ln -s /etc/pure-ftpd/conf/PAMAuthentication /etc/pure-ftpd/auth/50pam
serivce pure-ftpd restart
```

Normal UNIX users should be able to log in now with their standard passwords.

Quotas

There are two ways of specifying a quota. The default quota for a domain goes in `config/ftp-quota`. This controls the quota for the per-domain user in `public`, as well as the default quota for users specified in `config/ftp-users`. Its format is the same as that for email quotas.

For the multi-user configuration file, a user's quota can be specified in the final field, again in the same format as that used for email quotas.

FTP configuration layout

`config/ftp-password` Domain-wide FTP user's password.

`config/ftp-quota` Default FTP quota for the domain.

`config/ftp-users` Per-user configuration for a domain.

Chapter 8

Firewall Reference

The firewall component of the Symbiosis system serves to protect the system by controlling its inbound and outbound connections. It comprises of a set of rules, and automatic whitelist and blacklist generation.

The firewall should be configured over SFTP as the **admin** user, and any changes made will take affect immediately.

Allowing and denying access to services

All usual firewall configuration can be carried out by creating and deleting files in `/etc/symbiosis/firewall/`. In this directory there are a number of subdirectories. Permissions for inbound connections are stored in `/etc/symbiosis/firewall/incoming.d/`, and outbound connections in `/etc/symbiosis/firewall/outgoing.d/`.

These files are all of the format `number-name`. The number determines the position of the rule in the firewall, the name is the name of the service that we wish to permit. These names are stored in `/etc/services`. There are also names that do not correspond to services, which are documented in the next section.

Additionally if the name is not known then the file format can be `number-number` where the first number specifies the position of the rule in the firewall, and the second number is the port that should be opened. For example, the files `10-http` and `10-80` achieve the same effect.

Finally, each file can contain a list of hostnames or IP addresses to which that rule will apply, one per line. For example, if addresses were added to an incoming rule, named `incoming.d/10-accept`, all connections **from** those addresses would be **accepted**. If a file were added named `outgoing.d/20-reject` and address added to that file, then outgoing connections **to** those addresses would be **rejected**.

For example, to allow an incoming connection to arrive at your machine, and be accepted, on port 22, you would create the file `/etc/symbiosis/firewall/incoming.d/10-ssh`. The firewall will update as soon as the file has been created, so no commands are needed to be run.

If you were wishing to ensure that your host would only accept incoming SSH requests from your office you might create the same file with the contents `office.my-brilliant-site.com`.

This would ensure that when the firewall was generated incoming connections on the SSH port would be accepted from the host `office.my-brilliant-site.com` but not from anywhere else.

Note
If hostnames, rather than IP addresses are used, then they are translated to IP addresses at the time the firewall is generated using DNS. If the IP address of a hostname changes, then the firewall may not function as intended until any cached DNS entries have expired, and the firewall has been regenerated.

Predefined special rules

There are a number of rules that don't naturally fit the convention described above. This list describes rules that have been written specially for Symbiosis to cope with these situations. Each rule described below can be used in both `incoming.d/` and `outgoing.d/`, and for both IPv4 and IPv6 addresses, unless otherwise specified.

These rules are used in the same way as those described in the previous chapter. Files are added in the `incoming.d/` or `outgoing.d/` directory with the name prefixed by a number giving the position of the rule. The files can contain addresses or hostnames, one per line, against which the rule should be applied.

accept Accept all connections. Uses the iptables `ACCEPT` target.

allow Alias of **accept**.

blacklist Alias of **reject**.

collector Permit TCP connections on port 1919.

dns Accept incoming TCP and UDP connections from port 53 to high-numbered, unprivileged ports. Designed to allow replies to DNS queries. This rule can be removed in favour of **related**. This is for **incoming** connections only.

drop Drop all connections. Uses the iptables `DROP` target.

essential-icmpv6 Accept ICMPv6 packets that are essential for IPv6 networking to operate. Without this rule the machine IPv6 networking **will not work**. It permits ICMPv6 types destination-unreachable, packet-too-big, parameter-problem, router-solicitation, router-advertisement, neighbor-solicitation, and neighbor-advertisement. This is **IPv6** only.

established Permit connections that are already established. Uses the iptables `ESTABLISHED` target.

ftp Permit TCP connections on both ports 20 and 21, i.e. ftp and ftp-data.

icmp Permit all ICMP connections. This **IPv4** only.

icmpv6 Permit all ICMP6 connections. This is **IPv6** only.

imager Permit TCP connections on port 5000.

new Permit new connections.

ping Permit ICMP types echo-request, echo-reply, and ttl-exceeded, for allowing the machine to be pinged, and show up on traceroutes.

reject Reject all connections. Uses the iptables `REJECT` target. For TCP connections a TCP reset is sent. Otherwise it returns port unreachable.

related Accept new connections, but only if they are associated with an existing one, for example DNS queries, or FTP data transfer.

whitelist Alias of **accept**.

These rules are all contained in `/usr/share/symbiosis/firewall/rule.d/`. It is perfectly possible to write your own rules based on those in this directory, but they should be kept in `/usr/local/share/symbiosis/firewall/rule.d/`.

An example firewall

This example should be read in conjunction with the previous sections. A machine has the following firewall rules defined for its incoming connections.

- `incoming.d/00-related`
- `incoming.d/00-established`
- `incoming.d/05-essential-icmpv6`
- `incoming.d/05-ping`
- `incoming.d/07-ssh` which contains `1.2.3.4`, and `2001:41c8:1:dead:beef::/64` on separate lines.
- `incoming.d/10-http`
- `incoming.d/20-25`
- `incoming.d/99-reject`
- `incoming.d/100-666`

This would set up a firewall that would do the following tests, in order:

1. Accepted all packets from **established** connections.
2. Accepted all packets from **related** connections
3. Accepted all ICMPv6 packets required for IPv6 connectivity.
4. Accepted ICMP/ICMPv6 packets required for pings and traceroutes.
5. Accepted new TCP/UDP connections to port 22 (SSH), but only from 1.2.3.4 or addresses in the 2001:41c8:1:dead:beef netblock.
6. Accepted new TCP/UDP connections to port 666. Note that this rule comes before `10-http`, even though it is called `100-666`. This is because the order is given by the ASCII rather than numerical value of the filename.
7. Accepted new TCP/UDP connections to port 80 (HTTP).
8. Accepted new TCP/UDP connections to port 25 (SMTP).
9. Rejected anything that had not been accepted yet.

These rules would be installed for IPv4 and IPv6 connections using iptables and ip6tables respectively. To inspect the firewall rules at any given time, you can run `sudo iptables -L -v -n` which will return the current firewall status. In this example, the rules would look like this.

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in out source          destination
    0     0 ACCEPT    all  --  lo *   0.0.0.0/0       0.0.0.0/0
   13  1012 whitelist all  --  *  *   0.0.0.0/0       0.0.0.0/0
    0     0 blacklist all  --  *  *   0.0.0.0/0       0.0.0.0/0
    0     0 ACCEPT    all  --  *  *   0.0.0.0/0       0.0.0.0/0 state ESTABLISHED
    0     0 ACCEPT    all  --  *  *   0.0.0.0/0       0.0.0.0/0 state RELATED
    0     0 ACCEPT    icmp --  *  *   0.0.0.0/0       0.0.0.0/0 icmp type 8
```

```
    0     0 ACCEPT    icmp --  *  *   0.0.0.0/0       0.0.0.0/0 icmp type 0
    0     0 ACCEPT    icmp --  *  *   0.0.0.0/0       0.0.0.0/0 icmp type 11
    0     0 ACCEPT    tcp  --  *  *   1.2.3.4         0.0.0.0/0 tcp dpt:22
    0     0 ACCEPT    udp  --  *  *   1.2.3.4         0.0.0.0/0 udp dpt:22
    0     0 ACCEPT    tcp  --  *  *   0.0.0.0/0       0.0.0.0/0 tcp dpt:80
    0     0 ACCEPT    udp  --  *  *   0.0.0.0/0       0.0.0.0/0 udp dpt:80
    0     0 ACCEPT    tcp  --  *  *   0.0.0.0/0       0.0.0.0/0 tcp dpt:666
    0     0 ACCEPT    udp  --  *  *   0.0.0.0/0       0.0.0.0/0 udp dpt:666
    0     0 ACCEPT    tcp  --  *  *   0.0.0.0/0       0.0.0.0/0 tcp dpt:25
    0     0 ACCEPT    udp  --  *  *   0.0.0.0/0       0.0.0.0/0 udp dpt:25
    0     0 REJECT    all  --  *  *   0.0.0.0/0       0.0.0.0/0 reject-with icmp- ↩
       port-unreachable

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in out source          destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in out source          destination
    0     0 ACCEPT    all  --  *  lo  0.0.0.0/0       0.0.0.0/0
    7  1388 ACCEPT    all  --  *  *   0.0.0.0/0       0.0.0.0/0 state ESTABLISHED
    0     0 ACCEPT    all  --  *  *   0.0.0.0/0       0.0.0.0/0 state RELATED
    0     0 REJECT    all  --  *  *   0.0.0.0/0       0.0.0.0/0 owner UID match 33 ↩
        reject-with icmp-port-unreachable

Chain blacklist (1 references)
 pkts bytes target    prot opt in out source          destination
    0     0 REJECT    all  --  *  *   71.63.72.4      0.0.0.0/0 reject-with icmp- ↩
       port-unreachable
    0     0 REJECT    all  --  *  *   61.145.118.190  0.0.0.0/0 reject-with icmp- ↩
       port-unreachable

Chain whitelist (1 references)
 pkts bytes target    prot opt in out source           destination
   13  1012 ACCEPT    all  --  *  *   212.110.163.132 0.0.0.0/0
```

This listing shows how the rules in the files under `/etc/symbiosis/firewall/` are translated into iptables rules. It also shows that by default all connections on the loopback interface **lo** are permitted, and that the whitelist and blacklist tables have references in the `INPUT`, i.e. incoming, table before the rules defined in `/etc/symbiosis/firewall/incoming.d/` are applied.

IPv6 rules follow the same format, and can be checked by running `sudo ip6tables -L -v -n`.

Making custom additions to your firewall

The Symbiosis firewall package should allow you to carry out the most common tasks, simply by creating files named after the services you wish to permit or deny.

However there are times when you might wish to make your own custom additions, and for this purpose the firewall package allows you to run an unlimited number of custom scripts/programs once it has loaded the rules - these scripts may perform arbitrary actions, but will be most typically used to update the firewall rules, via the iptables or ip6tables commands.

The program run-parts is used to execute scripts in `/etc/symbiosis/firewall/local.d/`, after the firewall has finished loading. This means that the scripts have to have to fulfil the naming conditions described in the run-

parts(8) manual page. Essentially the script should be marked executable, and only contain alphanumeric characters in its name.

Warning
If any scripts in `local.d/` exit with a non-zero status the firewall will be deemed to have failed in some way, and the firewall will be restored to its prior state.

Blocking abusive remote hosts

The `symbiosis-firewall-blacklist` tool runs four times an hour, and is designed to scan your server's logfiles for abusive behaviour from malicious remote hosts. Malicious activity which is detected will result in the remote host being denied further access to your server.

Currently we regard malicious activity as:

- Invalid SSH logins.
- Invalid FTP logins.
- Invalid SMTP/POP3/IMAP/ManageSieve logins.

Every 15 minutes various logfiles are scanned for certain patterns to search for new malicious IPs, and the firewall is updated.

These patterns are defined in `/etc/symbiosis/firewall/patterns.d/`. For example, for SSH the following pattern definition is used:

```
#
#  The logfile we look for matches within.
#
file = /var/log/auth.log ❶

#
#  Any matches will be denied access to these ports.
#
#  Comma-separated values are expected.
#
ports = 22 ❷


#
#  Patterns we'll match upon.
#
Failed password for invalid user [^ ]+ from __IP__ port [^ ]+ ssh2 ❸
Failed password for [^ ]+ from __IP__ port [^ ]+ ssh2
```

❶ Is the file to search

❷ Are the ports to block

❸ Are the regular expressions to look for, where __IP__ is a pre-defined regular expression that matches both IPv4 and IPv6 addresses.

If an IP matches one of those patterns in the period since the last check was made, it is added to the blacklist.

Disabling the firewall completely will disable the blacklisting behaviour, but you might also wish to disable that separately.

To do this, login over SFTP as **admin** and create the file `/etc/symbiosis/firewall/blacklist/disabled`. This will immediately disable and clear the blacklist.

Note
IPv6 addresses are masked to a /64, which is the smallest assignment of addresses recommended for an end site.

Whitelisting "known-good" IP addresses

The `symbiosis-firewall-whitelist` tool runs once per hour, and is designed to perform the opposite task to the `symbiosis-firewall-blacklist` script - in short it is designed to ensure that any remote host which has successfully connected to your server in the past isn't (accidentally) blacklisted in the future.

Every hour the script will examine the successful logins which have been observed recently. Each IP address which has successfully been the source of a login attempt will be permitted access to the system on a global basis, and will thus not be locked out.

As with the automatic blacklist, IPv6 addresses are masked to a /64, which is the smallest recommended assignment for an end site.

To disable the automatic whitelist, login over SFTP as **admin** and create the file `/etc/symbiosis/firewall/whitelist.d/disabled`. This will immediately clear the whitelist, and prevent further updates.

You can add your own entries to the whitelist, which never expire, by creating entries in the directory `/etc/symbiosis/firewall/whitelist.d/`. Create the file `/etc/symbiosis/firewall/whitelist.d/<ip address>` and the specified IP address will not be blacklisted, or refused access to your server.

SYN-ACK/ACK flood protection

Symbiosis now comes with basic SYN-ACK/ACK flood protection. These are simple but effective denial of service attacks, which can leave the network stack inundated. Wikipedia has an article on the matter for the curious

To enable the protection, create the following file :

`/etc/symbiosis/firewall/incoming.d/00-syn-ack-flood-protection`

Disabling the firewall

If you wish you may disable the firewall completely, allowing remote users to connect to any service you have running upon your machine.

We'd not recommend that you disable the firewall, because it does provide a increase in system security, but if you wish it is possible by executing the following two commands:

```
touch /etc/symbiosis/firewall/disabled
sudo symbiosis-firewall flush
```

The presence of the disabled rule will not itself clear the firewall, merely prevent further updates to it, which is why the `flush` command is needed.

Configuration layout

All configuration of the firewall is conducted via the presence or absence of files in a number of directories beneath `/etc/symbiosis/firewall/`. Actions and rules are all kept under `/usr/share/symbiosis/firewall/`.

`/etc/symbiosis/firewall/blacklist.d/` A persistent record of IP addresses which are blacklisted, such that no connections will be permitted from them.

`/etc/symbiosis/firewall/blacklist.d/disabled` If this file is present, then the automatic blacklisting is disabled.

`/etc/symbiosis/firewall/disabled` If this file is present then the firewall will be disabled. However this will not clear the firewall rules. See Section 8.8.

`/etc/symbiosis/firewall/incoming.d/` Settings related to the incoming connections your machine will receive.

`/etc/symbiosis/firewall/local.d/` The place to add local customisations.

`/etc/symbiosis/firewall/outgoing.d/` Settings related to the outgoing connections your machine is permitted to initiate.

`/etc/symbiosis/firewall/patterns.d/` A collection of pattern files use by symbiosis-firewall-blacklist to automatically determine addresses to blacklist

`/etc/symbiosis/firewall/whitelist.d/` A persistent record of IP addresses which are always allowed to connect to your server.

`/etc/symbiosis/firewall/whitelist.d/disabled` If this file is present, then the automatic whitelisting is disabled.

`/usr/share/symbiosis/firewall/action.d/` This directory contains the various actions that the symbiosis-firewall uses to maintain the firewall. If you wish to write your own actions, or change the ones that come with symbiosis, they should go in `/usr/local/share/symbiosis/firewall/action.d/`.

`/usr/share/symbiosis/firewall/rule.d/` This directory contains the various pre-defined rules described in Section 8.2. If you wish to add your own rules, or change the ones provided, they should go in `/usr/local/share/symbiosis/firewall/rule.d/`.

Chapter 9

DNS Hosting

To take full advantage of the Symbiosis system, your domain needs to be configured to have Bytemark's name servers as authority for it.

What follows only applies if our name servers are used; if that is not the case you will need to manage your DNS data outside of the Symbiosis system. Section 9.1 gives a listing of the records needed for the correct functioning of the system.

All domains which are hosted upon a Symbiosis system will have their DNS records automatically uploaded to the Bytemark Content DNS servers.

By default a set of typical records is created for each hosted domain with MX records pointing to the local system, and aliases such as *www.* and *ftp.* for convenience. If you wish you may edit the records to make custom additions or otherwise make changes to those defaults.

For the domain "my-brilliant-site.com" you will find the auto-generated DNS records in `/srv/my-brilliant-site.com/config/dns/my-brilliant-site.com.txt`

The DNS files are uploaded to the Bytemark content DNS service every hour, and allow you to use the full range of available TinyDNS options. These options are documented upon the Bytemark Website and in the TinyDNS documentation.

Example DNS records

This is an example of the records Symbiosis generates for `my-brilliant-site.com`. They are created automatically and stored in `config/dns/my-brilliant-site.com.txt`.

DNS records example

```
#
#  Nameserver records. ❶
#
.my-brilliant-site.com::a.ns.bytemark.co.uk:300
.my-brilliant-site.com::b.ns.bytemark.co.uk:300
.my-brilliant-site.com::c.ns.bytemark.co.uk:300

#
#  The domain name itself ❷
#
=my-brilliant-site.com:89.16.174.65:300
```

```
#
#  Useful aliases. ❸
#
+ftp.my-brilliant-site.com:89.16.174.65:300
+www.my-brilliant-site.com:89.16.174.65:300
+mail.my-brilliant-site.com:89.16.174.65:300

#
# A record for MX ❹
#
+mx.my-brilliant-site.com:89.16.174.65:300

#
# The domain name itself -- AAAA record and reverse. ❺
#
6my-brilliant-site.com:200141c80001596d0000000000000065:300

#
# Useful aliases -- AAAA records only
#
3ftp.my-brilliant-site.com:200141c80001596d0000000000000065:300
3www.my-brilliant-site.com:200141c80001596d0000000000000065:300
3mail.my-brilliant-site.com:200141c80001596d0000000000000065:300

#
# AAAA record for MX
#
3mx.my-brilliant-site.com:200141c80001596d0000000000000065:300

#
#  MX record -- no IP defined, as this is done separately above. ❻
#
@my-brilliant-site.com::mx.my-brilliant-site.com:15:300
```

❶ These lines create *NS* and *SOA* records for `my-brilliant-site.com` pointing at a.ns.bytemark.co.uk, b.ns.bytemark.co.uk, and c.ns.bytemark.co.uk. The time-to-live for these records is 300 seconds. Note that the double colons in these records are deliberate as the IP addresses are defined elsewhere by Bytemark.

❷ This creates an *A* record pointing `my-brilliant-site.com` to the IP address `89.16.174.65`, and a *PTR* record for the reverse. Again, the TTL is 300 seconds.

❸ These three lines add *A* records for expected aliases. Once again, the TTL for these records is 300 seconds.

❹ This line adds in an *A* record for the *MX* record defined below.

❺ From here the IPv6 equivalents of **2**, **3**, and **4** are specified, using *AAAA* records is used instead of an *A* record. Note that IPv6 addresses are specified in full, without any colons.

❻ This last record creates an *MX* record directing mail for `my-brilliant-site.com` to `mx.my-brilliant-site.com`, with a distance of 15. The double colon is deliberate since we defined the *A* record for `+mx.my-brilliant-site.com` in <4>, and an *AAAA* record for the same name in <5>.

Adding a wild-card hostname record

In addition to these records for each domain, a wild-card *A* record is needed for the hostname such that the `.testing.` prefix works. If your machine is at Bytemark, this has already been setup for your machine's Bytemark alias, for example *example.default.bytemark.uk0.bigv.io*.

If your machine is not hosted at Bytemark, or your hostname does not end in `bytemark.co.uk` then you will need to set this alias up. Adding the following line to your DNS file will work, assuming the domain is hosted at Bytemark. This assumes that your machine is called `host.example.com` and that your machine's IP address is 1.2.3.4.

```
+*.host.example.com:1.2.3.4
```

Adding a custom TTL per domain

Symbiosis allows adding a custom TTL to a domain. If you're unfamiliar, you can read more about time to live (TTL) here. You can configure this by creating the file :

`/srv/my-brilliant-site.com/config/ttl`

The contents of the file should be a number, and it represents the time a name server can cache the record in seconds. A lower TTL is good for making frequent changes, as clients won't cache for too long. A longer TTL is good for times when DNS is unavailable for some reason.

Adding a DMARC policy per domain

There's also an easy way to add a DMARC policy on a per-domain basis. If you're unfamiliar with DMARC, Wikipedia has an article . It provides indication that emails are protected by SPF and/or DKIM. It can be configured by creating a file in the format :

`/srv/my-brilliant-site.com/config/dmarc`

You may leave this as an empty file, and Symbiosis will use its defaults. If you prefer, the file can contain your own DMARC string.

Moving domains between machines using the Bytemark content DNS service

If you wish to move your domains between two machines running Symbiosis and using the Bytemark content DNS service, you must contact Bytemark Support to arrange the domain to be moved between content DNS accounts.

This results from the necessity for ensuring that only people with the proper authorisation can change live DNS data. Once a domain has been hosted on our network, a content DNS account will have sole authority for it.

If you purchase a second server and move some of your domains onto it, or purchase a domain from another Bytemark customer you must contact us to move authority for the domain into the correct account.

Until this is done, although the Symbiosis system will be creating and uploading data it will not be to the account with the authority to make the data live.

Configuring SPF and DKIM records

SPF and *DKIM* are standards that help mail servers decide if email is legitimate, ensuring it is more likely to reach the intended recipient's inbox instead of being rejected or marked as spam. Both these technologies require creation of one or more DNS records.

Adding SPF records

Before adding any records, a policy needs to be decided. The guide at OpenSPF can help determine what the record should look like. The default policy Symbiosis uses is **v=spf1 +a +mx ?all**.

To create SPF records simply create the file `/srv/my-brilliant-site.com/config/spf`. Nothing more is required if the default policy is adequate. If you have decided on a different policy, then you can just write it to this file.

A task is run hourly to generate the DNS data and upload it to the Bytemark DNS servers, at which point the domain will start benefiting from it. If you wish to speed up this process, run **sudo symbiosis-dns-generate --verbose**.

Adding DKIM records

DKIM is a way of cryptographically signing email headers to provide a level of confidence surrounding the origin of said email. Configuring DKIM requires a private RSA key, and a DNS record specifying the public part of the key, along with a policy dictating how the key should be used. For DKIM to work in Symbiosis two files are required, one contains the private key, and the second contains the selector (or nothing).

1. To generate the private key, run **openssl genrsa -out /srv/my-brilliant-site.com/config/dkim.key 2048** on your server. This will generate a key that is 2048 bits long. Set the permissions of this key appropriately with **chmod 640 /srv/my-brilliant-site.com/config/dkim.key** and **chown admin:Debian-exim /srv/my-brilliant-site.com/config/dkim.key**.
2. Next, create the file `/srv/my-brilliant-site.com/config/dkim`, either as an empty file or with the selector in it. If the file is empty, the selector is left as the first component of the machines hostnome, or "default" if this cannot be determined.

Once both files are in place the hourly DNS task will update the DNS records for your domain and upload them as usual. If you wish to speed up this process, run **sudo symbiosis-dns-generate --verbose**.

Chapter 10

Scheduled tasks

Jobs can be scheduled to run on a per-domain basis. This is configured in the same style as the traditional crontab, and is kept in the `config/` directory of a domain.

Testing the crontab

The crontab can also be tested. To do this you have to SSH to the machine, usually as **admin** to run the command.

For example, to test the **my-brilliant-site.com** crontab navigate to `/srv/my-brilliant-site.com/config/` and run **symbiosis-crontab --test crontab**.

The **my-brilliant-site.com** crontab reads

```
# Send any output to Bob
#
MAILTO=bob@my-brilliant-site.com

#
# run at 18:40 every day
#
40 18 * * *       echo Hello Dave.

#
# run at 9am every Monday - Friday
#
0 9 * * mon-fri wget http://www.my-brilliant-site.com/cron.php

#
# Run once a month
#
@monthly          /usr/local/bin/monthly-job.sh
```

Therefore the output generated is

```
 Environment
 ---------------------------------------------------------------------------
 HOME = /srv
 LOGNAME = admin
 PATH = /usr/local/bin:/usr/bin:/bin
 MAILTO = bob@my-brilliant-site.com
```

```
==========================================================================

Jobs next due -- Local time 2010-06-17T17:57:37+01:00
--------------------------------------------------------------------------
Date                        Command
--------------------------------------------------------------------------
2010-06-17T18:40:00+01:00  echo Hello Dave.
2010-06-18T09:00:00+01:00  wget http://www.my-brilliant-site.com/cron.php
2010-07-01T00:00:00+01:00  /usr/local/bin/monthly-job.sh
==========================================================================
```

Note
The only environment variables that can be set within your crontab are PATH and MAILTO. All the rest are set automatically, and cannot be altered.

System scheduled tasks

There are various automated tasks which are executed upon a Symbiosis system. These scheduled tasks are responsible for automating things such as:

- The addition of new IP addresses to your system.
- The generation and upload of DNS data.

The following section document precisely which jobs are installed by default, along with their purpose.

These are the system tasks which are installed by default:

/etc/cron.d/symbiosis-common This carries the rudimentary password checks on mail and FTP passwords on an hourly and weekly basis.

/etc/cron.d/symbiosis-cron This is responsible for launching any user-scheduled jobs, as described in Chapter 10, and is run every minute.

/etc/cron.d/symbiosis-firewall The jobs here are responsible for checking for new blacklist and whitelist entries, as discussed in Section 8.5. The whitelist and blacklists are regenerated every 15 minutes. The whole firewall is reloaded hourly.

/etc/cron.d/symbiosis-monit This schedules the Symbiosis service monitor, which is described in Chapter 13. This is run every two minutes.

/etc/cron.d/symbiosis-dns This regenerates DNS data for all the domains in `/srv/`, and triggers an upload to the Bytemark DNS server.

/etc/cron.daily/symbiosis-httpd-rotate-logs This manages rotation of the webserver log files for each domain.

/etc/cron.hourly/symbiosis-configure-ips This adds IP addresses configured by domains in `config/ip` to the host.

/etc/cron.hourly/symbiosis-httpd-configure This task creates a per-IP Apache configuration file for new IP addresses, and is closely related to the previous task.

Chapter 11

Database configuration

Initially the **root** password for the database is the same as that of the **admin** user used to to connect to your machine via SSH or SFTP. To change this you can use the phpMyAdmin interface.

As a general rule, each application should have its own username and access rights, to make sure that there is a degree of separation between all the applications on a server. This can all be done through the phpMyAdmin interface.

In Symbiosis Stretch, MySQL uses unix socket auth for the **root@localhost** user by default, which is incompatible with phpMyAdmin. As such, a new **admin@localhost** user has been created to be used with phpMyAdmin. This user has privileges equivalent to root, but uses traditional username/password authentication instead.

Adding a user with remote privileges

There are two ways to do this, either using the MySQL command line tool, or via phpMyAdmin. This section will cover doing it with the latter.

1. In phpMyAdmin, select the Privileges link from the front page, once you've logged in to it as **root** (or **admin** on Symbiosis Stretch, as mentioned above).
2. The privileges section will present a **User Overview**, at the bottom of which there is a link to Add a new user.
3. In the **Add a new user** screen, fill out the details in the form as needed, making sure that the Host field is set to **Any host**.

   The privileges tick boxes lower down should be selected carefully. Most applications will need at least those in the Data section, and some of those in the Structure section. Check the documentation of the software you're using to see what it requires.

   If you want an account with all privileges, select check all.
4. Once you're satisfied with everything, click Go. This will confirm that a user has been created.
5. Now return to the home screen by clicking the phpMyAdmin logo at the top left of the screen.
6. Finally, on the front page click the Reload privileges link to make sure MySQL knows about this new user.

You should now be able to access the MySQL database remotely, using this new username and password.

Chapter 12

Backup Reference

The Symbiosis system includes a component designed to handle backups, using the flexible backup2l software.

backup2l was selected due to its simplicity and flexibility, which allows it to be used easily. By default the backup software executes once per day and archives the contents of significant directories to a local directory. Before the actual backup takes place, the total space needed is calculated. If there is not sufficient storage space to accomodate the new backup, the backup operation will not proceed and no backups will be made. An error is generated in this case.

Configuration

In Symbiosis the Backup2l configuration is generated from the snippets in `/etc/symbiosis/backup.d/conf.d/`.

- The local directories to backup (`/etc/`, `/srv/`, etc).
- The destination to which the backups should be stored (`/var/backups/localhost/`)
- The number of backups to keep.

Advanced Configuration

Additionally we've configured the backup software to easily execute a number of scripts before and after the backup is performed:

`/etc/symbiosis/backup.d/pre-backup.d/` Any executable script located in this directory is executed, prior to a backup execution.

`/etc/symbiosis/backup.d/post-backup.d/` Any executable script located in this directory is executed after a backup has completed.

Listing Backup Contents

To list the contents of your backup area you need to run backup2l with the "-l" flag:

```
all.1: /etc/.pwd.lock
all.1: /etc/GeoIP.conf.default
all.1: /etc/X11/Xresources/x11-common
all.1: /etc/X11/Xsession
all.1: /etc/X11/Xsession.d/20x11-common_process-args
all.1: /etc/X11/Xsession.d/30x11-common_xresources
all.1: /etc/X11/Xsession.d/40x11-common_xsessionrc
all.1: /etc/X11/Xsession.d/50x11-common_determine-startup
...
```

Here you will see the contents of the `/etc/` directory which have been archived.

If you'd like to restrict this view you can apply a regular expression to filter the results. For example we can list the files which match the pattern *passwd* with this command:

```
~$ sudo backup2l -l passwd
Listing locations...
all.1: /etc/exim4/passwd.client
all.1: /etc/passwd
all.1: /etc/passwd-
all.1: /etc/phpmyadmin/htpasswd.setup
all.1: /etc/pure-ftpd/pureftpd.passwd
...
```

Restoring From Backup

To illustrate how this works, an example is used. We're looking for a backup of the file `/etc/passwd`.

1. First log in to your machine over SSH as **admin**.
2. To find the available versions of the file, run **sudo backup2l -l '/etc/passwd$'**. The dollar sign is there to show that you want an exact match of `/etc/passwd`. The first time you run **sudo** you will be prompted for the **admin** password. The following output will be generated by backup2l.

   ```
   backup2l v1.5 by Gundolf Kiefer

   Active files in <all.1101>: 1
     found in all.1101:       0   (    1 left)
     found in all.11:         1   (    0 left)

   Listing locations...
   all.11: /etc/passwd
   ```

   This shows the latest available version of the file
3. To recover it you should run **sudo backup2l -r '/etc/passwd$'**. The following output will be generated

   ```
   backup2l v1.5 by Gundolf Kiefer

   Active files in <all.1101>: 1
     found in all.1101:       0   (    1 left)
     found in all.11:         1   (    0 left)

   Restoring files...
     all.11.tar.gz: 1 file(s) using 'DRIVER_TAR_GZ'
   ```

That has restored the file to `etc/passwd` in the current directory. It is **not recommended** to run this program in the `/` directory, as any existing files will get overwritten.

Recovery From Earlier Backups

It is also possible to pick which version of a file you wish to restore.

1. First login to your machine over SSH as **admin**
2. Then, to show all available versions of a file, run **sudo backup2l -a '/etc/passwd$'**. Again, the first time you run **sudo** you will be prompted for a password. The following output is generated.

```
backup2l v1.5 by Gundolf Kiefer

Listing available files...
all.101   -     1067 06/18/08 13:59:47 0000.0000 0644 /etc/passwd
all.101   +     1118 06/19/08 11:29:10 0000.0000 0644 /etc/passwd
all.108   -     1118 06/19/08 11:29:10 0000.0000 0644 /etc/passwd
all.108   +     1153 08/27/08 10:25:45 0000.0000 0644 /etc/passwd
all.11    -     1067 06/18/08 13:59:47 0000.0000 0644 /etc/passwd
all.11    +     1153 08/27/08 10:25:45 0000.0000 0644 /etc/passwd
all.1     +     1067 06/18/08 13:59:47 0000.0000 0644 /etc/passwd
```

   Note that the versions are not shown in date order, and that the dates are in the US `mm/dd/yy` format. In that list the + indicates that the file is new and thus contained in the archive file. A − indicates that the file has been removed (or replaced). Choose which backup you wish to recover from.

3. To recover the file dated 19th June 2008, you need backup number 101 — remember the + indicates that it is present in that archive. To recover that file, run **sudo backup2l -t 101 -r '/etc/passwd$'**

```
backup2l v1.5 by Gundolf Kiefer

Active files in <all.101>: 1
  found in all.101:        1   (    0 left)

Restoring files...
  all.101.tar.gz: 1 file(s) using 'DRIVER_TAR_GZ'
```

   Notice the -t 101 argument which specifies which backup we want to restore from.

We have now successfully restored the file to `etc/passwd` in the current directory. We can check by running **ls -la etc/**

```
total 16
drwxr-xr-x  2 root  root  4096 2008-09-09 09:56 .
drwxr-xr-x 14 root  root  4096 2008-09-09 09:51 ..
-rw-r--r--  1 root  root  1118 2008-06-19 11:29 passwd
```

Offsite backup storage

The Symbiosis system assumes that it has access to an associated external storage area. It will try and use rsync to upload the backups to this area, via a script named `/etc/symbiosis/backup.d/post-backup.d/99-upload-bac`

If the host is on Bytemark's network, this script can establish the backup space name automatically. Otherwise you can specify it manually by setting the full rsync path in `/etc/symbiosis/dns.d/backup.name`.

Recovering from the offsite backup storage

Before each backup a second script will synchronise the remote backup space locally, ensuring that a complete set of backups are held in both places. This means that if disaster strikes your machine, it is straightforward to recover your backups. This is done by running `/etc/symbiosis/backup.d/pre-backup.d/00-download-backup`.

This also helps to maintain the integrity of the differential backups provided by backup2l by replacing any files accidentally removed from the local backup directory before the backup starts.

Trimming the size of the local backups.

It is possible to reduce the size of the backups stored locally. The first thing to do is check the current status of the backups by running **sudo backup2l -s**. This will present a summary of the current backups. For example:

```
backup2l v1.5 by Gundolf Kiefer

Summary
=======

Backup       Date       Time  |  Size   | Skipped  Files+D |  New  Obs. | Err.
- ------------------------------------------------------------------------
all.1        2010-08-10 02:52 |   41.7M |       0     3836 | 3836     0 |    0
all.11       2010-11-01 04:45 |   38.1M |       0     3935 | 1517  1418 |    0
all.12       2011-01-21 04:27 |   39.7M |       0     3985 |  561   511 |    0
all.121      2011-01-30 04:38 |   10.5M |       0     4001 |  137   121 |    0
all.122      2011-02-08 03:54 |    1.5M |       0     4029 |  129   101 |    0
all.123      2011-09-07 05:08 |   33.8M |       0     3892 | 1437  1574 |    0
all.124      2011-09-16 05:07 |    1.3M |       0     4791 |  956    57 |    0
all.125      2011-09-25 04:45 |    868K |       0     5676 |  928    43 |    0
all.126      2011-10-04 05:15 |   11.3M |       0     6559 |  990   107 |    0
all.127      2011-10-13 04:29 |    894K |       0     7444 |  928    43 |    0
all.128      2011-10-22 04:59 |    345K |       0     8329 |  935    50 |    0
all.13       2011-10-31 05:03 |   45.7M |       0     9218 | 6833  1600 |    0

Filesystem            Size  Used Avail Use% Mounted on
/dev/vda               10G  1.9G  7.6G  20% /
```

From here it is possible to see which levels of backups that can be pruned. In the above example the third-level backups `all.121` to `all.128` can be pruned, as there has been a subsequent second level backup, `all.13`. The downside of this is that any changes contained in those backups will be lost, and only changes from the `all.12` will be available.

To prune these backups run **sudo backup2l -p 121**. This will then show Backup2l removing `all.121` and all its dependent backups.

```
backup2l v1.5 by Gundolf Kiefer

Purging <121>...
  removing <all.121>
  removing <all.122>
  removing <all.123>
  removing <all.124>
  removing <all.125>
  removing <all.126>
  removing <all.127>
```

```
  removing <all.128>
```

Finally we need to make sure these deletions are synchronised to the remote backup space, to ensure that our deleted files do not mysteriously return again prior to the next backup run.

```
sudo /etc/symbiosis/backup.d/post-backup.d/99-upload-backup
```

Which will provide output similar to that shown below.

```
Sending backups to example.backup.bytemark.co.uk::example/example.default.bytemark ↩
    .uk0.bigv.io...
building file list ... done
deleting localhost/all.lock
deleting localhost/all.128.tar.gz
....
deleting localhost/all.121.error.gz
deleting localhost/all.121.check
localhost/

sent 2.95K bytes  received 22 bytes  1.98K bytes/sec
total size is 400.59M  speedup is 134742.36
```

Those level three backups will no longer exist.

Making changes to the backup2l configuration

The configuration is kept in `/etc/symbiosis/backup.d/conf.d`. If you need to make changes, you should make them to the files in that directory and then run **make** to generate the live configuration file.

Chapter 13

Service Monitoring

The Symbiosis system is comprised of several distinct components, which we've documented throughout the course of this reference:

- The MySQL database server.
- Exim & Dovecot servers for handling email.
- Apache for serving websites.
- The FTP server, proftpd
- The inotify cron daemon, incron.
- Prosody an XMPP server

Each of these services runs in an independent fashion, and it is possible under certain circumstances that these services might fail, or stop themselves.

To handle the case of services failing to execute normally we've included an automated service checker as part of the Symbiosis system. The service checker will check upon the health of your system, by default once every two minutes, and it will automatically restart any services which have failed.

The 'symbiosis-monit` command is responsible for testing each of the available services, and restarting the failed ones. By default it is executed every two minutes, such that it may respond quickly to failures. It will also stop services that are not required. For example if the machine is not configured to scan any domains' mail, then SpamAssassin will be stopped.

At any time you wish to check upon the health of your system you may launch it manually, when connected to your server via SSH.

```
admin@example:~$ sudo symbiosis-monit
= Symbiosis service test report ========================================

 Tests conducted on symbiosis-test.default.uk0.bigv.io.
 Tests started at 2015-10-15T12:50:35+01:00.

 * apache2: PASSED
 * cron: PASSED
 * dovecot: PASSED
 * exim4: PASSED
 * incrond: PASSED
```

```
 * mysqld: PASSED
 * prosody: PASSED
 * pure-ftpd: PASSED
 * spamassassin: PASSED
 * sshd: PASSED

 Tests finished at 2015-10-15T12:50:36+01:00.

= End of service test report ============================================
```

In this case all services are working correctly, so "PASSED" was reported instead of the failing "FAILED". The possible output status are:

FAILED The service failed.

PASSED The service appears to be running correctly.

Chapter 14

Glossary

BSD, Berkeley System Distribution

A family of Unix versions developed by Bill Joy and others at the University of California at Berkeley, originally for the DEC VAX and PDP-11 computers, and subsequently ported to almost all modern general-purpose computers. BSD Unix incorporates paged virtual memory, TCP/IP networking enhancements and many other features [FOLDOC].

DKIM, Domain Keys Identified Mail

This adds a DKIM signature to each outbound email message on a system which can then be verified by recipients. Recipient SMTP servers will look up the DKIM selector of the mail, and verify that the key the mail is signed with matches the public key in DNS.

DNS, Domain Name System

This system is used to convert IP Addresses into hostnames. It is also used to determine where mail should be routed for a domain.

FTP, File Transfer Protocol

FTP used to be used to transfer large files over the internet. It is an archaic protocol.

FTPS, File Transfer Protocol Secure

FTPS is an extension to FTP that allows encryption using TLS or SSL. It is not to be confused with SFTP, which is a subsystem of SSH.

HTML, Hypertext Markup Language

A system to mark up documents. It is the most common format used for documents on the world-wide web, and is the format that web browsers display.

HTTP, Hypertext Transfer Protocol

This protocol was originally used to transfer HTML documents between machines connected to the internet. It has become the standard protocol for transferring all types of documents over the world-wide web.

IMAP, Internet Message Access Protocol

The Internet Message Access Protocol (IMAP) is one of the two most prevalent Internet standard protocols for e-mail retrieval, the other being the Post Office Protocol (POP). Virtually all modern e-mail clients and mail servers support both protocols as a means of transferring e-mail messages from a server.

IP, Internet Protocol

The network layer for the TCP/IP protocol suite widely used on Ethernet networks, defined in STD 5, RFC 791. IP is a connectionless, best-effort packet switching protocol. It provides packet routing, fragmentation and re-assembly through the data link layer.

+ IPv4 is the version in widespread use and IPv6 was just beginning to come into use in 2000 but was still not widespread by 2008 [FOLDOC].

IP Address

IP addresses come in two flavours, reflecting the two versions of *IP* used.

+ An IPv4 address is a 32 bit number generally represented as a dotted quad e.g. 10.20.30.40. There is a limit of just under 4.3 billion IPv4 addresses, which is slowly being reached, which necessitated the invention of IPv6.

+ An IPv6 address is a 128 bit number, generally represented as a hexadecimal number, split into nibbles of up to four digits, separated by colons, e.g. 2001:41c8:12::34. There are up to 2^{128} or 3×10^{38} addresses available in IPv6.

ISP, Internet Service Provider

A company which provides other companies or individuals with access to, or presence on, the Internet. Most ISPs are also Internet Access Providers; extra services include help with design, creation and administration of World-Wide Web sites, training and administration of intranets and domain name registration [FOLDOC].

ManageSieve

ManageSieve is a protocol that is allows *Sieve* filters to be managed remotely, testing any filters before allowing them to be used.

MTA, Mail Transfer Agent

A mail transfer agent is a computer process or software agent that transfers electronic mail messages from one computer to another, in single hop application-level transactions. A MTA implements both the client (sending) and server (receiving) portions of the Simple Mail Transfer Protocol.

MUC, Multi User Chat

A Multi User Chat is a feature of XMPP allowing many users to converse in the same window. This is often used to ease communication between groups in different offices, and for the sake of ease can be thought of as the point at which mailing lists and instant messages meet.

NTP, Network Time Protocol

A protocol built on top of TCP/IP that assures accurate local timekeeping with reference to radio, atomic or other clocks located on the Internet. This protocol is capable of synchronising distributed clocks within milliseconds over long time periods.[FOLDOC].

PHP

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. [PHPNET]

POP3, Post Office Protocol 3

Version 3 of the Post Office Protocol. POP3 is defined in RFC 1081, written in November 1988 by Marshall Rose, which is based on RFC 918 (since revised as RFC 937). POP3 allows a client computer to retrieve electronic mail from a POP3 server via a (temporary) TCP/IP or other[?] connection. It does not provide for sending mail, which is assumed to be done via SMTP or some other method [FOLDOC].

Secure File Transfer Protocol, SFTP

SFTP is a file transfer protocol which involves using an SSH server to manage the file uploads. It is secure in the sense that file contents are encrypted during transfer, and that plain-text passwords are never sent over the internet. SFTP is the logical successor to FTP, which is less secure, and more complex to firewall.

Sieve

Sieve is a language that can be used to filter email messages. It is a powerful language that provides a safe environment for filtering to occur during mail delivery, allowing messages to be delivered directly into mailboxes configured by the user.

SMTP, Simple Mail Transfer Protocol

A protocol defined in STD 10, RFC 821, used to transfer electronic mail between computers, usually over Ethernet. It is a server to server protocol, so other protocols are used to access the messages [FOLDOC].

SPF, Sender Policy Framework

An anti-spam measure designed to let domain administrators choose how mail sent on their domain's behalf will be treated by recipients, which can help send spoofed mail to spam and protect your domain's reputation.

SSH, Secure Shell

A Unix shell program for logging into, and executing commands on, a remote computer. ssh is intended to replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel [FOLDOC].

SSL, Secure Sockets Layer

A protocol designed by Netscape Communications Corporation to provide secure communications over the Internet using asymmetric key encryption. SSL is layered beneath application protocols such as HTTP, SMTP, Telnet, FTP, Gopher and NNTP and is layered above the connection protocol TCP/IP. It is used by the HTTPS access method [FOLDOC].

TCP, Transmission Control Protocol

The most common transport layer protocol used on Ethernet and the Internet. It was developed by DARPA.

TCP is the connection-oriented protocol built on top of Internet Protocol (IP) and is nearly always seen in the combination TCP/IP (TCP over IP). It adds reliable communication and flow-control and provides full-duplex, process-to-process connections.

TCP is defined in STD 7 and RFC 793 [FOLDOC].

TLS, Transport Layer Security

A protocol designed to allow client/server applications to communicate over the Internet without eavesdropping, tampering, or message forgery.

TLS is defined in RFC 2246 [FOLDOC].

UDP, User Datagram Protocol

Internet standard network layer, transport layer and session layer protocols which provide simple but unreliable datagram services. UDP is defined in STD 6, RFC 768. It adds a checksum and additional process-to-process addressing information [to what?]. UDP is a connectionless protocol which, like TCP, is layered on top of IP.

UDP neither guarantees delivery nor does it require a connection. As a result it is lightweight and efficient, but all error processing and retransmission must be taken care of by the application program [FOLDOC].

URL, Uniform Resource Locator

A Uniform Resource Locator (URL) is a Uniform Resource Identifier (URI) that specifies where an identified resource is available and the mechanism for retrieving it. In popular usage and in many technical documents and verbal discussions it is often incorrectly used as a synonym for URI. The best-known example of a URL is the "address" of a web page e.g. http://www.example.com [WIKIPEDIA_URL].

XMPP, Extensible Messaging and Presence Protocol

A protocol enabling instant messaging, contact list maintenance, and presence information. Addresses usually take the same form as an email address, eg, user@domain.tld. Various common extensions exist, including file transfer, voice and video (`Jingle`), service discovery, and multi user chat. Federation is another key feature of XMPP, which allows any user of XMPP to contact any other user, provided they are able to connect that user's XMPP server. XMPP is not limited to chat, but can also be used to deliver push notifications, file sharing, and identity services.

Chapter 15

Bibliography

Bibliography

[1] [FOLDOC] Denis Howe (ed). ‘The Free On-line Dictionary of Computing’, http://foldoc.org/

[2] [PHPNET] The PHP Group. ‘PHP: Hypertext Preprocessor’, http://php.net/

[3] [WIKIPEDIA_URL] Wikipedia contributors. ‘Uniform Resource Locator’, Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Uniform_Resource_Locator&oldid=367676813 (downloaded 2010-06-10).

Appendix A

GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical

connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties — for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See Copyleft.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright © YEAR YOUR NAME

Permission is granted to copy, distribute and/or modify this document under the
terms of the GNU Free Documentation License, Version 1.3 or any later version
published by the Free Software Foundation; with no Invariant Sections, no
Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in
the section entitled "GNU Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with... Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts
being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Chapter 16

Index