

Implementing Postgrey on Mac OS X 10.5.x/10.4.x

- 1. - Introduction**
- 2. - Requirements**
- 3. - Getting and installing the required components**
- 4. - Getting and installing Postgrey**
- 5. - Using Postgrey to greylist incoming messages**
- 6. - Caveats - READ this chapter!**

DISCLAIMER: The author(s) claim(s) no responsibility for any damage that may occur from the use of any information found here or found on links followed from this document. The author(s) will take credit for the good stuff though.

All items and/or companies mentioned on this and other pages are or may be trademarks, registered trademarks, or service marks of the respective companies or individuals.

1. - Introduction

The purpose of this document is to provide instructions on how to implement Postgrey on OS X 10.5.x and OS X 10.4.x Server.

You will not find many explanations as to why something is done one way or the other. Also, I will not discuss whether Postgrey is useful or not. This is a decision you must make for yourself. There are enough discussions about this available on the internet.

Postgrey is a Postfix policy server implementing greylisting developed by David Schweikert. The official website can be found here: <http://postgrey.schweikert.ch/>

Postgrey functionality depends on several Perl modules and scripts to be installed.

Postgrey works as a policy server in conjunction with Postfix.

This document will require you to use the command line. If you do not feel comfortable with using the command line, you should look for a ready made installer package or for somebody to assist you.

This document is written for Mac OS X 10.5.x/10.4.x. It does not apply to 10.3.x.

DISCLAIMER: Whatever you do based on this document, you do it at your own risk! Just in case you haven't understood: Whatever you do based on this document, you do it at your own risk!

This tutorial has been tested on a standard Mac OS X 10.5.x/10.4.x Server installation. If you have already tinkered with your system, be aware that things might differ. It is impossible for me to foresee all changes that one might have applied to a server.

This tutorial contains step-by-step instructions for terminal. Although you could just type them in line by line, it is recommended you have a basic understanding of the terminal.

2. - Requirements

Before you get started, you need to make sure some basic requirements are met:

- You have made a backup or clone of your system.
- You have the latest version of Apple's Developer Tools (Xcode 2.5 or higher for 10.4.x and XCode 3.1.2 or higher for 10.5.x) installed.
Dev Tools are available as a free download from Apple's Developer Connection: <http://connect.apple.com/>
- You do have a backup or clone.
- You are running Mac OS X 10.4.x/10.5.x Server or greater
- Although not mandatory, for Mac OS X 10.4.x it is STRONGLY (very STRONGLY) recommended you first update amavisd-new and SpamAssassin.
Some of the Perl modules necessary, may conflict with older versions of amavisd-new and SpamAssassin. Instructions for updating amavisd-new and SpamAssassin on OS X 10.4.x server can be downloaded from <http://osx.topicdesk.com/>
- Familiarity with a command line editor or alternatively a GUI plain text editor (do NOT use Word or similar)
- Not a requirement, but it is recommended you subscribe to our newsletter(s) or follow us on Twitter to be informed when updated versions of this and other tutorials become available:
<http://osx.topicdesk.com/newsletter/>
<http://twitter.com/topicdesk/>

3. - Installing missing and required components

As mentioned, you will need a few perl modules to be able to use Postgrey. This chapter will guide you through getting and installing them.

So let's get going:

Make sure you are logged in as root (or alternatively use sudo).

Install the required modules by issuing the following commands (*in oblique type*). Issue them one after the other making sure you do not miss any dots or slashes. Also note that the download URLs given may change in the future. In that case just replace the URLs in this document with the current ones.

NOTE: Lines wrapping without line spacing are a single command.

The easiest way to install them is by using CPAN. To do so issue:

```
sudo perl -MCPAN -e shell
```

If you have never used CPAN before you will be prompted to supply a few parameters. Just accept the default values. Once done, you should see the CPAN prompt (*cpan >*):

When at the CPAN prompt issue:

```
o conf prerequisites_policy ask
```

This will prompt you when a module relies on other prerequisites that have to be installed first. You should allow it to go ahead if asked.

Now you are ready to install the missing modules. Be aware that some modules already exist on your server, but are outdated so it is best to install them all.

Just issue:

```
install Net::Server
```

```
install IO::Multiplex
```

```
install BerkeleyDB
```

This will install the modules and bring you back to the CPAN prompt.

Now issue

```
exit
```

to exit CPAN.

NOTE: It is possible that some of the modules will not install. In that case use "*force install*" instead of "*install*" at the CPAN prompt.

NOTE: If you had previously tried to use CPAN without having the Developer Tools installed, you will need to make sure that Developer Tools are now correctly installed and you will also need to re-configure CPAN. To do so get to the CPAN prompt and issue:

```
o conf init
```

You will be prompted to supply a few parameters. Just accept the default values.

NOTE: It may be that you have an old/incorrect version of db.h on your system. Should you have difficulties installing BerkeleyDB via CPAN, then please see the Caveats section!

4. - Getting and installing Postgrey

This chapter will guide you through getting and installing Postgrey.

Postgrey is written and maintained by David Schweikert. The official website can be found here: <http://postgrey.schweikert.ch/>

There are other tools and combinations to implement greylisting available out there, but this one works best and makes most sense for OS X Server.

So let's get going:

Make sure you are logged in as root (or alternatively use sudo).

Install the latest version of Postgrey by issuing the following commands (*in oblique type*). Issue them one after the other making sure you do not miss any dots or slashes. Also note that the download URLs given may change in the future. In that case just replace the URLs in this document with the current ones.

NOTE: Lines wrapping without line spacing are a single command.

```
mkdir -p /SourceCache
```

```
cd /SourceCache
```

```
sudo curl -O http://postgrey.schweikert.ch/pub/  
postgrey-1.32.tar.gz
```

```
sudo tar xzf postgrey-1.32.tar.gz
```

```
cd postgrey-1.32
```

```
cp postgrey /usr/local/sbin
```

```
chmod -R 755 /usr/local/sbin/postgrey
```

```
mkdir -p /var/postgrey
```

NOTE: Instead of creating a new system user for Postgrey, we will use the existing user for other anti-spam measures. 10.4.x and 10.5.x use different system users. 10.4.x uses user "clamav", while 10.5.x uses user "_amavisd". Choose only one of the following two commands based on your OS version:

```
chown -R clamav:clamav /var/postgrey (for 10.4.x)
```

```
chown -R _amavisd:_amavisd /var/postgrey (for 10.5.x)
```

```
cp postgrey_whitelist_clients /etc/postfix/  
postgrey_whitelist_clients
```

```
cp postgrey_whitelist_recipients /etc/postfix/  
postgrey_whitelist_recipients
```

Next we need to set up a property list for launchd, so that Postgrey gets started on system startup.

```
cd /System/Library/LaunchDaemons
```

```
sudo touch ch.schweikert.postgrey.plist
```

Above command created a new empty property list. Edit

```
/System/Library/LaunchDaemons/  
ch.schweikert.postgrey.plist
```

with your favorite editor (pico, vi, etc.) and add the following content:

NOTE: 10.4.x and 10.5.x use different system users for running amavisd, clamav and spamassassin. 10.4.x uses user "clamav", while 10.5.x uses user "_amavisd". Below snippet is ready for 10.4.x. Please substitute "clamav" with "_amavisd" for 10.5.x (2 occurrences)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST
1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>ch.schweikert.postgrey</string>
  <key>OnDemand</key>
  <false/>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/local/sbin/postgrey</string>
    <string>--inet=127.0.0.1:10029</string>
    <string>--dbdir=/var/postgrey</string>
    <string>--user=clamav</string>
    <string>--group=clamav</string>
  </array>
  <key>ServiceIPC</key>
  <false/>
  <key>UserName</key>
  <string>root</string>
</dict>
</plist>

```

(NOTE: `<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">` is a single line.)

The last step is to start Postgrey by issuing:

```

sudo /bin/launchctl load -w /System/Library/
LaunchDaemons/ch.schweikert.postgrey.plist

```

You are now all set and should have Postgrey installed and running.

The next step is to configure Postfix to use Postgrey for greylisting of incoming messages.

5. – Using Postgrey to greylist incoming messages

As mentioned, we will use Postgrey in combination with Postfix to implement greylisting of incoming messages.

To do so, you will need to modify a parameter in your Postfix configuration.

Edit `/etc/postfix/main.cf` and look for
`smtpd_recipient_restrictions = ...`

remove `permit` at the end and instead add:
`check_policy_service inet:127.0.0.1:10029`

make sure `smtpd_recipient_restrictions` contains
`reject_unauth_destination` BEFORE `check_policy_service`

The result should look something like:

```
smtpd_recipient_restrictions = ...,  
reject_unauth_destination, ..., check_policy_service  
inet:127.0.0.1:10029
```

When done, reload Postfix by issuing:

```
sudo postfix reload
```

You are now all set and should have greylisting in place.

If you would like to adjust the senders and/or recipients where greylisting is bypassed, you can change them inside:

```
/etc/postfix/postgrey_whitelist_clients  
/etc/postfix/postgrey_whitelist_recipients
```

6. - Caveats

The most frequent issues to watch out for are:

- a) Incompatible perl modules and typos made
- b) Long lines seen as multiple lines. Watch for incorrect line breaks
- c) If you have modified any paths and or environment variables, make sure you check them against above instructions.

NOTE: Should you have difficulties installing BerkeleyDB via CPAN, please follow these instructions first (only then!):

```
mkdir -p /SourceCache
```

```
cd /SourceCache
```

```
curl -O http://download.oracle.com/berkeley-db/  
db-4.2.52.tar.gz
```

```
sudo tar xzf db-4.2.52.tar.gz
```

```
cd db-4.2.52
```

```
cd build_unix
```

```
sudo ../dist/configure --prefix=/usr
```

```
cp db.h /usr/include/
```

That's all folks.

Hope this helps.

Have fun,

Alex

Doc. Version 1.0.2, 6.7.2009

Athanasios Alexandrides

Lugano, Switzerland

tutorials -at- topicdesk.com
