Professional FTP Daemon FAQ

Mark Lowes

<hamster@vom.org.uk>

Table of Contents

Preface

This document sets out many of the FAQs related to the installation, functioning and configuration of ProFTPD. It also provides some guidance on policy and security issues.

Chapter 1. Introduction to ProFTPD

1. What is ProFTPD?

ProFTPD is a ftp server primarily written for the various unix variants though it will now compile under win32. It has been designed to be much like Apache in concept taking many of the ideas (configuration format, modular design, etc) from it.

2. What is the current version?

- Stable: 1.2.4
- Release Candidate: 1.2.5rc1
- Unstable: 1.3.x

3. Version numbering scheme

At the moment there is a little irrationality in the numbering scheme however it can be summarised as follows

1.0.x
: This is the previous stable version.

1.1.x
: Development code

1.2.0prex
: Pre−release testing versions, development code.

1.2.0rcx
: Release candidate code, these releases are pretty much bug free and are testing releases prior to the final stable code.

1.2.x

This will be the stable cycle with the final .x being the incremental patches to fix bugs discovered after the release version is issued.

1.3.x

1.3.x is in development, nightly tarballs from the CVS tree are available from ftp.proftpd.org and it's mirror sites.

4. Website & documentation

http://www.proftpd.org is now online and contains copies of this FAQ, other documentation resources and information on the project. The documentation is being brought back into shape at the moment, the configuration on the website is now approaching where it should be but more work is required and is ongoing. There are a number of geographic mirror sites, see http://www.proftpd.org for details or try www.<isocode>.proftpd.org (ie www.uk.proftpd.org).

Helping with documentation

Writing documentation is time consuming and requires some work but it's not actually difficult. Get the source code from CVS, run "ShowUndocumented" in the doc directory. This will list what needs work. Grep through the code in the looking for something like

```
CHECK_CONF(cmd,CONF_ROOT|CONF_VIRTUAL|CONF_ANON|CONF_GLOBAL)
```

to figure out where the directive is valid (server config, <VirtualHost>, <Anonymous>, <Global> for the above example). Once you think you understand what it does, test, play, break (if possible). Then copy the format in Configuration.html and add the new documentation.

Once the documentation is complete run

```
cvs diff -uw Configuration.html > Configuration.html.patch
```

and submit it via the bug reporting system.

5. Bug reporting

Bug reports should be made via http://bugs.proftpd.org/ which uses the bugzilla tracking system. Patches should be mailed to the ProFTPD−Devel mailing list or MacGyver directly.

6. I've found a security hole

Please report all security problems with the code to `<security@proftpd.org>` before releasing the information into the public domain. It would be appreciated if you give the core team a few days to put together a patch and/or new release to address the issue.

Please adhere to the proceedures and timescales given in the RF Policy document http://www.wiretrip.net/rfp/policy.html, this will give the core development team a chance to get a fix or workaround in place before the problem becomes fully public domain.

7. Downloading

There are two main methods of getting the software. Downloading a compressed tarball or rpm (there is also a Debian package available in the main distribution) from proftpd.org or from a mirror site, alternatively if you wish to run the latest bleeding edge code then collecting from the cvs server is the best method.

Mirror sites

There is a complete and maintained list of ftp mirror sites available from http://www.proftpd.org/download.html

CVS

cvs –d :pserver:anonymous@cvs.proftp.sourceforge.net:/cvsroot/proftp login (Hit Enter when prompted for a password.)

Then do:

cvs –d :pserver:anonymous@cvs.proftp.sourceforge.net:/cvsroot/proftp –z3 co proftpd

To obtain the latest/greatest updates, just hop into the proftpd directory and do: cvs update

A couple of sites generate downloadable tarballs of the latest CVS code to make obtaining the test code easier.

8. Mailing lists

There are a number of mailing lists for ProFTPD

Announce

proftpd–announce@proftpd.org

This is a very low traffic list where only ProFTPD announcements/changes will be announced. Subscribe by sending a message to `<proftpd-announce-request@proftpd.org>` with "subscribe" in the subject.

Web interface: https://lists.sourceforge.net/lists/listinfo/proftp–announce

Users

proftp–user@proftpd.org

This is intended to the the user support channel for the software, in most likelihood this is going to be a high traffic list and slightly chatty. Please read the FAQ, the documentation and the list archives before posting a question.

Subscribe by sending a message to `<proftpd-user-request@proftpd.org>` with "subscribe" in the subject.

Web interface: https://lists.sourceforge.net/lists/listinfo/proftp–user

Development

proftpd–devel@proftpd.org

This list is intended for discussion of development–related issues of ProFTPD, and feature design. It is NOT intended to be a "user help" group.

Subscribe by sending a message to `<proftpd-devel-request@proftpd.org>` with "subscribe" in the subject.

Web interface: https://lists.sourceforge.net/lists/listinfo/proftp–devel

Archives

The mailing list archives can be found at:

- http://www.proftpd.org/proftpd–announce–archive/
- http://www.proftpd.org/proftpd–l–archive/
- http://www.proftpd.org/proftpd–devel–archive/

Unsubscribing

Before posting to any of the lists or mailing the list admins please try and remove yourself first. Either by emailing <listname>–request@lists.sourceforge.net with the subject "unsubscribe" or visiting the web interface and unsubscribing from there.

I've (lost / never had) a password to the interface. Easy, enter the address you are subscribed to the list as into the form and hit the "email me my password" button.

9. Copyright Issues

The software is currently distributed under the GNU General Public License (version 2 or later) as published by the Free Software Foundation. Copyright is held by Public Flood Software.

Chapter 2. Compilation and installing

1. What platforms will it compile on?
2. Why not libc5 on Linux?
3. CVS

1. What platforms will it compile on?

There have been reports of ProFTPD compiling on all the following platforms (and versions).

- Linux 2.0.x & 2.2.x (glibc 2.x only) & 2.4.x
- BSDI 3.1 & 4.0
- IRIX 6.2, 6.3, 6.4, 6.5
- Solaris 2.5.1, 2.6, 2.7, 8 (Sparc)
- AIX 3.2 & 4.2
- OpenBSD 2.2/2.3
- FreeBSD 2.2.7
- Digital UNIX 4.0A
- DEC OFS/1

2. Why not libc5 on Linux?

There are several known problems with libc5−based systems, including improperly implemented library routines (vsprintf and vsnprintf are examples). There are known problems with the resolver library. For these reasons and others lib5 is not being supported at all, the latest versions of the major distributions (inc Debian, Redhat and Suse) are all glibc.

3. CVS

CVS (Concurrent Versions System), is a version control system which allows multiple developers (scattered across the same room or across the world) to maintain a single codebase and keep a record of all changes to the work.

The CVS repository for ProFTPD is available for non−developers in read−only mode, however this code is right on the bleeding edge and is not guaranteed to even compile let alone work. Access to CVS is given to allow important security patches out into the wild and to allow users and interested users to test out the latest changes on real systems.

Recommended ~/.cvsrc settings

```
cvs −z 3
update −Pd
diff −u
```

Where can I get information on cvs?

CVS is produced by Cyclic Software (http://www.cyclic.com/) and details on CVS can be found on their website. The CVS documentation is clear, detailed and above all heavy when printed. I'd recommend reading it if you're planning on using CVS a lot.

4. How do I get debug output

The easiest way is to fire up proftpd manually from the command line with the debug level cranked up.

```
/usr/local/sbin/proftpd −d9 −n
```

This will result in maximal debug output direct to the console. Warning, this can get messy on a busy server, for testing I would suggest copying the config and altering the port the server binds to and then testing.

5. Patches

Any patches should be submitted in Universal format, this makes integrating them into the main cvs source a lot easier. When generating a diff against the current cvs source use "cvs diff −uw" to generate the patch.

```
cvs diff −u filename > filename.patch

  or

cvs diff −u > bigger.patch
```

Patches that add configuration directives without proper documentation. Will be rejected. New features without documentation are less than useless to the community at large.

6. Using non−default modules

Simply configure ProFTPD with

```
./configure −−with−modules=mod_module1:mod_module2:mod_module3
make
make install
```

7. Plans for next version (1.3.x)

The new development series will be 1.3.x, using the same number scheme as the linux kernel developers. The targets/goals are:

- refining/redefining the module API to make it more extensible and useful.
- dynamic modules
- security APIs and implementations
- mod_ls rewrite
- Implementing some security−related RFCs
- Creating a web and GUI configuration interface to ProFTPD

2.0.x will be the production release of the 1.3.x development set.

8. NT Support

If/when a port is undertaken for NT, it will only be after a near complete rethinking of ProFTPD. This is planned for 2.0 and onwards.

9. New features/modules

While anything new is welcomed it's probably better to at least float the idea first on the devel mailing list to ensure that someone else isn't already hacking on it. Also when submitting the patch or module for inclusion into the ProFTPD source full documentation is needed.

Suggestions made for future development

- GUI based configuration tool

Chapter 3. Compatibility and Integration

1. SQL

ProFTPD has support for authentication and logging via SQL databases using the mod_sql module as supplied in the main distribution.

2. SSH

There is a mini−HOWTO at http://www.castaglia.org/proftpd/doc/ detailing how to tunnel ftp connections over ssh.

3. sendfile()

sendfile() is a system call which streamlines the copying of data between the disk and the tcp socket. The call copied from the page cache directly rather than requiring a kernel −> user space −> kernel space copy for every read() and write() call. Generally the advantages are only felt on heavily loaded servers. The call is supported in ProFTPD for Linux and FreeBSD.

Linux 2.0.x

sendfile is not supported under 2.0.x, this is not an issue when compiling for 2.0.x on a 2.0.x system. However when compiling on a 2.2.x system for use on 2.0.x use the −−disable−sendfile flag.

Runtime detection of sendfile()

Johnie Ingram (aka netgod)'s:
http://www.proftpd.org/proftpd−devel−archive/99−10/msg00073.html

John Pierce <hawkfan@pyrotechnics.com>
http://www.proftpd.org/proftpd−devel−archive/99−10/msg00112.html

Problems with sendfile

There appear to be a number of problems with sendfile() particularly with the directives and features which require accurate determination of filesize. Such as the Rate* functions and downloading large files, the best advice at the moment appears to be to disable sendfile by default (−−disable−sendfile).

Sendfile() also appears to be the source of a number of file corruption problems.

4. IPv6

There is currently no official support for IPv6 within the 1.2.x code tree, however there is an http://www.t17.ds.pwr.wroc.pl/~misiek/ipv6/ and more comprehensive support will probably be developed during the 1.3.x development cycle.

5. Filename case sensitivity

ProFTPD is utterly dependant on the underlying OS to handle filename case sensitivity. If the underlying OS is case sensitive then ProFTPD will be, there are currently no plans for a module to handle this.

6. FXP

FXP is capable of bouncing data between websites. There have been a number of reports of problems in configuring ProFTPD to function cleanly with this program (http://flashfxp.skuz.net/).

To support FXP when connecting as a user place "AllowForeignAddress on" in the Global or VirtualHost context.

To support FXP when connecting as anon "AllowForeignAddress on" must be placed in the Anonymous context.

The config will happily support "AllowForeignAddress on" in multiple places within the config.

Chapter 4. Common Running problems

1. ProFTPD doesn't seem to work.

Starting ProFTPD in standalone mode it doesn't show in "ps" It could be many things, possibly something like not running ProFTPD as root (it needs to be run as root initially, but will switch to a non−privileged user). Regardless, ProFTPD logs all errors via the standard syslog mechanism. You need to check your system logs in order to determine what the problem is.

It doesn't work!

There are many times when there's a completely random problem which appears to be insoluble. The best place to ask for help is definately the mailing list (proftpd−l) but it's not productive to ask for help without giving enough information for intelligent debugging.

Have you?

- ♦ Checked your logs
- ♦ Tried the server in debug mode
- ♦ Read the FAQ?
- ♦ Checked the mailing list archive?
- ♦ Are you running the latest version?

When posting try giving enough information, this might include but not be limited to.

- OS and server version (proftpd −vv)
- List of included modules (proftpd −l)
- Appropriate log extracts
- Output fom debug mode
- Configration fragment

2. "inet_create_connection() failed: Operation not permitted".

You aren't starting ProFTPD as root, or you have inetd configured to run ProFTPD as a user other than root. The ProFTPD daemon must be started as root in order to bind to tcp ports lower than 1024, or to open your shadow password file when authenticating users. The daemon switches uid/gids to the user and group specified by the User/Group directives during normal operation, so a "ps" will show it running as the user you specified.

3. Unable to bind to port/Address already in use

0.0.0.0 is INADDR_ANY, which means to bind to any interface. The "address in use" will normally mean that something has already bound to that address.

Under linux it is possible to run:

```
fuser −n tcp 21
```

to get the PID of the process currently bound to port ProFTPD is configured to run as.

The most common cause is that ProFTPD is configured standalone and inetd is still configured for port 21. Comment out the line starting "ftp" in /etc/inetd.conf and restart (killall −HUP inetd or something similar should do the trick) and try again.

4. "Fatal: Socket operation on non−socket"

You have ProFTPD configured to run in inetd mode rather than standalone. In this mode, ProFTPD expects that it will be run from the inetd super−server, which implies that stdin/stdout will be sockets instead of terminals. As a result, socket operations will fail and the above error will be printed. If you wish to run ProFTPD from the shell, in standalone mode, you'll need to modify your proftpd.conf configuration file and add or edit the ServerType directive to read:

```
ServerType standalone
```

5. "Fatal: unable to determine IP address of "hostname:

The hosting machine has a poorly configured hostname setup to the point where the resolver library cannot determine the IP from the name. Solutions include, fixing the DNS for the domain, fixing the hostname, fixing the /etc/hosts file. Which one works for you will largely depend on your OS and exactly what is wrong.

6. I'm having problems with FTP clients behind firewalls

The FTP Specification defines that two sockets should be used for all communications. The first runs over port 21 and is the control channel over which all commands and response codes are sent. Whenever data is required to be transfered, for example for a file download, a directory listing etc etc. A second channel is created on demand, this socket can take one of two forms.

non−Passive

The server end of the data socket uses port 20. This is nice and easy to work into a firewall configuration.

Passive

The port at either end is dynamically allocated. This is virtually impossible to cater for in a firewall configuration given that the port mapping will be different for every data connection.

The solution is to force the users to configure their clients to use the non−passive mode (ie port 20)

7. Can I run more that one VirtualHost on a single IP?

No, or at least not in the HTTP/1.1 manner of virtual hosting. This is an inbuilt limitation of the current FTP RFC., unlike the HTTP/1.1 spec there is no mechanism comparable to the "Host: foo.bar.com" HTTP header for specifying which host the connection is for. Therefore the only method for determining which VirtualHost the connection is destined for is by the destination IP.

The one exception to this is if you host multiple servers on the same IP but using different ports, however this requires that the connecting client uses a non−standard port and therefore is probably not a good solution for mass hosting.

Is there anything in the pipeline to fix this?

There is a draft standard http://search.ietf.org/internet−drafts/draft−ietf−ftpext−mlst−12.txt with the IETF which extends and improves on the FTP specification including support for a HOST command. However given that the IP crunch is coming from websites and not virtual ftp servers this is unlikely to be pushed through any time soon.

8. How do I run ProFTPD from inetd?

Find the line in /etc/inetd.conf that looks something like this:

```
ftp stream tcp nowait root in.ftpd in.ftpd
```

Replace it with:

```
ftp stream tcp nowait root in.proftpd in.proftpd
```

Then, find your inetd process in the process listing and send it the SIGHUP signal so that it will rehash and reconfigure itself. You may also need to add in.ProFTPD to hosts.allow on your system.

9. Can I use tcp−wrappers with ProFTPD?

Yup. Although ProFTPD has built−in IP access control (see the Deny and Allow directives), many admins choose to consolidate IP access control in one place via in.tcpd. Just configure ProFTPD to run from inetd as any other tcp−wrapper wrapped daemon and add the appropriate lines to hosts.allow/deny files.

If running ProFTPD in standalone mode, mod_wrap can be used to direct the server to use the normal hosts.allow/deny files.

10. Can I run an FTP server on a non−standard port?

Yes. Use a <VirtualHost> block with your machine's FQDN (Fully Qualified Domain Name) or IP address, and a Port directive inside the <VirtualHost> block. For example, if your host is named "myhost.mydomain.com" and you want to run an additional FTP server on port 2001, you would:

```
...
<VirtualHost myhost.mydomain.com>
Port 2001
...
</VirtualHost>
```

11. Can control upload/download ratios?

Yes the mod_ratio module provides for doing just this.

The ratio directives take four numbers: file ratio, initial file credit, byte ratio, and initial byte credit. Setting either ratio to 0 disables that check.

The directives are HostRatio (matches FQDN, wildcards allowed), AnonRatio (matches password entered at login), UserRatio (accepts "*" for "any user"), and GroupRatio.

```
Ratios on                                    # enable module
UserRatio ftp 0 0 0 0
HostRatio master.debian.org 0 0 0 0          # leech access (default)
GroupRatio proftpd 100 10 5 100000           # 100:1 files, 10 file cred 5:1 bytes, 100k byte cred
AnonRatio billg@microsoft.com 1 0 1 0        # 1:1 ratio, no credits
UserRatio * 5 5 5 50000                      # special default case
```

This example is for someone who (1) has downloaded 1 file of 82k, (2) has uploaded nothing, (3) has a ratio of 5:1 files and 5:1 bytes, (4) has 4 files and 17k credit remaining, and (5) is now changing directory to /art/nudes/young/carla. The initial credit, not shown, was 5 files and 100k (UserRatio * 5 5 5 100000).

Version 2.0 and above of this module integrate with mod_sql.

Limitations of mod_ratio

It appears that the ratio limits in mod_ratio are only maintained on a per session basis and there is no ongoing tracking of usage.

12. Slow logins

This is probably caused by a firewall or DNS timeout. By default ProFTPD will try to do both DNS and ident lookups against the incoming connection. If these are blocked or excessively delayed a slower than normal login will result. To turn off DNS and ident use:

```
UseReverseDNS off
IdentLookups                    off
```

IdentLookups and tcpwrappers ***

13. Lots of "FTP session closed" messages

Oct 7 12:30:48 salvage2 proftpd[8874]: FTP session closed. Oct 7 12:30:48 salvage2 proftpd[8874]: FTP session closed. Oct 7 12:30:48 salvage2 proftpd[8874]: FTP session closed. Oct 7 12:30:48 salvage2 proftpd[8874]: FTP session closed.

The above log extract is likely to be caused by a local monitoring system or a particularly aggressive DoS attack. Most service monitoring systems try opening the ftp port on the target server to detect whether it is active and running. Most of the time these tests are followed by an immediate "QUIT" or disconnection.

TCPdump/TCPshow on the server in question should show which machine on your network is is generating these connections.

14. How do I see who is connected?

The ftpwho command lists the state of each ftp connection to the server and what it's current activity is. However this does not detail the connection information on a virtual by virtual basis.

15. Can I force ProFTPD to listen on only one IP?

Sort, of it's not quite as clean as the socket binding under Apache but the principle works something like this.

Standalone mode

To listen on the primary IP of a host use the SocketBindTight directive

To listen on a interfaces which are not the primary host interface use the SocketBindTight directive, place your server configuration in a <VirtualHost ftp.mydomain.com> block and use "Port 0" for the main host configuration and and "Port 21" inside the VirtualHost block.

inetd

There are two approaches possible, the first is to use the patch from Daniel Roesen <droesen@entire–systems.com> (check the mailing list archives).

The second method is to run ProFTPD from xinetd (http://synack.net/xinetd/), a more advanced replacement of inetd. An entry for this in xinetd.conf would be something like this:

```
service ftp
{
        disable = no
        flags                   = REUSE
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/proftpd
        log_on_success          += DURATION USERID
        log_on_failure          += USERID
        nice                    = 10
        #bind                   = [IP to bind to]
}
```

16. "FTP server shut down ... please try again later."

Check for /etc/shutmsg and delete it.

17. How do I shutdown the server without killing proftpd?

ftpshut, allows the server to disallow connections with a message without actually taking down the service. The shutdown can be scheduled for a point in the future or right now, existing connections can be allowed to finish, or be terminated now. Re−enabling is done by removing the /etc/shutmsg file.

18. Is is possible to shutdown a single VirtualHost?

No, the shutmsg file works at a daemon level not at a virtual host level.

19. Error 421

This appears to be a general catch all error code meaning "something nasty has gone wrong".

- Connection has timed out
- The DefaultRoot specified doesn't exist
- The parent server has been killed
- Check /etc/services
- Wrong permissions on the DefaultRoot

You get the idea...

20. proftpd doesn't show in the processlist

Two possible reasons, first that it's simply not running, try proftpd −n −d2 to run in debug mode and see what happens. The other is that it's running from inetd and there are no active sessions at the moment.

21. How do I restart/reload the server?

This depends on the mode you're running the server in.

inetd

Unless you're making a configuration change to inetd itself nothing needs doing. The server reloads the configuration everytime a new connection is made.

Standalone

Either stop and start the server completely (a little aggressive for most admins tastes) or send a SIGHUP to the master daemon process.

22. 503 No PORT command issued

A bug was introduced in 1.2.0rc2 which prevented the PORT command working properly and therefore breaking the data socket under certain conditions. The bug was documented as bug 240 and has been fixed in CVS. A rc3 release is due before the end of Jan 2001.

23. Fatal: unable to determine IP address of

Proftpd was unable to work out what IP is associated with the hostname in the VirtualHost block. Normally caused by a problem with the DNS resolution of the host, check the resolv.conf file and that your chosen nameservers are functional.

24. 451 append/restart not permitted, try again

AllowStoreRestart is disabled by default because it will allow any writable file to be corrupted by a malicious user. It is recommended that this option is only used with authenticated users and then only in certain directories.

25. 501 REST not compatible with server configuration

As mentioned in the description of the HiddenStor configuration directive, use of that directive is incompatible with the FTP command REST. Either disable use of REST with the AllowRetrieveRestart and AllowStoreRestart directives, or do not use HiddenStor.

26. The time being displayed is wrong

The default behaviour for ProFTPD is to display all times relative to GMT. To use local time set "TimesGMT off" in the server section of the config. There is a known issue with Redhat 7, with regard to time handling. http://www.redhat.com/support/errata/rh7−errata−bugfixes.html

27. Authentication is taking too long

Make sure that ReverseDNS is disabled, turn off ident lookups. Additionally check the size of your /etc/passwd (or shadow) file, if it is large then the only solution may be to move to another authentication scheme.

28. Corrupted files

There appear to be some problems with both the use of sendfile() in ProFTPD and with the implementation within certain operating systems.

29. Can I upgrade ProFTPD without terminating the current sessions?

Short answer, no. Longer answer is no, but you can minimise the effects. The cleanest approach on servers which have significant amounts of traffic appears to be to use ftpshut to block new connections and terminate existing ones after a pre−determined time period and then to upgrade and restart. This approach limits the number of downloads which are terminated part way through.

30. No such group "nogroup"

The default ProFTPD configuration file uses the user "nouser" and the group "nogroup", some systems / distributions do not have the group "nogroup" defined. The solution is to either add the group "nogroup" to /etc/groups or to change the "nogroup" entry in the proftpd.conf to a group which does exist.

31. Why do I see "unable to set groups: Invalid argument"?

The setting of the group privileges for a process uses the setgroups(2) system call. This call will fail with the above error message for one of two reasons: there is a negative GID value for one of the groups, or the maximum number of groups for a single user has been exceeded.

Ideally, all IDs, both UID and GID, will be positive. Unfortunately, it is common on many systems to use −1 or −2, especially for such users as 'nobody', or group 'nogroup'. Use of these values uses C's treatment of data types to make the actual numeric value very high; some functions, like setgroups(), do not like this, though. In general, always use positive ID numbers.

The other limitation is the number of supplemental groups for a user (eg non−primary groups, the ones configured in /etc/group). The maximum number of supplemental groups to which a user may belong is defined by the operating system constant NGROUPS_MAX. On some operating systems, such as Solaris, this limitation may be tunable.

Some other applications may not encounter this error if they use the initgroups(3) function, which reads the /etc/group file for a user's supplemental group memberships, and sets those groups. This function, however, silently ignores any supplemental groups for user greater than NGROUPS_MAX, unlike setgroups(2), which complains.

If this is the cause of your error message, any solution will most likely involve reducing the number of groups your users are members of, or tuning the NGROUPS_MAX value, if your operating system allows it.

32. Why do I see error messages like these when I logout?

```
PAM(exit): Permission denied
open_module: stat(/usr/lib/security/pam_unix.so.1) failed: No such file or directory
load_modules: can not open module /usr/lib/security/pam_unix.so.1
PAM(exit): Dlopen failure.
```

These messages appear when the DefaultRoot configuration directive is in effect. This directive causes a user to be confined using the chroot(2) system call. This call, however, affects other system utilities, such as PAM. In this case, PAM's configuration is causing the PAM library to attempt to open PAM modules using a path that is no longer valid, thus the errors. This happens on logout because the chroot has already happened by that point; on login, the PAM modules are successfully found and loaded before the chroot, so no errors. These are merely cosmetic reporting errors, and do not really affect the functionality or security of the server.

Chapter 5. Configuration problems

Problems encountered in trying to make the server behave exactly as required after compilation and installation are complete and the server is running.

1. How do I add another anonymous login or guest account?

You should look in the sample−configurations/ directory from your distribution tarball. Basically, you'll need to create another user on your system for the guest/anonymous ftp login. For security reasons, it's very important that you make sure the user account either has a password or has an "unmatchable" password. The root directory of the guest/anonymous account doesn't have to be the user's directory, but it makes sense to do so. After you have created the account, put something like the following in your /etc/proftpd.conf file (assuming the new user/group name is private/private):

```
<Anonymous ~private>
AnonRequirePassword off
User private
Group private
RequireValidShell off
<Directory *>
<Limit WRITE>
DenyAll
</Limit>
</Directory>
</Anonymous>
```

This will allow ftp clients to login to your site with the username "private" and their e−mail address as a password. You can change the AnonRequirePassword directive to "on" if you want clients to be forced to transmit the correct password for the "private" account. This sample configuration allows clients to change into, list and read all directories, but denies write access of any kind.

2. How do I ftp as root?

First off this is a *bad* idea ftping as root is insecure, there are better more secure ways of shifting files as root.

To enable root ftp ensure that the directive "RootLogin on" is included in your configuration.

3. How do I provide a secure upload facility?

The following snippet from a sample configuration file illustrates how to protect an "upload" directory in such a fashion (which is a very good idea if you don't want people using your site for "warez"):

```
<Anonymous /home/ftp>
  # All files uploaded are set to username.usergroup ownership
  User username
  Group usergroup
  UserAlias ftp username
  AuthAliasOnly on
  RequireValidShell off

  <Directory pub/incoming/>
     <Limit STOR CWD>
        AllowAll
     </Limit>
     <Limit READ RMD DELE MKD>
        DenyAll
     </Limit>
  </Directory>
</Anonymous>
```

This denies all write operations to the anonymous root directory and sub−directories, except "incoming/" where the permissions are reversed and the client can store but not read. If you used <Limit WRITE> instead of <Limit STOR> on <Directory incoming>, ftp clients would be allowed to perform all write operations to the sub−dir, including deleting, renaming and creating directories.

4. How can I stop my users from using their space as a warez repository

The above fragment will control anonymous users however if a local user with a full account with up and download capability is abusing their space then the technical measures which can be taken are limited. Applying a sane system quota is a good start, using the mod_quota and mod_ratio modules may control the rates of upload/download making it less useful as a warez repository. In the end it comes down to system monitoring and good site AUP's and enforcement.

5. Can I rotate files out of an upload directory after upload?

Yes. You'll need to write a script which either checks the contents of the directory regularly and moves once it's detected no size change in a file for xyz seconds. Or a script which monitors an upload log. There is no automatic method for doing this.

6. How can I hide a directory from anonymous clients.

Use the HideUser or HideGroup directive in combination with the proper user/group ownership on the directive. For example, if you have the follow directory in your anonymous ftp directory tree:

```
drwxrwxr-x 3 ftp staff 6144 Apr 21 16:40 private
```

You can use a directive such as "HideGroup staff" to hide the private directory from a directory listing. For example:

```
<Anonymous ~ftp>
...
<Directory Private>
HideGroup staff
</Directory>
...
</Anonymous>
```

7. File/Directory hiding isn't working for me!

You need to make sure that the group you are hiding isn't the anonymous ftp user's primary group, or HideGroup won't apply.

8. I want to prevent users from accessing a hidden directory

You can either change the permissions on the directory to prevent the anonymous FTP user from accessing it, or if you want to make it appear completely invisible (as though there is no such directory), use the IgnoreHidden directive inside a <Limit> block for one or more commands that you want to completely ignore the hidden directory entries (ignore = act as if the directory entry does not exist).

9. How do I setup a virtual FTP server?

You'll need to configure your host to be able to handle multiple IP addresses. This is often called "aliasing", and can generally be configured through an IP alias or dummy interface. You need to read your operating system documentation to figure out how to do this. Once your have the host configured to accept the additional IP address that you wish to offer a virtual FTP server on, use the <VirtualHost> configuration directive to create the virtual server:

```
<VirtualHost 10.0.0.1>
ServerName "My virtual FTP server"
</VirtualHost>
```

You can add additional directive blocks into the <VirtualHost> block in order to create anonymous/guest logins and the like which are only available on the virtual host.

10. I only want to allow anonymous access to a virtual server.

Use a <Limit LOGIN> block to deny access at the top−level of the virtual host, then use <Limit LOGIN> again in your <Anonymous> block to allow access to the anonymous login. This permits logins to a virtual anonymous server, but denies to everything else. Example:

```
<VirtualHost 10.0.0.1>
ServerName "My virtual FTP server"
```

```
<Limit LOGIN>
DenyAll
</Limit>
<Anonymous /usr/local/private>
User private
Group private
<Limit LOGIN>
AllowAll
</Limit>
...
</Anonymous>
</VirtualHost>
```

11. How does <Limit LOGIN> work, and where should I use it?

The <LOGIN> directive is used to control connection or login access to a particular context (the directive block which contains it). When a client initially connects to ProFTPD, the daemon searches the configuration tree for <Limit LOGIN> directives, and attached parameters (such as Allow, Deny, etc). If it determines that there is no possible way for the client to ever be allowed to login, such as a "Deny from" matching the client's source address, without an overriding "Allow from" at a lower level, the client is disconnected without being offered the opportunity to transmit a user and password.

However, if it is possible for the client to be allowed a login, ProFTPD continues as per normal, allowing the client to login only if the proper <Limit LOGIN> applies. Normally, <Limit> directive blocks are allowed in the server config, <VirtualHost>, <Anonymous> and <Directory> contexts. However, <Limit LOGIN> should not be used in a <Directory> context, as clients do not connect/login to a directory (and thus it is meaningless).

By way of example, the following configuration snippet illustrates a <Limit LOGIN> deny which will cause any incoming connections from the 10.1.1.x subnet to be immediately disconnected, without a welcome message:

```
...
<Limit LOGIN>
Order deny,allow
Deny from 10.1.1.
Allow from all
</Limit>
...
```

Next, an example of a configuration using <Limit LOGIN> that will not immediately disconnect an incoming client, but will return "Login invalid" for all login attempts except anonymous.

```
...
<Limit LOGIN>
DenyAll
</Limit>
<Anonymous ~ftp>
...
<Limit LOGIN>
AllowAll
</Limit>
...
```

12. How can I limit users to a particular directory tree?

For general open access you can use an <Anonymous> directive context block, possibly in combination with a UserPassword/AnonRequirePassword directive.

However if you wish to jail an entire group (or groups) of users, you can use the DefaultRoot directive. DefaultRoot lets you specify a root jailed directory (or "~" for the user's home directory), and an optional group−expression argument which can be used to control which groups of users the jail will be applied to. For example:

```
...
<VirtualHost myhost.mynet.foo>
DefaultRoot ~
...
</VirtualHost>
```

This creates a configuration where all users who log into myhost.mynet.foo are jailed into their home directories (cannot chdir into a higher level directory). Alternatively, you could:

```
...
<VirtualHost myhost.mynet.foo>
DefaultRoot /u2/public users,!staff
...
</VirtualHost>
```

In this example, all users who are members of group "users", but not members of group "staff" are jailed into /u2/public. If a user does not meet the group−expression requirements, they login as per normal (not jailed, default directory is their home). You can use multiple DefaultRoot directives to create multiple jails inside the same directive context. If two DefaultRoot directives apply to the same user, ProFTPD arbitrarily chooses one (based on how the configuration file was parsed).

Security Implications

The DefaultRoot directive is implemented using the chroot(2) system call. This moves the "/" (or root) directory to a specified point within the file system and jails the user into this sub−tree. However this is not the holy grail of security, a chroot jail can be broken, it is not a trivial matter but it's nowhere near impossible. DefaultRoot should be used as part of a general system of security not the only security measure.

A more detailed http://www.bpfh.net/simes/computing/chroot−break.html on this subject and on the breaking of chroot jails has been written by Simon Burr

Non−root server issues

The chroot() system call will not work under a non−root ftp server process, the call requires root privaliges. Without them it simply doesn't work, there doesn't appear to be any checking in the code of the uid/gid before calling chroot so using DefaultRoot in such a setup will cause the server to fail.

Symlinks

Symlinks will not work from within a chrooted area. The reason should be clear from a casual inspection of the nature of the chroot command. It is not possible to have a symbolic link to a directory which can"t be reached beacuse it's outside of the current chroot. Work arounds to allow access to other parts of the file system include exporting the part of the filesystem to be accessed from inside the chroot and mounting via NFS, using hard file links or (on Solaris) using lofs to mount the directory via the loopback.

```
mount -Flofs /home/data1 /ftp/data1
mount -Flofs /home/data2 /ftp/data2
```

As of the 2.4.x Linux kernel tree it is possible to mount filesystems multiple times and to mount subdirectories of filesystems elsewhere on the filesystem.

13. How do I create individual anonymous FTP sites for my users?

There are two methods of accomplishing this (possibly more). First, you can create a directory structure inside your anonymous FTP root directory, creating a single directory for each user and setting ownership/permissions as appropriate. Then, either create a symlink from each user's home directory into the FTP site, or instruct your users on how to access their directory.

The alternate method (and more versatile) of accomplishing per–user anonymous FTP is to use AnonymousGroup in combination with the DefaultRoot directory. You'll probably want to do this inside a <VirtualHost>, otherwise none of your users will be able to access your system without being stuck inside their per–user FTP site. Additionally, you'll want to use a deferred <Directory> block to carefully limit outside access to each user's site.

1. Create a new unix group on your system named `anonftp". Please each user who will have per–user anonymous FTP in this group.

2. Create an `anon–ftp" and `anon–ftp/incoming" directory in each user's home directory.

3. Modify your /etc/proftpd.conf file to look something like this (you'll probably want to customize this to your needs):

```
<VirtualHost my.per-user.virtual.host.address>

# the next line limits all logins to this virtual host, so that only
anonftp users can connect

<Limit LOGIN>
DenyGroup !anonftp
</Limit>

# limit access to each user's anon-ftp directory, we want read-only
except on incoming

<Directory ~/anon-ftp>
```

```
<Limit WRITE>
DenyAll
</Limit>

</Directory>

# permit stor access to each user's anon-ftp/incoming directory,
but deny everything else

<Directory ~/anon-ftp/incoming>

<Limit STOR>
AllowAll
</Limit>
<Limit READ WRITE>
DenyAll
</Limit>

</Directory>

# provide a default root for all logins to this virtual host.
DefaultRoot ~/anon-ftp
# Finally, force all logins to be anonymous for the anonftp group
AnonymousGroup anonftp

</VirtualHost>
```

14. I want to support normal login and Anonymous under a particular user

You can use the AuthAliasOnly directive to control how and where real usernames get authenticated (as opposed to aliased names, via the UserAlias directive). Note that it is still impossible to have two identical aliased names login to different anonymous sites; for that you would need <VirtualHost>.

Example:

```
...
<Anonymous ~jrluser>

 User jrluser
 Group jrluser
 UserAlias ftp jrluser
 UserAlias anonymous jrluser
 AuthAliasOnly on
 ...

</Anonymous>
```

Here, the <Anonymous> configuration for ~jrluser is set to allow alias authentication only. Thus, if a client attempts to authenticate as "jrluser", the anonymous config will be ignored and the client will be authenticated as if they were a normal user (typically resulting in `jrluser" logging in normally). However, if the client uses the aliased username `ftp" or `anonymous", the anonymous block is applied.

15. Why doesn't Anonymous ftp work (550 login incorrect)?

Things to check

Check the following first:

- Make sure the user/group you specified inside the <Anonymous> block actually exists. This must be a real user and group, as it is used to control whom the daemon runs as and authenticates as.
- If RequireValidShell is not specifically turned off, make sure that your "ftp user" (as specified by the User directive inside an <Anonymous> block), has a valid shell listed in /etc/shells. If you do not wish to give the user a valid shell, you can always use "RequireValidShell off" to disable this check.
- If UseFtpUsers is not specifically turned off, make sure that your "ftp user" is not listed in /etc/ftpusers.

If all else fails, you should check your syslog. When authentication fails for any reason, ProFTPD uses the syslog mechanism to log the reason for failure; using the AUTH (or AUTHPRIV) facility. If you need further assistance, you can send email, including related syslog entries and your configuration file, to the ProFTPD mailing list mentioned elsewhere in this FAQ.

16. Bandwidth control

The Bandwidth directive has been removed as of 1.2.0pre8, this directive acted on a per−virtual basis. It was generally held that it worked on the principle that a single connection to a given virtual could take the full bandwidth limit until other connections were made. However, the server uses either separate server (inetd) or forked (standalone) model there is no way for the various processes to communicate, therefore is no way they could share the bandwidth allocation.

The replacement actually does the same but does it in a more rigorous manner and more precisely. The directives RateReadBPS, RateReadFreeBytes, RateReadHardBPS work by limiting on a per−connection basis.

```
Bandwidth                       81920

is replaced with something like

RateReadBPS                     81920
RateReadFreeBytes               5120
RateReadHardBPS                 on
```

To achieve a total limit on a per virtual basis a mix of RateReadBPS and MaxClients is needed. ie RateReadBPS x MaxClients = Total Bandwidth allocation. There is no way (at the moment) to specify that virtual server xyz has a maximum total bandwidth of 200K/s that it can use between all connections.

Per−virtual, per−user and global limits are currently in the "to be coded" pile and are being penciled in for the 1.3.x development series. There is some work in providing for a shared communication system between servers before this can happen.

Rate controls aren't working

In pre9 and earlier Rate* does not work if sendfile is enabled, recompile with −−without−sendfile and all should be as expected.

17. CHMOD isn't working

As of rc1 the AllowChmod command was added to allow control over who is allowed to use the CHMOD command. The default value for this directive is off.

18. How can I limit the size of uploaded files?

There is no way within ProFTPD itself to control how large a file can be uploaded. The best solution to this problem at the moment is to use whatever disk quota tools are available within your OS.

19. Can I disable Anonymous logins?

Yes, just remove all the <Anonymous> sections from your configuration file and reload the daemon.

20. Limiting the connections per loginID

This is not possible at this time, the directive appeared in the docs for a short period but the code was never added. The closest possible at the moment is to limit the number of connections from a given host.

21. How do I configure proftpd to allow transfer resumption (for downloads and uploads)?

To allow downloads to be resumed, you need to use the AllowRetrieveRestart configuration directive.

To allow uploads to be resumed, you need to use both the AllowOverwrite and AllowStoreRestart directives. The reason that both need to be allowed is that a restarted/resumed upload is a form of overwriting the file.

Also note that using HiddenStor and AllowStoreRestart is incompatible, as mentioned in the documentation for the AllowStoreRestart and HiddenStor directives.

22. When should the Bind directive be used?

The Bind directive is used to specify additional interfaces (addresses) for a given server; it is *not* used to configure the main interface for the server. For <VirtualHost> servers, this is not a problem, as the main interface for the server is set in the <VirtualHost> line.

For the main "default" server, however, the controlling of the main interface is more problematic. There is currently a bug report opened for this issue:

http://bugs.proftpd.org/show_bug.cgi?id=1253

Chapter 6. Security

1. General

As with all software there have been a number of security issues during the life of the project. The most recent information can always be found on http://www.proftpd.org/security.html

Versions 1.2.0 and above should be considered to be production code and few if any new features will be added to this code branch to maintain stability.

What about using Stackguard?

Stackguard (http://immunix.org) is a gcc variant which can protect programs from stack−smashing attacks, programs compiled using Stackguard dies without executing the stack code. While this approach is a good first line of defense against future problems it''s not a complete cure−all. Some of the buffer overflows were found on static variables, which are not protected by stack protection mechanisms.

2. Surely running ProFTPD as non−root will help?

Running ProFTPD as a non−root user gives only a marginal security improvement on the normal case and adds some functional problems. Such as not being able to bind to ports 20 or 21, unless it's spawned from inetd.

ProFTPD takes a middle road in terms of security. It only uses root privileges where required and drops to the UID defined in the config file at all other times. Times when root is required include, binding to ports < 1024, setting resource limits, reading configuration information and some network code.

For Linux 2.2.x kernel systems there is the POSIX style mod_linuxprivs module which allows very fine grain control over privileges. This is highly recommended for security−conscious admins.

3. How can I control what commands the server accepts?

Use a sane Allow/DenyFilter, these directives use regular expressions to control all text sent over the control socket. (If anyone has some good examples please let me know.)

4. How can I prevent the server version from being displayed?

Setting SeverIdent to "off" should turn off the information about what type of server is running. To have maximum effect this directive should either be in the Global context or included in every virtual host block and the default block.

```
ServerIdent  On "Linux.co.uk server"
```

```
ServerIdent  Off
```

5. I want to show a message prior to login

Use the DisplayConnect directive to specify a file containing a message to be displayed prior to login.

```
DisplayConnect /ftp/ftp.virtualhost/login.msg
```

6. I want to display a message after login

Use the DisplayLogin directive, this sends a specified ASCII file to the connected user.

```
DisplayLogin       /etc/proftp.msg
```

7. Can I have a custom welcome response?

Use the AccessGrantMsg directive, this sends a simple single line message back to the user after a successful authentication. Magic cookies appear to be honoured in this directive.

```
AccessGrantMsg "Guest access granted for %u."
```

Note, this directive has an overriding default and needs to be specified in both VirtualHost and Anonymous blocks.

8. External Programs

ProFTPD has been designed to run as a secure ftp server, this means that it tries to keep as much as possible under it's control. An external program is a security risk in itself because it's behaviour is not controllable from within the ftpd code.

9. Why do I see "No certificates found!"?

This message is generated by mod_tls, the third−party module that can be used to encrypt both the control and data connections with TLS (Transport Layer Security), the next generation of SSL. Certificates are used to establish the security context for this secure transport.

Generation of certifications is beyond the scope of this document; however, more information can be found here:

http://www.linuxdoc.org/HOWTO/SSL−Certificates−HOWTO/

10. I can delete files owned by root. Why is this?

ProPTPD follows the UNIX file permission rules when determining the level of access and/or control a user is granted when working with a file. UNIX systems divide the world into three classes when determining the permissions that a user is granted for a particular file:

- User − the owner of the file

Group − a collection of users defined in /etc/group

- Others − neither the owner, nor a member of the group

Every file in a Unix filesystem has a permission definition associated with it. At a minimum, the permission established for a file will determine whether a particular user may READ, WRITE, or EXECUTE the file in question. A directory listing will show the permissions associated with a file in the format shown below:

```
rwx  r-x  r-x
 |    |    |
 |    |    |______________ Others:  READ/NO WRITE/EXECUTE
 |    |___________________ Group:   READ/NO WRITE/EXECUTE
 |________________________ User:    READ/WRITE/EXECUTE
```

In the sample directory listing shown below, READ/WRITE/EXECUTE privileges are granted to the owner of the directory, and READ/EXECUTE privileges are granted to members of the `users` group and everyone else. Note the letter "d" at the beginning of each entry, denoting that the entry is actually a directory.

```
prince> ls -l /home/ftp

total 8
drwxr-xr-x    2 andrea   users        4096 May  3 00:40 andrea
drwxr-xr-x    2 eve      users        4096 May  3 00:40 eve

prince> ls -l /home/ftp/andrea

total 156
-rw-r--r--    1 andrea   users       85991 May  3 01:12 bland.txt
-rwxr-xr-x    1 root     root        65107 May  3 01:12 secret.txt
```

The answer to this question is shown in the above example. When describing the permissions associated with a directory, WRITE means that permission is granted to modify the contents of a directory by adding or deleting files. Thus, the user `andrea` may delete the file `secret.txt`, even though she cannot modify the file itself.

Refer to the documentation for the *IgnoreHidden* and *HideNoAccess* directives for a method to mitigate this hazard.

Chapter 7. User Authentication

This section is being re−written due to major structural changes to the SQL module prior to 1.2.0

1. Why is PAM the default authentication system?

Security, pure and simple. PAM is the most secure (or securable) of the available authentication systems. Many of the issues and configuration hints for PAM are contained in README.PAM which is bundled with the server source and in the various packaged builds. To use /etc/passwd manual compilation will be required with the configure script being run with the −−without−pam flag. Unless the PAM subsystem is properly configured authentication will fail.

2. Authentication methods supported

- PAM
- Standard /etc/passwd lookups
- NIS
- Shadow passwords
- Indvidual passwd/group files for each virtual
- SQL databases

If these don't fit in with your system then writing a custom module or using such as the "ld.so.preload" approach to intercept getpwbynam() system calls works happily with ProFTPD.

3. Problems with non−PAM authentication

Generally these problems will be cured by either disabling PAM completely or by ensuring that these directives are set

```
PersistentPasswd   off
AuthPAMAuthoritative off
```

4. AuthPAMAuthorative is an unknown directive!

Check the spelling it should be AuthPAMAuthoritative not AuthPAMAuthorative or any other variation.

5. Configuring PAM

There is a README.Pam in the top directory of the ProFTPD install directory :

Redhat Linux

```
#%PAM-1.0
auth       required     /lib/security/pam_listfile.so item=user
sense=deny file=/etc/ftpusers onerr=succeed
auth       required     /lib/security/pam_pwdb.so shadow nullok
account    required     /lib/security/pam_pwdb.so
session    required     /lib/security/pam_pwdb.so
```

SuSE Linux

SuSE appears to uses pam_unix rather than pam_pwdb which is the Redhat approach. All references to pam_pwdb should be replaced with "pam_unix" on SuSE systems.

The following fragment is reported to work fine on SuSE 6.2

```
/etc/pam.d/ftpd
#%PAM-1.0

# Uncomment this to achieve what used to be ftpd -A.
# auth       required     /lib/security/pam_listfile.so item=user sense=allow file=/etc/ftpchroot onerr=fail

auth     required       /lib/security/pam_listfile.so item=user
sense=deny file=/etc/ftpusers onerr=succeed
auth     sufficient     /lib/security/pam_ftp.so
auth     required       /lib/security/pam_unix.so
auth     required       /lib/security/pam_shells.so
account  required       /lib/security/pam_unix.so
password required       /lib/security/pam_unix.so
session  required       /lib/security/pam_unix.so
```

FreeBSD

FreeBSD does not support PAM session directives. If you remove the following line from the FreeBSD section of README.PAM, PAM should work properly under recent versions of FreeBSD.

```
 ftp session required    pam_unix.so         try_first_pass
```

6. pam_sm_open_session errors

ProFTPD requires PAM version 0.59 or better. pam_sm_open_session is not part of previous versions.

7. Normal users can't login, only anon.

Check that the /etc/pam.d/ftp file exists on the system and is configured as detailed in README.PAM

8. AuthPAMAuthoritative

Currently AuthPAMAuthoritative defaults on "ON" resulting in login failures if PAM cannot authenticate the user. This breaks the AuthUserFile directive as it never gets a chance to authenticate the user unless the AuthPAMAuthoritative directive is set to "OFF"

The reasoning behind the current default is to ensure that the system is secure by default requiring that the admin explicitly and knowingly has to disable it. There are discussions underway which may result in the directive flipping to a default of "Off" if AuthUserFile is specified.

Note: as of the current CVS and the forthcoming pre9 release the default has changed to "Off"

9. LDAP

mod_ldap is currently stable; there were a couple bugs that were squashed after release 1.0 of the module. it is still udner development , check the http://horde.net/~jwm/software/proftpd−ldap/ for more information. There is an example config fragment on the author's site which gives a reasonable idea on how to use this module.

10. Encrypted passwords

There are patches which are being merged in at the moment to provide SHA encryption. The plan is to have the server get all user information except passwords via an anonymous bind. The server will then reconnect as a user is logging in and attempt to get the password via an encrypted connection. This should be in the next major release (2.5)

11. SecureID

No support yet

12. One time passwords

This is possible using either PAM or the Opie modules. The module passes back a challenge which the user puts into a key generator along with their "pass phrase" and it gives them back 5 words which get sent as the password. As long as you do it correctly it will never repeat.

It requires http://inner.net/opie/ to be installed on the server. There are key gen clients for win95/98, *nix, mac.

ftp://ftp.urbanrage.com/pub/c/mod_opie.c

13. RADIUS

Radius support isn't built into ProFTPD, though there's nothing stopping someone writing a module and submitting it for inclusion in the code tree. Possibly the easist way to implement Radius is by using the modules available for PAM and using the inbuilt PAM support.

14. Anonymous password checking

Is it possible to check an offered email address in an anonymous login before allowing access. Simple answer, not a hope in hell, anonymous access is pretty much designed to be freely open without checks and restrictions other than those placed on upload/download from the site. The best that can be hoped for is decent logging and tracking of accesses, and the requesting IP.

15. Why do I see "PAM(name): Authentication failure", but I can login anyway?

If the operating system supports PAM (Pluggable Authentication Modules) proftpd will perform PAM authentication by default. However, this authentication is not "authoritative" by default, meaning that a PAM authentication failure will not necessary cause a login to fail. The use of PAM can be configured using the AuthPAM configuration directive; the "authoritativeness" of any PAM checks is controlled via the AuthPAMAuthoritative configuration directive.

Chapter 8. FAQ Notes

1. History

The original text for this document was based on the configuration FAQ on www.proftpd.org. It was taken over in Sept 1999 when the maintainer of the software changed.

The faq is maintained by a group of people (usually lurking on irc or on proftpd−docs) using CVS and Docbook.

2. Acknowledgements and Thanks

Thanks to the developers, anyone who's posted useful information to the mailing lists and those who've mailed me direct.

This document couldn't have been maintained without the Sgml Tools package and the document layout defined by the Linux HOWTO maintainers.

Some specific mentions, in no particular order, and I've missed anyone please drop me a line.

Updates to the SQL section, Michael Grabenstein <mgrabens@popd.isinet.com>
Matt Mozur, who's been cleaning up some of my mess and generally stuffing patches in my direction.
TJ Saunders, for the HOWTOs and other docs.