Proftpd

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Proftpd

A User's Guide

Mark Lowes

Dedication

This book is dedicated to Lady Kayla.

Table of Contents

AccessGrantMsg −− Customise the response on successful authentication
Allow −− Access control directive
AllowAll −− Allow all clients
AllowChmod −− Enable the CHMOD command (deprecated)
AllowFilter −− Regular expression of command arguments to be accepted
AllowForeignAddress −− Control the use of the PORT command
AllowGroup −− Group based allow rules
Allow −− Permit logging to symlinked files
AllowOverwrite −− Enable files to be overwritten
AllowRetrieveRestart −− Allow clients to resume downloads
AllowStoreRestart −− Allow clients to resume uploads
AllowUser −− User based allow rules
AnonRatio −− Ratio directive
AnonRequirePassword −− Make anonymous users supply a valid password
Anonymous −− Define an anonymous server
AnonymousGroup −− Treat group members as anonymous users
AuthAliasOnly −− Allow only aliased login names
AuthGroupFile −− Specify alternate group file
AuthPAM −− Enable/Disable PAM authentication
AuthPAMAuthoritative −− Set whether PAM is the authoritive authentication scheme
AuthPAMConfig −− Select PAM service name
AuthUserFile −− Specify alternate passwd file
AuthUsingAlias −− Authenticate via Alias−name instead of mapped username
Bind −− Bind the server or Virtualhost to a specific IP address
ByteRatioErrMsg −− Ratio directive
CDPath −− Sets "search paths" for the cd command
Class −− Definition statements for class based tracking
Classes −− Enable Class based connection tracking
CommandBufferSize −− Limit the maximum command length
CwdRatioMsg −− Ratio directive
DefaultChdir −− Set starting directory for FTP sessions
DefaultQuota −− Sets the default quota
DefaultRoot −− Sets default chroot directory
DefaultServer −− Set the default server
DefaultTransferMode −− Set the default method of data transfer
DeferWelcome −− Don't show welcome message until user has authenticated
DeleteAbortedStores −− Enable automatic deletion of partially uploaded files
Deny −− Access control directive
DenyAll −− Deny all clients
DenyFilter −− Regular expression of command arguments to be blocked
DenyGroup −− Group based deny rules
DenyUser −− User based deny rules
Directory −− FIXME FIXME
DirFakeGroup −− Hide real file/directory group
DirFakeMode −− Hide real file/directory permissions
DirFakeUser −− Hide real file/directory owner
DisplayConnect −− Sets connect banner file
DisplayFirstChdir −− FIXME FIXME
DisplayGoAway −− FIXME FIXME
DisplayLogin −− FIXME FIXME
DisplayQuit −− FIXME FIXME

DisplayReadme −− FIXME FIXME
ExtendedLog −− FIXME FIXME
FileRatioErrMsg −− FIXME FIXME
FooBarDirective −− FIXME FIXME
Global −− FIXME FIXME
Group −− FIXME FIXME
GroupOwner −− FIXME FIXME
GroupPassword −− FIXME FIXME
GroupRatio −− Ratio directive
HiddenStor −− Enables more safe file uploads
HideGroup −− FIXME FIXME
HideNoAccess −− Block the listing of directory entries to which the user has no access permissions
HideUser −− FIXME FIXME
HostRatio −− Ratio directive
IdentLookups −− Toggle ident lookups
IgnoreHidden −− Treat 'hidden' files as if they don't exist
Include −− Load additional configuration directives from a file
LDAPAuthBinds −− FIXME FIXME
LDAPDefaultAuthScheme −− Set the authentication scheme/hash that is used when no leading {hashname} is present.
LDAPDefaultGID −− Set the default GID to be assigned to users when no uidNumber attribute is found.
LDAPDefaultUID −− Set the default GID to be assigned to users when no uidNumber attribute is found.
LDAPDNInfo −− Set DN information to be used for initial bind
LDAPDoAuth −− Enable LDAP authentication
LDAPDoGIDLookups −− Enable LDAP lookups for user group membership and GIDs in directory listings
LDAPDoUIDLookups −− Enable LDAP lookups for UIDs in directory listings
LDAPForceDefaultGID −− Force all LDAP−authenticated users to use the same GID.
LDAPForceDefaultUID −− Force all LDAP−authenticated users to use the same UID.
LDAPHomedirOnDemand −− Enable the creation of user home directories on demand
LDAPHomedirOnDemandPrefix −− Enable the creation of user home directories on demand
LDAPHomedirOnDemandPrefixNoUsername −− FIXFIXFIX
LDAPHomedirOnDemandSuffix −− Specify an additional directory to be created inside a user's home directory on demand.
LDAPNegativeCache −− Enable negative caching for LDAP lookups
LDAPQueryTimeout −− Set a timeout for LDAP queries
LDAPSearchScope −− Specify the search scope used in LDAP queries
LDAPServer −− Specify the LDAP server to use for lookups
LDAPUseTLS −− Enable TLS/SSL connections to the LDAP server.
LeechRatioMsg −− Sets the 'over ratio' error message
Limit −− FIXME FIXME
LogFormat −− Specify a logging format
LoginPasswordPrompt −− FIXME FIXME
LsDefaultOptions −− FIXME FIXME
MasqueradeAddress −− Configure the server address presented to clients
MaxClients −− Limits the number of users that can connect
MaxClientsPerHost −− Limits the connections per client machine

MaxHostsPerUser −− Limit the number of connections per userid
MaxInstances −− Sets the maximum number of child processes to be spawned
MaxLoginAttempts −− Sets how many password attempts are allowed before disconnection
MultilineRFC2228 −− FIXME FIXME
MySQLInfo −− Configures the MySQL driver
Order −− Configures the precedence of the Limit directives
PassivePorts −− Specify the ftp−data port range to be used
PathAllowFilter −− FIXME FIXME
PathDenyFilter −− FIXME FIXME
PersistentPasswd −− FIXME FIXME
PidFile −− FIXME FIXME
Port −− FIXME FIXME
PostgresInfo −− Postgres backend configuration (Deprecated)
PostgresPort −− Sets the port postgres is listening on
QuotaBlockName −− FIXME FIXME
QuotaBlockSize −− FIXME FIXME
QuotaCalc −− FIXME FIXME
QuotaExempt −− FIXME FIXME
Quotas −− FIXME FIXME
QuotaType −− FIXME FIXME
RateReadBPS −− FIXME FIXME
RateReadFreeBytes −− FIXME FIXME
RateReadHardBPS −− FIXME FIXME
RateWriteBPS −− FIXME FIXME
RateWriteFreeBytes −− FIXME FIXME
RateWriteHardBPS −− FIXME FIXME
RatioFile −− Ratio directive
Ratios −− FIXME FIXME
RatioTempFile −− Ratio directive
RequireValidShell −− Allow connections based on /etc/shells
RLimitCPU −− Configure the maximum CPU time in seconds used by a process
RLimitMemory −− Configure the maximum memory in bytes used by a process
RLimitOpenFiles −− Configure the maximum number of open files used by a process
RootLogin −− Permit root user logins
SaveRatios −− FIXME FIXME
ScoreboardPath −− Sets the path to the scoreboard file
ServerAdmin −− Set the address for the server admin
ServerIdent −− Set the message displayed on connect
ServerName −− Configure the name displayed to connecting users
ServerType −− Set the mode proftpd runs in
ShowDotFiles −− Toggle display of 'dotfiles'
ShowSymlinks −− Toggle the display of symlinks
SocketBindTight −− Controls how TCP/IP sockets are created
SQLAuthenticate −− Specify authentication methods and what to authenticate
SQLAuthoritative −− FIXFIXFIX
SQLAuthTypes −− FIXME FIXME
SQLConnectInfo −− FIXME FIXME
SQLDefaultGID −− FIXME FIXME
SQLDefaultHomedir −− FIXFIXFIX
SQLDefaultUID −− FIXME FIXME

List of Examples

Preface

Welcome to this text on the ProFTPD server software, this document grew out of a need for good documentation for the software. ProFTPD was written as an Open source software project released under the Gnu Public License (GPL). Many of the concepts have been inspired by or derived from the Apache webserver project.

This book grew out of a small FAQ on the proftpd.org website prior to the change in maintainer in Sept 1999. The need for a accurate and comprehensive FAQ as obvious, it rapidly became clear that a simple FAQ would not be suffcient. In Oct 1999 I started work on developing this document using the DocBook DTD in conjunction with the jade.

The software is currently designed for the Unix operating system and it's derivatives including Linux and the BSD variants. It is also reported to compile under win32, however it has not been designed for this environment.

This Book's Audience

This text is primarily targetted at system administrators who wish to make the most of the Proftpd software package. I expect that most readers will have at least a grasp of the ftp protocol and reasonable skills in compiling and maintaining a live Unix based system. For a list of resources which I consider to be useful reading to give this base knowledge consult Appendix A.

It is my hope, however, that the text is sufficiently generic in approach that it will be of use to those simply wishing to know more about ftp and the function of a typical ftp server.

The later chapters go into more depth on complex configurations and discuss the needs of a live server hosting multiple virtual hosts and hopefully suggest ways in which to keep the administration of these configurations to a managable scale.

Why Read This Book?

This book is designed to be the clear, concise, informative reference to the Proftpd FTP server software, I hope that this document will become the official documentation for this software.

I hope to answer, all the questions you might have about the issues concerning setting up and configuring Proftpd and running the server software in the open and sometimes hostile environment of the Internet. In particular I cover the following subjects:

- How FTP operates, is defined and how it fits into todays Internet.
- How to configure a basic anonymous ftp server and a basic user based ftp server.

Request for Comments

Please help me improve future editions of this book by reporting any errors, inaccuracies, bugs, misleading or confusing statements, and plain old typos that you find. An online errata list is maintained at http://deliberatly broken link/. Email your bug reports and comments to us at hamster@vom.tm.

Organization of This Book

This book is divided into xxxmultiplexxx parts. *Part I: Introduction* is an introduction to ftp, security and your first ftp server:

Chapter 1
: A quick introduction to FTP

Part II: Configuration is a guide to getting the server configured and running

Chapter 1
: A quick introduction to FTP

Acknowledgements

Many thanks to the Proftpd developers, anyone who's posted useful information to the mailing lists and everyone who has mailed me direct.

Copyrights and Trademarks

I. Introduction

Table of Contents

Chapter 1. Background

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.

Who codes/maintains Proftpd?

As with all Open source projects no one person can really lay claim to the entire package. The ProFTPD project was started by Floody who took it to approximately 1.2.0pre2/3 before he found that his available time was insufficient to handle this project as well as his other commitments. Since then (mid−late 1999) MacGyver has taken over the project and is pushing towards cleaning up the outstanding patches and getting 1.2.0 shipped.

There are also numerous people involved in developing modules, and documentation for the project. A number of these have been merged into the core distribution and more are likely to follow.

Website & documenation

The official website for the project is http://www.proftpd.org/, both should be a mirror of the other.

The documentation is being brought back into shape at the moment, the configuration on the website is reasonable but the documentation supplied in the source should be considered to be cannonical. Even this is still being brought up to date.

Bug reporting?

At the moment the best way to report a bug is to email the ProFTPD−Devel mailing list or MacGyver directly.

Availability

Primary FTP: ftp.proftpd.org (primary site)

Available formats

Tarball

This is the canonical source for ProFTPD. It is provided in both tar−gzip and tar−bzip2 formats.

RPM

ProFTPD is packaged up for Redhat, the default build uses PAM for authentication

DEB

Package: ProFTPD Priority: optional Maintainer: Johnie Ingram <johnie@debian.org>

This package is being actively maintained and seems to follow the available releases within 24−48 hours or so.

Mailing lists

There are three 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.

Users

proftpd@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−request@proftpd.org with "subscribe" in the subject.

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.

Copyright Issues

The Proftpd 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.

The FTP protocol

FTP was defined initially in RFC959 and has been updated in RFC2228. The protocol pre−dates RFC959 by over a decade during which time various RFC's were written to move the protocol towards a clear stable standard. This standard has now served the Internet well for fourteen years and shows only minor signs of it's age. RFC2228 currently only has standards track status but shows all the signs of becoming a full IETF standard for the internet. This new RFC extends the protocol to include encrypted and authenticated connections and to provide methods of assurance of data integrity. Proftpd is RFC959 compliant and there are plans to make it RFC2228 compliant in version 1.4 and later.

The File Transfer Protocol (FTP) does exactly what it says, it allows the movement of files from one place to another. Like most of the services on the internet it's designed round the client−server model. Given this software related to ftp can be split along these lines, Proftpd is a ftp server.

FTP servers allow access by authenticating users against a password database of some description. Historically this has been the unix /etc/passwd file (and later /etc/shadow) more recently support for other authentication systems as been provided including NIS, Radius, SQL, LDAP and many others. For most servers the username and password are sent over the network in plaintext. There is a RFC defining the specification for encrypted passwords for use with ftp servers but this not had a widespread takeup.

Anonymous Servers

In addition to properly authenticated users there ftp has historically allowed a special class of user. The "anonymous" connection, primarily used for public archives of data, programs or general "stuff" anonymous logins allow anyone on the network to connect to a server. Normally anonymous connections are limited in number to prevent the free aspect to the server from overwhelming it's primary function and the access permissions and rights of the anonymous user are locked down.

Anonymous servers are one of the great resources of the Internet over the years they collectively have become a massive redundant public storage system for information and programs. This is partly due to the open nature of many admins in what they will allow to be hosted and partly in the habit of "mirroring" other sites to spread the load. Without anonymous servers it's unlikely that the Open Source community would have been able to achieve the critical mass and accessibility required for it's current success.

Sockets and ports

FTP was designed round a two socket model, streaming data down one socket and control information down the other. This design makes it possible for a well designed client to be uploading and downloading while still permitting the user to perform other adminitrative tasks on the server.

Normally the control socket uses port 21 (ftp) at the server end, the data socket handling is more complex. Two modes of operation are defined for ftp connections.

Active

Active mode connections run control over port 21 and allow the server to decide which socket to use locally for data traffic.

Passive

Passive Mode connections work the same way as normal (Active Mode) connections, except the data connection is also made from the client to the server. This avoids the problem of incoming data connections being blocked by the firewall by making both connections from the client.

Problems

Unfortunately, not all FTP clients are capable of passive mode transfers, and not all users are aware of their existence or the problems they solve. Some firewalls can be configured to allow incoming FTP data connections while blocking all other incoming TCP connections. (The firewall recognizes FTP data connections because they originate from port 20, the FTP data port). This allows Active Mode FTP transfers through the firewall without blocking the incoming FTP data connections. Support for port connections established on the traditional FTP data port (20) was added in Rumpus 1.2, so older versions of Rumpus will not work correctly with firewalls configured this way.

Passive Mode connections work the same way as normal (Active Mode) connections, except the data connection is also made from the client to the server ie made to port ftp−data (20). This avoids the problem of incoming data connections being blocked by the firewall by making both connections from the client. What it boils down to is Active control channel, port 21 data channel, server specifies random port. Passive control channel, port 21 data channel, port 20 I guess it's doc time :)

Chapter 2. Compilation and installing

Being Open Source code Proftpd is availble primarily as source code for local compilation. There are a number of maintainers within the project who create the packaged builds for the primary platforms and distributions. For most users the packaged builds will prove to be suffcient and the least hassle route to installing Proftpd. However to use the daemon to it's fullest or to explore some of the more interesting features a local custom build will be required.

Architecture

Proftpd was designed from the ground up to be both extensible and as secure by design as possible. Security is discussed in depth later in this document, however while there are no known security holes it cannot be said of this or any other piece of software that there are no problems waiting to be found. The extensibility is provided by means of a modular architecture which takes many lessons and features from the Apache webserver project. Almost all the functionality has been supplied by moving most functions into modules. This includes features such as "ls" and the authentication handling, this approach allows third party developers to provide additional modules to latch onto these hooks to add or extend the basics provided. Most of the more interesting modules have to be compiled in as they are not part of the standard builds. Unfortunately dynamically loadable modules are not available within the 1.2.x code tree though development and testing is planned for the 1.3.x development branch.

Installing packaged versions

Linux (RPM)

The Linux RPM package is available on the main distribution sites for Proftpd. Installtion is as simple as `rpm --install proftpd-{version}.rpm`

Note: check multiple rpm route now supported.

Linux (DEB)

The Debian package is equally as simple to install as the RPM with either

```
dpkg --install {debfile}
```

or

```
apt-get install proftpd
```

FreeBSD

Does anyone have any comments on the BSD install?

Compiling from source

Supported Platforms

Proftpd is reported to compile and function on the following platforms,

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

Some platforms require that compilation is done with gcc (or one of it's variants) while other platforms only function properly if compiled with the compiler shipped with the OS. Experiences vary and at this time there is no reliable list of which platform requires what.

FreeBSD

ProFTPD is part of the FreeBSD ports collection. The minimal install commands on a system with a properly installed ports tree are:

cd /usr/ports/ftp/proftpd
make install

More information can be found in the README.html file in the same directory.

Including additional modules

Only the essential core modules are compiled in by default, most of the more interesting features such as sql support, upload/download ratios etc etc are contained within the non−standard modules.

Including additional modules is only possible at compile time, at the moment there is no chance of dynamically loadable modules entering the 1.2.x code tree. This is primarily due to time and the need for some major structural changes within the code to support dynamically loadable modules.

Example 2−1. Configuring for additional modules

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

Compatibility Issues

sendfile bsd linux 2.2 vs 2.0 mod_linuxprivs libc5

linux

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.

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.

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.

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 −u" 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.

Using non−default modules

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

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.

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

Longer term development

For 1.9.x/2.0.x there are plans to completely recode some core sections of the software and creating an abstract layer to build the 2.x version on. The abstract layer will handle all filesystem and OS−specific stuff. This layer will then have backends onto the major environments (ie Unix and NT)

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.

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 CDB based Authentication

Chapter 3. Security Issues

As with all services there is the risk that abuse can happen or that a crack attempt will be made on the hosting server. As a general rule crackers will attempt to break in through known holes in the various server daemons running.

The cautious and security concious system admin should be aware of the two main avenues for abuse, external and internal. I will consider external attacks to be those made by individuals without valid accounts or "user" level access to the server. Internal I will consider as being those individuals with authenticated user access of some form to the server.

Server Security security holes weak passwords Abuse of server warez dumping ground

Securing ftp servers

In general there is not much more to securing a ftp server than there is to any other public access server. However the twin socket design and thus the requirement to never quite give up root privileges completely leaves a window ajar for the competant cracker to climb through. Or occasionally a thumping great sign and open door for a script kiddie with some time to spare.

Proftpd provides for some additonal security by it's use of chroot(), user and IP access limits, command and path filters to limit what and where files can be uploaded and it's attention to when root privs are needed and when they are not. However a buffer overflow in the wrong place and it's possible that the server is compromised beyond hope.

Simple steps which can be taken to tighten security include

- Log to a separate machine
- Traffic filtering upstream of the server
- chroot() all sessions
- Don't give a valid shell where it's not needed
- Run an intrustion detection system
- If possible place the OS itself on a bootable CDrom
- Tripwire
-

Decent backups

Daemon security

Recently (between versions 1.2.0pre3 – 1.2.0pre7) there have been a number of buffer overflow type security problems with ProFTPD, with the coming release of pre7 these should be under control. Though no absolute statement can be given on the security of the software (this is true for every piece of software out there). A significant amount of effort has been put into removing the more 'dangerous' system calls which are prone to overflow attacks.

There is a known security problem with ALL unix FTP daemons, which requires the daemon to retain root privileges even after a client has fully authenticated. In ProFTPD versions 1.0.x, a decision was made to ignore RFC959's port 20 requirements in the interests of security. This approach has now been abandoned in favour of a more rfc compliant approach.

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.

Password Issues

One of the biggest security problems about the whole FTP protocol is the need to have the password transmitted in clear text across the network. In effect the username and passowrd pair are available at all times during the authentication sequence, resulting in this information being available to crackers and sniffers alike.

Encrypted passwords

Currently (as of 1.2.0pre9) Proftpd does not support encrypted passwords for authentication. Development for this feature is scheduled for post 1.2.0rel1, and it will remove the absolute need to send the password in clear text over the network. There are some additional approaches involving ssh (secure shell) which I will not cover in detail in this text which can be used to secure a ftp session without encrypted keys.

FTP as root

This is a bad idea simply because it's a major security risk to send the root passowrd in clear text over any network. If there is a need to get files onto a server there are always better ways of achieving it than connecting via ftp as root.

Example 3–1. Other approaches

- rcp
- ssh/scp
-

ftp as a safe user and change the ownership later.

If you really must ftp as root then our thoughts go with you on this dangerous journey as you add "RootLogin on" to your proftpd configuration and may your god go with you.

Server attacks

As with all server processes the primary method of cracking remains the buffer overflow. Due to the nature of the protocol and the requirement for root level privileges this leaves ftp daemons open to attack. Buffer overflows are the result of weak programming where boundary condition checks have been skipped or "unsafe" system calls have been used. These allow a fixed length storage area to be overflowed, this overflow can then be used as the transport to allow the execution of arbitary commands as the root user. In combination this is known as a "root exploit".

Stack smashing protection

What about using Stackguard?

Stackguard 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.

Libsafe

Libsafe implements a 'middleware' layer which sits between the OS and the daemon process and protects against buffer overflows. This is achieved by intercepting all calls known to be vunerable to overflow. http://www.bell−labs.com/org/11356/html/security.html

Running Proftpd as non−root

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. The inability to bind to ports 20 and 21 makes this approach usless for commercial hosting environments where the customers are expecting the connection to be on a "standard" port.

Linux

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.

Firewall issues

Generally ftp and firewalls are quite capable of co−existing on the same or separate networks with the minimum of fuss. The source of problems stem from the fundamental design of ftp and it's twin socket approach to data transfer. Firewalls, good ones at least, approach security by assuming everything is hostile and then starting to open up holes to trusted ports and destinations.

FTP, as has been mentioned in an earlier chapter has two main methods of operation, passive and active. Passive mode support is difficult in the extreme to support within a firewall, it requires the tracking of port 21 connections in and outbound and opening up complete tcp holes for that connection on the fly and tearing down once the control socket is closed. Active support is brainlessly simple by comparision, opening ports 20 and 21 is sufficient, nothing more complex is required.

ProFTPD behind a firewall

Due to the multiple socket and semi−random port assignment nature of the ftp protocol Because of the bi−socket nature of the ftp protocol additional care must be taken when setting up ProFTPD behind a firewall. Setting the firewall to allow the control socket through is easy enough, allow tcp packets destined for port 21 on the target server. However the data socket in passive mode may be targetted on a random port number on the server side resulting in either a highly complex or very weak firewall. The PassivePorts directive allows the admin to specify the range of ports the server will use to service ftp−data connections, this range can then be configured on the firewall.

Security by obscurity and warnings

Good security practice works on a combination of locking down all the holes as tightly as possible and letting as little information about the network out as possible. Additionally some legal systems require that explicit warnings are put up letting the casual connecting host know that unauthorised access is not permitted. To provide these features Proftpd supplies a number of directives which control the message presented to the user.

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
```

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

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

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.

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.)

Secure Sockets Layer (SSL)

There is currently no support and no plans by the primary developer to add support for SSL or any other security layer to the 1.2.x code tree. There are plans to implement security layer hooks and functionlity into the next development branch (1.3.x/1.4.x).

The planned solution for 1.3.x will include a generic security layer onto which other methods can be placed. This should provide a suitably generic position to start from allowing multiple solutions to be developed.

Chapter 4. Day to day issues

Starting and stopping your server

inetd, standalone, hosts.allow, HUP, PID, /etc/shutmsg

Timezone issues

http://proftpd.org/docs/configuration.html#TimesGMT says the default for this config option is 'on' in versions 1.2.0pre9 and beyond, and that the command exists in those same versions. That said, my install (from src) of 1.2.0pre10 neither supports the directive nor uses GMT.

(And, of course, exporting TZ=GMT before running it doesn't help, since it overwrites its environment after starting. I presume this is why the directive was added.) Jim, the ChangeLog file in the current CVS source tree contains these entries dated after the release of 1.2.0pre10 (17 Jan 2000):

My copy of the pre10 doc/Configuration.html doesn't contain the TimesGMT directive, nor is there any code for it. So, it looks like it was added after pre10, and the documentation is flat out wrong about the time of its introduction.

Well, with the environment overwritten time will be reported GMT, so I don't think that was the motivation. FYI, the environment overwrite bug should be fixed (finally!) in the current CVS sources (but it had nothing to do with the TimesGMT directive). However, you also may need to apply the suggested fix for Bug#76, if you wish to compile the current CVS sources on most non−Linux systems: http://bugs.proftpd.org/show_bug.cgi?id=76 http://bugs.proftpd.org/showattachment.cgi?attach_id=27 So, it looks like it was added after pre10, and the documentation is flat out wrong about the time of its introduction.

It will be reported in whatever zoneinfo file /etc/localtime is (or is a symlink to). At least it is on my box.

Log management

rotation, location, opening, log analysis

Rotating the log

Any of the common tools for managing log rotatation can be used with Proftpd. The most commonly used package is logrotate as shipped with Redhat. Some suggested configurations are shown below.

Example 4−1. logrotate configuration

```
# cat /etc/logrotate.d/proftpd
/var/log/proftpd {
        nocompress
        missingok
}
```

Example 4−2. logrotate configuration

```
/var/log/xferlog {
    # ftpd doesn't handle SIGHUP properly
    nocompress
}
/var/log/proftpd {
    nocompress
}
```

Example 4−3. logrotate configuration

```
/var/log/xferlog {
     postrotate
         /usr/bin/killall −HUP proftpd
     endscript
}
```

Proftpd does not use SIGHUP to close and reopen the logfiles so one of two basic stratagies have to be employed to ensure that the logfiles are not being held open. The first and most aggressive is to shutdown proftpd, rotate the logs and restart. This might be acceptable on a small server but not on a commercial system

A second approach would be to rotate the logfiles and not perform any parsing or compression until all the live connections have ended. This time can either be based on guesswork (ie I'm pretty sure everyone will have finished the active connection within 60 minutes) or by employing a script to kill off any remaining connections after a suitable time period (by using such as the fuser command).

Analysis of logfiles

So, you want to know what's happening with your ftp server, are those logs any help. Not normally is the most common response, as a general rule logfiles are unreadable and while providing the raw information for spotting trends are not the best format for presenting the information.

There are a number of different packages and approaches available to the sysadmin on the go to process his logs into a more readily understandable format.

Webalizer
: Webalizer is primarily designed as a web server log analysis tool. However it is capable of handling ftp server logs (set the logtype configuration option to 'ftp'). The latest version uses the png graphic format.

  http://www.mrunix.net/webalizer/

http−analyze
: http−analyze is the system from which webalizer was derived. It requires more work in setting up proftpd's logging format however it can give far more detailed reports.

HTTP−analyze
: http://www.netstore.de/Supply/http−analyze/

analog, *http://www.analog.cx/*
: If you want to use Analog (works fine for me) this is your logformat:

```
LOGFORMAT (%j %M %d %h:%n:%j %Y %t %S %b %r %j %j %j %j %u %j %j %j)
```

Report Magic, *http://www.wadsack-allen.com/digitalgroup/reportmagic/*
: Produces more 'professional' looking reports based on analog data.

logwatch,

Others
: Logsurfer (need URL) and a Perl custom reporting module (http://www.cpan.org/modules/by−authors/id/S/SN/SNEEX/)

Custom Logging

Thank you so much! This has GREATLY reduced the load on my server! Now I just have my ftp log, and the secure log with proftpd entries. Thanks again! LogFormat xfer_fmt "%t %u %f" ExtendedLog /var/log/upload write xfer_fmt ExtendedLog /var/log/dnload read xfer_fmt You can use this directive to disable the syslogd usage : SystemLog /usr/local/proftpd/logs/system_log a) will proftpd support piped logs? b) anyone intersted in make a mod_cronolog? http://www.ford−mason.co.uk/resources/cronolog/ im running Proftpd 1.2.0pre3 and i'm having trouble finding a log analyser that will support the type of logs i run through it. the main problem being i have extended characters and white spaces in file names. all log analysers i've tried interperet the whitespace as the end of the file name. is there any way to have proftpd use %20 instead of a space in the log file? or better yet, have proftpd keep a log CLF style?

FXP

FlashFXP is a Windows program which allows site to site transfers via the port bouncing technique described in rfc2577 (FTP Security Considerations [informational]). As a general rule allowing port bouncing is a bad idea and a major security hole.

Configuring Proftpd to allow port bouncing is simple, add "AllowForeignAddress on" in either the Global or Anonymous sections as appropriate and reloading the configuration will suffice. Without these directives the server will report "425 Passive PASV port theft" to syslog.

Example 4−4. Configuration fragment

```
ServerName                      "Frostbite FTPserver"
ServerType                      standalone
.
.
.
<Global>
.
.
.
        ExtendedLog    /var/spool/syslog/proftpd/fascist.log ALL default
        ServerIdent                     on "Linux.co.uk server"
        AllowForeignAddress     on
        PathDenyFilter                  "(\.htaccess)|(\.ftpaccess)$"
</Global>
.
.
```

```
.
<VirtualHost 195.200.4.15>
ServerAdmin             zathras@linux.co.uk
ServerName              "Linux.co.uk FTP Archive"
.
.
.
<Anonymous /ftp/ftp.linux.co.uk>
        User                    ftp
        Group                   ftp
        UserAlias               anonymous ftp
        RequireValidShell       no
        AllowForeignAddress     on
.
.
.
```

II. Configuration

Table of Contents

Chapter 5. Getting ready

What do you want from your server?

Deciding what you want to get from your server is often the most important part and usually the most often ignored part of the whole process of configuring any server software for use. Working out the details of the server, the loading expected and the levels of access to be given are critical to ensuring that you provide the service levels required.

Config file

Do you know where the daemon is expecting to find the config file? If not check now, the two most likely places are /usr/local/etc/proftpd.conf and /etc/proftpd.conf. The compile time default as shipped with the bare source is the former however the latter is the the default for many of the packaged versions of ProFTPD.

Scoreboard file

Standalone or inetd?

On Thu, Nov 30, 2000 at 08:01:42PM −0000, Tanuj Shah, − mailings wrote: > Is ProFTPd (1.2.0) better to run as standalone or via (x)inetd? Both runs fine. Only on one Solaris box I was forced to run in standalone mode cause it said all the time that there is another programm listening on Port 21 whenn I tried inetd. > What are the differences etc. etc. etc. ? One difference is that the process controll (childs etc) is mananaged bei the inetd. Another thing is that you can start proftpd with the tcpd when you're using the inetd. In the standanlone mode you can use Virtual Hosts. Personal preference, inetd for lightly used systems where resources are an issue. Standalone for production machines which are likely to get pounded into the dirt and I need the additional configuration features not available under inetd. Well, after reading here about Redhat 7 having xinetd, and needing to = put the server in standalone I noticed something fairly big.... I used to be able to edit the proftpd.conf file and the changes would = take place immediately, now I have to kill the process and restart the = server....anyone have any solutions? If I'm not mistaken, that's normal. A big advantage of inetd (or xinetd) is that it listens on ports for you. Only when it gets a connection on a port does it launch the respective program. So basically proftpd gets re−launched with every connection, thus you can edit the config and it will be in effect for the next user. Standalone mode though is always running with the config it saw when it first started up, so you do have to kill it and restart it to see the new config. Well, after reading here about Redhat 7 having xinetd, and needing to put the server in standalone I noticed something fairly big.... I used to be able to edit the proftpd.conf file and the changes would take place immediately, now I have to kill the process and restart the server....anyone have any solutions? If you send the main proftpd process the HUP signal, it will re−read it's configuration file without stopping... I'm a Linux (RH6.0) newbie and I'm trying to get ProFtpD running on my box... I'm having some little problems tough :(My first question... should I run it in standalone or inetd mode? My ftp won't have much traffic... the box is a 486 dx 33 w/ 8 megs of ram... nothing fancy... Second question... I tried to run it from commandline in inetd mode... it said that in order to run it from commandline it needs to be in standalone mode... and for inetd mode, proftpd has to be started by the inetd super−server. What is this super−server and how do I get this thing to start proftpd? Right now when I do ftp localhost, i get a 'connection refused' error message... maybe proftpd isn't even running (that's my guess)... how do I make sure it is running? On Sun, 13 Aug 2000, Carl Mercier wrote: > > My first question... should I run it in standalone or inetd mode? My ftp > won't have much traffic... the box is a 486 dx 33 w/ 8 megs of ram... > nothing fancy... if you won't be taking on that much traffic, inetd is the preferred method. If it's going to be a busy or "production" FTP server, standalone is best. Frankly, it doesn't matter that much in your case. > > Second question... I tried to run it from commandline in inetd mode... it > said that in order to run it from commandline it needs to be in standalone > mode... and for inetd mode, proftpd has to be started by the inetd > super−server. What is this super−server and how do I get this thing to > start proftpd? type "man inetd". Reading the manual page will tell you everything you need to know. > > Right now when I do ftp localhost, i get a 'connection refused' error > message... maybe proftpd isn't even running (that's my guess)... how do I > make sure it is running? in standalone mode, you will see "proftpd" in the output of 'ps −ef'. In inetd mode, it will be running provided you have inetd up and configured to accept connections for proftpd. One thing to add... if you run proftpd in standalone mode and not through the "inetd" server, then you must edit your /etc/inetd.conf file and comment out the reference to ftp (the only line starting with ftp). If you are going to run it through inetd, instead of commenting out that one line, change it to run proftpd... Again see "man inetd." My 2 cents would be on your system to run it in inetd. That way you don't have a ftpd server taking up memory all the time. With inetd, the server will only take up memory when you want to use it. Not to mention processor time, even idle processes have to be polled by the kernel. Later, Hello, I have a limited use server 10+ logins a week, 20mb a week transfers (usually upload). I have the server setup as inetd (changing to xinetd). Can anyone give a guideline table of when you want to use standalone vs inetd server model? Well, off the top of my head: INETD PROS Can use TCP wrappers Not using system resources when not in use Does not have to run as root (better security) CONS Can't use MaxClients Overhead from launching process for each session (although in my experience this is negligible) DAEMON PROS Better performance, since the daemon is always ready to take calls Can use MaxClients to allot resources or avoid a DOS attack CONS Daemon must run as root to bind to port 21, although I believe ProFTPD has some internal mechanisms to reduce risks Is always using system resources even when idle There are certainly other reasons that I am sure other users can add. I have always felt that the primary reason for choosing one over the other is volume. Low volume tends to indicates inetd, while high volume almost always indicates daemon. But its a balance of security and performance either way. A few less important pro/cons: INETD no User lockouts after too many false logins no reset needed after changing configuration DAEMON may suffer from memory leaks (system libs, modules,..) Things that run on ports <1024 (as does everything in inetd.conf) have to be run as root initially, which opens the possibility of exploitation. I think (keyword=think) running as standalone uses more memory than inetd. Speed isn't an issue for me since I have logins capped at 3 simultaneous. As for security I have a firewall router between the ftp box and well...all I can do is all I can do. Stand alone is faster in theory. I don't run anything from inetd. My ftp, www, ssh all run standalone. Given proftpd 1.2.0pre10, what are the relative merits of running it via inetd as opposed to standalone? I imagine that there's greater security with inetd given its use of host.deny. True? Are there any other security issues related to these 2 mode of running proftpd? If you insist not running it in standalone mode, something like tcpserver would be much better. Inetd does nasty things to busy systems because of the rate limiting it has. I run proftpd in standalone, and used to run it from inetd ;) Are there any performance differences between the two implementations or is the gap down entirely to the inetd overhead? if so another superserver (tcpserver?) could be used instead and one could have the superior rules access with a minimal overhead and performance degredation.

Contexts

At present, ProFTPD has seven different configuration contexts: "main" server, <Anonymous>, <Directory>, <Global>, <Limit>, <VirtualHost>, and .ftpaccess files. These contexts are checked for in configuration handlers using the CHECK_CONF macro.

Valid Configuration contexts

Main server

The "main" server context, listed as "server config" in the configuration directive documentation, encompasses everything outside of the other contexts (i.e. every configuration directive that is not explicitly contained within another configuration context), and signalled by the macro CONF_ROOT in a configuration directive handler.

<Anonymous>
The <Anonymous> section is used to set up the very common configuration of an anonymous FTP server. It does a chroot() to the anonymous FTP directory by default, and turns off the requirement for a valid password, requesting only a valid email address as the password. Other system binaries or files need not be contained within the <Anonymous> directory.

Note that since an <Anonymous> section is not considered a separate server, but rather a "subset" of its containing server, any configuration directives set for that server will be in effect for the <Anonymous> as well, unless overridden by a directive of the same name in the <Anonymous> context itself.

<Directory>
The <Directory> context is for configurations specific to directories, of course. This includes views of the contained files based on the logged−in user's username or group membership or on the name of the files (e.g. Unix−style "hidden" files), and on whether the user has permission to see the files. .ftpaccess files occur within this context by definition; <Limit> sections often appear in a <Directory> section as well.

<Global>
The description in the documentation for the <Global> context is good. Another point to know is that if a directive is set in this context, and then the same directive is used in the main or <VirtualHost> contexts with different parameters, those parameters take precedence over the <Global> parameters. This allows you to configure things for everyone equally, then tweak specific ervers individually, on a per−server basis.

<Limit>
The <Limit> context is used to place limits on who and how individual FTP commands, or groups of FTP commands, may be used.

.ftpaccess
These files are akin to Apache's .htaccess files, which are parsed−on−the−fly configuration files −− with restricted scopes −− that users can place in their own directories. Note that .ftpaccess are similar to Apache's .htaccess files, they are not the same. For example, ProFTPD's .ftpaccess files do not support a "require" directive, nor Apache's AuthRealm directive. That particular area of Apache configuration is targetted for restricting access to anonymous connections; by its nature, ProFTPD handles anonymous connections as special cases of the normal authenticated connections.

Chapter 6. Generic issues

File permissions and UMASK

What is a UMASK?

The umask is the method of automatically defining the default set of permissions a file will have when created or uploaded. you know how unix files have permissions, something like the following: −rwxr−x−−− which you will see when doing an "ls −l" in a directory. here's what they mean: −rwxrwxrwx 1234567890 1) For normal files, the 1st character is "−". For directories, it's "d". For symbolic links, it's "l". for other files (devices and FIFOs, see the ls man page) 2,3,4) Read, write and execute permissions for the file's owner. 5,6,7) Read, write and execute permissions for the file's group. 8,9,0) Read, write and execute permission for everyone else. So, lets say that I have a file with these permissions: −r−x−−−−−− If I want to change it to −rwxr−xr−x I have to do something like this chmod a+rx file chmod u+w file Do you know how to count in octal? If not, use this cheat, I mean, shortcut: r = 4 w = 2 x = 1 so, read + execute = 4 + 1 = 5, and read + write + execute = 7. so If I did a chmod 755 file I would get −rwxr−xr−x When one does a "chmod xyz file": The x is the file's owner permissions, the y is the file's group permissions, and the z is the files others permissions. This way, one can do a single chmod, and effect ALL the file's permissions at once. As for a umask, this is the REVERSE of the permissions: −rwxr−x−−− is 750 The REVERSE of the umask is is 027 (−−−−−w−rwx). I guess you could think of a umask as the permissions to TAKE AWAY from a file. by setting one's umasks to 027 would make it so any file you create, will be created with the permissions 027 REMOVED from the file. Like: −rwxrwxrwx minus −−−−−w−rwx ___________________ equals −rwxr−x−−− }}What is a umask that will create 775? 002 Want to know how you can tell? 777; where each 7 is equal to rwx, therefore 002 is 775. So, then 775 = −rwxrwxr−x Where the r = 4 the w = 2 and the x = 1 −−−−−− 7 HTH, −Sneex− :] (Note that the leading 0 is assumed −− yes it's octal :)

I've got a quick question.. We've picked proftpd as our best bet at our site and we're trying to configure it. Everything looks great except for one problem. Users can create directories just fine, but they can't change to them once they're created − when the directories are created, they lack the execute bit. We're shooting for permissions of 640 (rw−r−−−−−) for files and 750 (rwxr−x−−−) for directories. I've been using the Umask configuration directive as shown below: Umask 0137 0027

---

proftpd.umask

I am having some trouble. I looked through all of the FAQs, but couldn't find anything. I looked in the different configurations, and it wasn't much help. I was hoping someone could help me with this problem, if not through umask then some other method. Everytime I create a directory through FTP, it is automatically chmodded to 022. Now, I found in proftpd.conf it said: Umark 022 So I got smart and changed it to 755, as that is what I want directories to be at when they are created. But it still is at 022. Does anyone know how to solve this? Is there another way? Thanks.

I set the umask back to 022 and now when I upload files or make directories, the permissions are set tono one allowed to read/write/execute. Any other way around this? I do have umask 022 in a global block too.

Vincent Paglione wrote: > > Hello, > I set the umask back to 022 and now when I upload files or make > directories, the permissions are set tono one allowed to > read/write/execute. Any other way around this? I do have umask 022 in a > global block too. So your effective umask is 0777 now ? If your configuration does contain only umask 022 or less, your server is apparently started with this setting. Check the environment from which you start proftpd.

Vincent Paglione wrote: > > > So your effective umask is 0777 now ? If your configuration does contain > > only umask 022 or less, your server is apparently started with this > > setting. Check the environment from which you start proftpd. > > –job > > Can you please explain this a little more? Thank you. If you play around with the "umask" command in the shell you will get a good idea what it is doing. For instance: rm a; umask 0 ; touch a ; ls –l a rm a; umask 0777; touch a ; ls –l a rm a; umask 022 ; touch a ; ls –l a The umask is a property of a process and is inherited by its children, so if you start proftpd from a shell script that sets its umask it will start proftpd with that umask. Wait – hold that, i just looked in the source, and i think proftpd resets its umask to 022 (or to the value in the config file). So forget what i said about checking the environment. It must be something in that file. –job

Well from the way i undersood it, Umask 022 sets all directories to be 755 and all files 644, i think..... I think your problem came when it said umark instead of umask in the config file. Also by setting the umask to 755, your chmod becomes 022. To get your chmod from umask you subtract the umask # (In this case it's 022) from 777. So: 777 – 022 = 755 Sorry if I couldn't state this more clearly but I am not good at explaining things and I am still a Linux/Unix newbie.

execute should NOT be set as the default on files (on directories it is), so you can't do it. use chmod to get them executable if they need to be executable.

I'm running ProFTPD 1.2.0pre10 for a few weeks now on a server mainly use= d for customers websites. Someone made me notice this problem today : All files uploaded have 644 permissions whereas directories do have 755... Since I've put a "Umask 022" directive in my main server config part, I don't understand why I don't get 755 permission on created files... I used to have BeroFTPD working fine (for a few years), I've also tried t= o search for help or clues in the faq or in the ML archives but without success :(Here's what my config file looks like : [Begin proftpd.conf ...] ServerAdmin ftp@asi.fr ServerIdent on "FTP Server Ready – Webpro asi.fr" ServerType inetd DefaultServer on SystemLog /var/log/proftpd LogFormat default "%h %l %u %t \"%r\" %s %b" LogFormat detail "%{%a %b %d %H:%M:%S %Y}t %h %b %f (\"%r\= ")" LogFormat fichiers "%{%a %b %d %H:%M:%S %Y}t <%h> %b bytes, = %f" TimeoutStalled 300 RootLogin on # Umask 022 is a good standard umask to prevent new dirs and files # from being group and world writable. Umask 022 User nobody Group nogroup IdentLookups off DefaultRoot ~ !root DefaultTransferMode binary # Normally, we want files to be overwriteable. <Directory /*> AllowOverwrite on Umask 022 </Directory> [...End] (the <VirtualHost>s have been removed) The Umask in the <Directory /*> has been added just to test... but it doesn't work better. Any comments, ideas are very welcome since all my users have to use the chmod command for now! Thanks :)

This is usually treated in introductory Unix courses, it has been the custom since 1973 (+/–) to set the default file permissions on files to 666 and directories to 777, and subtract umask from that. Are your users uploading programs ? –job

Yes, the problem in having 644 (666 minus the 022 Umask according to what you say), is that all uploaded files (including cgi scripts) aren't executable... Is there a way to have them all in 755 mode again? Or mabe with a <Directory ...> only for ~/web/cgi–bin/ ? Thanks for any help, I'm desperate to have those cgis +x right after uplo= ad :)

I just looked in the source, but not long enough to find a spot where the default permissions could be set to 777. Perhaps in fs.c:std_creat – which would probably mess up the umask directive. The proper thing would be something like a new directive FileMode, with same context as Umask, so it could be used in .ftpaccess files. At my current programming speed it would take me 3 years if i would not get distracted. You could shellscript it in 6 minutes of course, from the log.

I had a big problem with the second parameter of the "Umask" directive. (Better said I couldn't get it worked.) I think there is a bug in the dir_check(_full) function in the file dirtree.c.

The functions check only the existing directories when searching for the right umask while when MKD (or XMKD) command is issued, the directory (usually :−) doesn't exist. After adding the test whether actual command is (X)MKD is umask set up properly.

I have been checking the ProFTPD archives, and am currently using ProFTPD 1.2.0pre10 to provide FTP functionality to the web server I administer. My problem is that the permissions on my server are screwed = up (mask 0755 for all files and directories). I have been able to set the directory permissions correctly using the Umask directive, but not the file directives (which need to be the directory equivalent of 0755, if that makes sense). This is actually two questions − one, is there still a problem with the Umask directive, and two, how do you calculate octal permissions? I understand that you need to generate the octal code from a list (which is how I came up with 0755) and subtract it from 777 to get the correct umask for directories and 666 for files, but doing 666−755 results in a negative, non−octal number − how would I convert the directory octal mask that I already have to the file octal mask that I need? Thanks in advance for any help you can give on either = of these questions.

The UNIX permissions are not octal. The permissions are a combination of the following values: 1=eXecute, 2=Write, 4=Read. To get the mask just subtract each value from 7: 777 755 022 <− This is the umask that you want.

Hm. Well, I've recompiled ProFTPD, this time with the Y2K patch that was just posted to the list, along with linuxprivs (thought I had it compiled in but I guess not..?). The server still completely ignores the second paramater on Umask.. however, if I set the Umask to 0027, the execute bits are automagically stripped from regular files (they stay on new directories though). By exploiting this behavior I've been able to set the default file permissions on upload files to the way I originally wanted them to be, and since users in the group public (A) have no real shell, (B) are denied use of the SITE CHMOD command, and (C) can't even talk to the machine on ports other than 20, 21, and 80, they'll stay that way. Woo.

However, I am getting one strange problem. I've got this in my config: <Directory ~> <Limit ALL> Order deny,allow DenyGroup public AllowAll </Limit> <Limit TYPE STRU MODE STOU ABOR STAT ALLO APPE REST READ LIST NLST \ STOR RETR DELE MKD RMD CWD RNFR RNTO XMKD XRMD XPWD XCUP \ NOOP PWD CDUP> Order allow,deny AllowAll </Limit> </Directory>

That long line I split a couple times isn't actually like that in my config.. it's all one line.

Strangely enough, this results in users who are not members of group public to be denied access to NLST while in their home directories. They can NLST anywhere else on the filesystem, but once they get within their home directory (and within the scope of those limits, I presume), they're denied NLST (but strangely enough, any of the other commands listed in the block with NLST work fine). I am completely confused by this. Anybody have any idea how this ends up happening? My goal is to ensure that as far as group public goes, anything not explicitly permitted is forbidden. In particular, we don't want users in group public changing permissions on their files, although after reading the RFC, there's a large amount of other stuff that I'm cutting them off from, too. Any user in any other group besides public should have access to the full set of ftp commands.

On Mon, Mar 06, 2000 at 09:08:57AM −0500, Matthew Eash wrote: > What is a umask that will create 775? 002 However it will only be the case for entries which would have the execute bit set on them by default − directories in otherwords. Files will have the mode of 664 with the above umask.

That's true (having forgot to account for that in my other comments :)

The directories need x set to allow 'searching'. You cannot execute a directory ;)

which, unless I'm having a fit of moronic stupidity (which could indeed be the case =) should set those permissions correctly. It half works − permissions on normal files are set right, but permissions on directories are set to the same thing as normal files. It seems that the second parameter to Umask isn't been recognized at all.. I've tried mucking about with it to see if there were any changes when the second parameter was changed, but no dice..

<Directory ~> Umask 0007 </Directory> Note I have two umask settings, one in the server context for files and one in the directory context for dirs. This is very strange that it works like this, but this is the only way I've found! I run 1.2pre9 and FreeBSD 3.3. I would be very happy if the 1.2pre10 with y2k and umask fixes released this month!

Setting the Umask

I am using Proftpd 1.2.0Pre9 on Linux on my ftp/web server. I would like all the files uploaded into a directory (/cgi−bin) to automatically receive the execute attribute (the rights should be "−rwxr−xr−x"). I tried to do it by setting a "Umask 000" in the proftpd.conf file but it doesn't work.

Proftpd will not do this automatically; clients should be able to do it with the "chmod" command, which translates to "SITE CHMOD", but I always get "permission denied" when i try it, i have not found why yet. Anyway, several ways to get it automatic: a script doing chmods every x minutes, or a script reading the logfile and looking for cgi−bins being uploaded, I have not done this, but this sounds like what you want. Though it is very thinly documented... The AllowFilter/DenyFilter: http://www.proftpd.org/docs/proftpdfaq−7.html#ss7.3

Chapter 7. Virtual Hosting

What is virtual hosting

When ftp was first concieved it was only possible to host a single ftp server on any given box. A method to increase the hosting density from one site per server to many sites on a given server grew. This many to one mapping is Virtual hosting. The design change in the server software was to allow for multiple unique ftp server configurations and binding these to particular interfaces on the server. Densities of hundreds of ftp sites per serving machine are not unknown on today's Internet.

IP address space considerations

Unlike the HTTP/1.1 protocol there is no method to host more than one ftp server on a single IP. HTTP/1.1 provides an additional transaction header, "Host:", to allow the server software to route the request to the correct virtual configuration. Currently this capability does not exist in the FTP protocol specification.

The only workaround to this limitation if address space is tight would be to host multiple servers on the same IP but different ports. however this is not a viable solution for a "normal" hosting farm because of the use of non−standard ports.

IETF draft standard

There is a draft standard draft standard under consideration with the IETF which extends and improves on the current 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.

Port based VirtualHosts

The Ports directive only makes sense within a proftpd.conf in standalone mode. In inetd mode the opening, closing and handling of the listening ports is handled entirely by the inetd super server daemon.

VirtualHost directive

basic usage and concepts of virtualhost

Setting up a basic virtual host

virtual hsot.

Preparing the system

The host system will need configuring with the additional IP addresses for each virtual host to be installed. On most unix systems this can be done as aliases on the primary ethernet interface or by dummy interfaces.

Minimal Configuration

<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.

Anonymous only servers

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>

---

vhost notes

I have tried to configure a name−based Virtual Host, but I always get to = the Directory which I configured in the <global>−area. My system: SuSE−Linux 6.3, ProFTP 1.2.0pre10. Yes, I've read the FAQ = :−). All Hosts should have the same IP (212.172.160.148).

```
my proftpd.conf:
# START
ServerName "Webmasters FTP-Server"
ServerType inetd
ServerAdmin admin@webmasters.at

DeferWelcome on

Port                  21
Umask                002
TimeoutLogin         120
TimeoutIdle          600
TimeoutNoTransfer    900
TimeoutStalled      3600
User    ftp
Group    nogroup
#DefaultRoot   ~
UseReverseDNS        off
ScoreboardPath   /var/run/proftpd
TransferLog   /var/log/proftpd/xferlog.legacy
LogFormat       default "%h %l %u %t \"%r\" %s %b"
LogFormat auth     "%v [%P] %h %t \"%r\" %s"
LogFormat write    "%h %l %u %t \"%r\" %s %b"

<Global>
 DisplayLogin     /usr/local/ftp/msgs/welcome.msg
 #DisplayFirstChdir    readme
 MaxClients 30
 AllowOverwrite     yes
 IdentLookups         off
 ExtendedLog /var/log/proftpd/access.log WRITE,READ write
```

```
 ExtendedLog  /var/log/proftpd/auth.log AUTH auth
  #ExtendedLog    /var/log/proftpd/paranoid.log ALL default
</Global>

<VirtualHost www.joydisco.at>
 ServerName  "www.joydisco.at"
 ServerAdmin  admin@joydisco.at
 #TransferLog  /var/log/proftpd/xferlog.www
 MaxClients  50
 #DefaultServer  on
 DefaultRoot  /www/www.joydisco.at
 AllowOverwrite  yes

</VirtualHost>
# END
```

> I have tried to configure a name−based Virtual Host, but I always get > to the Directory which I configured in the <global>−area. > All Hosts should have the same IP (212.172.160.148). > My system: SuSE−Linux 6.3, ProFTP 1.2.0pre10. Yes, I've read the FAQ Including http://www.proftpd.org/docs/proftpdfaq−5.html#ss5.6 ? I have tried to configure a Virtual Host, but I always get to the = Directory which I configured in the <global>−area. My system: SuSE−Linux 6.3, ProFTP 1.2.0pre10. my proftpd.conf: # START ServerName "Webmasters FTP−Server" ServerType inetd ServerAdmin admin@webmasters.at DeferWelcome on Port 21 Umask 002 TimeoutLogin 120 TimeoutIdle 600 TimeoutNoTransfer 900 TimeoutStalled 3600 User ftp Group nogroup #DefaultRoot ~ UseReverseDNS off ScoreboardPath /var/run/proftpd TransferLog /var/log/proftpd/xferlog.legacy LogFormat default "%h %l %u %t \"%r\" %s %b" LogFormat auth "%v [%P] %h %t \"%r\" %s" LogFormat write "%h %l %u %t \"%r\" %s %b" <Global> DisplayLogin /usr/local/ftp/msgs/welcome.msg #DisplayFirstChdir readme MaxClients 30 AllowOverwrite yes IdentLookups off ExtendedLog /var/log/proftpd/access.log WRITE,READ write ExtendedLog /var/log/proftpd/auth.log AUTH auth #ExtendedLog /var/log/proftpd/paranoid.log ALL default </Global> <VirtualHost 212.172.160.148> ServerName "www.joydisco.at" ServerAdmin admin@joydisco.at #TransferLog /var/log/proftpd/xferlog.www MaxClients 50 #DefaultServer on DefaultRoot /www/www.joydisco.at AllowOverwrite yes </VirtualHost> # END many thx for your help Thomas, tom@goisern.net Von: Falk Kuehnel [mailto:mailing−falk@salia.de] Gesendet am: Freitag, 24. M=E4rz 2000 13:15 An: proftpd@proftpd.org Betreff: [ProFTPD] Virtual FTP−Server Hi There! I was wondering if there is an way to set up severall VirtualFtpServers=20 with just one IP−Adress which can be connected to by anonymous users? I know this is not possible just by referring to the name of the server,=20 but if i understood correctly, it can be done by using different ports. I= s=20 there a howto, where the solution ist described? Thanx for your help Falk I'm using proftpd on several of my servers and I like its flexibility and security mechanisms. I'm running also a few virtualhosts (ip− and port−ba= sed). Now I would like to make a plan (or scheme) for adding new virtualhosts s= erving access for directories containing WWW services. Since particular users (i.e. website developers) should have access only to their projects= and often one project is developed by many of them, I want to make one (e.g. = port based) virtual host for one project. Do you see any disadvantages of such= a solution? How many port−based virtualhosts can proftpd (running on a linu= x system) handle? Are there any limitations other than CPU speed and RAM availability? this is my first post to the list. My question is: − is possible to create accounts that are only valid for FTP access?(I don't want that the user have a UNIX account) . Send me an example please. − IP restrict access doesn't works for me (I see an example in the documentation...but ...) so can someone send me his **proftpd.conf** where I can see that? On the users side of things, you just need to set the users' shell to /bin/false. Easy Way: In your proftpd.conf [or your virtual host line in there] AuthUserFile /config/ftp.passwd AuthGroupFile /config/ftp.group Then copy the SAME FORMAT AS /etc/passwd and /etc/group for example user:<hashed password>:<id>:<group>::<homedir>:/bin/false mark:x:980:100::/ftp/mark:/bin/false x being an encrypted password Enjoy! Its a great feature− especially if you make a quick 10 line web interface for the owner of vhosts to be able to change their own passwd files. −− Mike Krieger phyre@home.com On the users side of things, you just need to set the users' shell to /bin/false. − is possible to create accounts that are only valid for

FTP access?(I don't want that the user have a UNIX account) . Send me an example please. − IP restrict access doesn't works for me (I see an example in the documentation...but ...) so can someone send me his **proftpd.conf** where I can see that? Can I make a VirtualHost write to a separate wtmp file? I already have it writing to a separate xferlog but I'd like to write to a separate wtmp if possible. I'd like an easy way of seeing if someone is connected to a given VirtualHost. I guess I could compare the users that are still on (via ftpwho) to the output of netstat to see who connected to where. That's not very elegant though. Ideas? Another question, with DisplayGoAway, will it display the file to the user if they aren't allowed to connect in general, via a Limit block? The docs don't really say. They just say that it "will be displayed to the user if the class they're a member of has too many users logged in". It doesn't say if it will do that for all denied requests. In my case, I'm limiting this VirtualHost to certain IP ranges. I am limiting it to 75 anonymous users on that virtualhost but I don't care about displaying the file then, just when the user is connecting from and IP that isn't authorized. Any ideas if it will work or if there's a better way or if I'm just SOL? I had a 3rd question but I forgot it so it must not be important. Does proftpd support virtual directories (not necessarily virtual servers). Here's my situation, I wish to provide a group of users with access to a common directory (Group A), and another group of users with access to another common directory (Group B). Group A must not have access to Group B's files. Using AuthUserFile and AuthGroupFile to establish separate authentication. My hunch would be using multiple DefaultRoot entries. Something like: <Global> ... DefaultRoot /var/ftp/data/group−a groupa,!staff DefaultRoot /var/ftp/data/group−b groupb,!staff ... </Global> Would the above even be parseable or work? Read the FAQ and docs, but examples didn't quite apply. If anyone has any suggestions I'd appreciate it. −− George M. Ellenburg S1 Corp. That's hard to say. For security purposes, I'm faking the user/group in my anonymous block. DirFakeUser on Willie DirFakeGroup on Wildcat I'm not using seperate Auth files either. From the way that the AuthGroup config directives are worded, it would appear that all authentication is done via the the AuthUser/Group files (unless they aren't defined) but to make HideGroup/User work the files must be group or owned by the appropriate user on the actual system. I'm sure if there is a way around that. Just for the hell of it, chgrp 70 groupa's directory. Make sure 70 doesn't conflict with something else on your system. Maybe it does work. I've hidden a directory from users before. To see that directory you had to belong to a certain group. <Directory private> HideGroup crack </Directory> <Directory pub/consult/> HideGroup consult </Directory> I use both and they work well. I believe the file(s) have to be grouped and writable by the respect group. There are also other things you can do to keep the Group A from getting an "permission denied" error (even though the can't see the directory) when trying to cd into Group B directory. This is the situation: I have a <virtualhost> that allows anonymous logins, users can log in and upload files, but not download them. I need to give ONE user all permissions to the same virtualhost. What should my proftpd.conf look like? :) <VirtualHost xxx.xxx.xxx.xxx> DefaultRoot /usr/local/httpd/htdocs/ ServerName "xxx mainserver" ExtendedLog /var/log/proftpd.paranoid_log ALL default <Limit STOR> AllowAll </Limit> <Directory /> AllowOverwrite on <Limit STOR CWD CDUP> AllowAll </Limit> </Directory> </VirtualHost> why am i not allowed to write in any directory ??? when connecting this server ?? Hi, Oh. I just found the FAQ and it seems to answer the question. However the liink to the "draft standard" has gone stale: "File Not Found The requested URL /internet−drafts/draft−ietf−ftpext−mlst−08.txt was not found on this server."

DNS issues

Hosting VirtualHosts on a single IP

This is not possible to do in the same way as it is with Apache / http, this is not a failing of ProFTPD but rather a problem with the basic ftp protocol which as no method of uniquely identifying the target host during a session. The only work around at this time is to use a different primary port for each virtual if more than one per IP is required.

DNS entry not resolving

If the <VirtualHost> block is built using names rather than IP's there[A is a chance that a configuration reload will cause the server to die. Proftpd treats DNS resolution as a fatal error "Fatal: unable to determine IP address of `www.blah.com'". The best solutions to this problem are either to use raw IP addresses in the config thus removing all the resolution problems or to use the –t option to check the config prior to reloading.

Reloading the config

Two basic methods, stop and restart the server, or send a SIGHUP to the master proftpd lister. The scripts which come with both the normal distribution and the various packaged versions will do both. There is a minor bug in the SIGHUP handling which has not yet been found and dealt with. When reloading servers with many virtual hosts about 30% of the time the reload will fail in some way taking out the entire daemon.

Non resolving names

problem with non–existant names killing the daemon

Part of being a decent system administrator is solving the problem –– at the core. Apache lets you "pretend the problem doesn't exist" (yes, it spits out crap to stdout on runtime, but that doesn't necessarily mean you're going to be there to see it), allowing people to slack off and avoid doing their job in it's entirety. I care about the technicalities, as well as the principle behind the above situation. If my webserver (or FTP server) "skips" a host, it's more than likely going to cause a cust– omer or client to throw a fit.

DNS

It isn't absurd when you are running for than a few virtual hosts. Software isn't supposed to die at the first sign of trouble. If you had a few hundred virtual hosts, you wouldn't want apache to completely die because one of them wouldn't resolve, or wasn't aliased, etc.

UID/GID

many vhost entries death

What happens to connected users?

Wait...does it not make sense that you shouldn't have users logged in when you refresh the daemon?I'm no expert, but I've never seen something that will allow you to bind an application to a port that's already in use. It doesn't make sense to be able to do that. If you in effect kill the proftpd daemon and restart it...and orphan it's children, when it restarts it will attempt to bind to IPs and Ports. If a child process is still running on one of those IPs or Ports, how should proftpd handle it? Kill the process holding the port and then start? If it can't bind to the port or IP, I don't see how it will be able to recover.

Since no one else is responding to my message, I'm writing my own follow–up. The problem is worse than I thought: if you send a HUP to the master process (again, in standalone mode) to get it to recognize configuration changes, and someone is connected to one of the VirtualHosts, the whole process fails because it can't bind to that address. This is another case that I don't think should cause the entire server to fail. Am I the only one having these problems? Is anyone else running standalone? Thanks for any feedback.

I don't like the idea of kicking off all our connected users just to add a VirtualHost, and less, having all the servers die if it can't bind to 1 of 60 addresses. I understand that it can't bind to a port that is in use, but the bound process could understand that the configuration has changed and NOT stop/restart; I thought that was part of the benefit of sending a HUP rather than killing the whole thing and restarting (sort of like rebooting Windows for every little change you make)...

I just connected to one of the Virtual Hosts (running close to 200) on my FTP server (running ProFTPD in standalone) then HUP'd it while still connected which did not cause any fatal errors for me. It didn't cause my connection to drop either. HUP should only cause the process to reread the config not stop and restart as far as I know. So there is no logical reason why a HUP doesn't work, unless you attempt to use a domain name/IP that is not yet aliased to the NIC on that system. DNS shouldn't have anything to do with it as long as the IP/domain is aliased to the NIC, usually with ifconfig. But hey, I could be wrong, it has happened before :)

Chapter 8. Authentication

One of the core functions of every ftp is how it authenticates it's local users and assigns them the access rights to the ftp filesystem. At the moment Proftpd only supports the standard plaintext USER/PASS authentication interface, there is work underway to support crypted passwords, this will probably surface in the 1.3.x development series and the 1.4.x stable codebase which results from it.

Providing the backend to the user authentication interface there re a host of methods for storing user information and querying these databases of users for valid authentication sequences. The standard in ProFTPD is the Pluggable Authentication Modules system, or PAM. Support is also provided for the classic /etc/passwd and /etc/shadow password files as well as more "interesting" solutions such as SQL and LDAP.

Password files

Three variants on the password file theme are supported by the core Proftpd authentication code, these are /etc/passwd, /etc/shadow and uderdefined files by using the AuthUserFile and AuthGroupFile directives.

Support for passwd and shadow files is simple and well documented and conforms to the accepted standards and methods for handling these authentication sources. It should be noted that Proftpd unless told otherwise, by using "PersistantPassword off" directive, will attempt to open and leave open the passwd file throughout the life of the server process. /etc/passwd /etc/shadow AuthUserFile crypt, code fragment for generating cryoted passwords NIS ld.so.preload magic...

Pluggable Authentication Modules (PAM)

PAM has become a standard method of providing secure authentication services within the UNIX environment in the past few years. PAM acts as the interface between the program or system daemon and the underlying authentication methods. It's great strengths are the higher levels of security it affords to the system administrator and it's flexability. As the name suggests the coding interface is common for all PAM supported methods, however behind the scenes many different methods of authentication can be supported. Even to the extent of (for example) supporting RADIUS for ftp access and /etc/shadow for telnet.

ProFTPD requires PAM version 0.59 or better. The pam_sm_open_session system call is not provided in earlier versions and is a requirement of the PAM implementation within Proftpd.

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.

AuthPAMAuthoritive

AuthPAMAuthorative defaults to "off"" allowing other authentication methods to get a look in at authentication time. Setting this to "on" will break support for external files such as AuthUserFile.

Preloading

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.

Typical PAM configuration

Proftpd itself should need little or no configuration to support PAM, however some configuration of the PAM subsystem may be required. One of the most common problems encountered when configuring and using Proftpd is a missing /etc/pam.d/ftp file, if this file isn't installed the authentication requests will fail.

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

Linux

Most of the development of Proftpd is done on Redhat based systems, however this should not prevent users of other distributions running the daemon without problems.

Example 8−1. Generic Linux PAM config

```
#%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
```

Redhat Linux

Example 8−2. Redhat 6.* configuration

```
#%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

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.

Example 8−3. SuSe configuration

```
/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.

Example 8−4. FreeBSD configuration

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

pam_sm_open_session errors

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

Conficts with PAM authentication

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

```
PersistentPasswd   off
AuthPAMAuthorative off
```

SQL

You are in a maze of twisty SQL statements, none alike.

This section has been removed completely and needs a complete re−write to account for the new approach to SQL handling as of 1.2.0

UserPassword

I've been waiting patiently and trying new versions (right now, I have 1.2.0pre7−3 from debian potato), but UserAlias in anonymous ftp now forces me to use the password of the user I alias to, and not the user I log in as.

Example 8−5. ...

```
<Anonymous ~ftp/sub/dir/>
```

```
  AnonRequirePassword on
  RequireValidShell off
  User ftp
  Group nobody

#  UserPassword ftp encpasswd
  UserPassword ftpuser1 encpasswd1
  UserPassword ftpuser2 encpasswd2
  (...)

  UserAlias ftpuser1 ftp
  UserAlias ftpuser2 ftp
  (...)
</Anonymous>
```

So, I used to be able to log as ftpuser1 and use ftpuser1's password with older versions of proftpd. Now I'm forced to uncomment the "UserPassword ftp encpasswd" line and everyone would have to log with ftp's password. I really do not want to go back to wuftpd (with which I got this to work jusr fine). 1) Can this still be made to work somehow? 2) If not, how do I provide anonymous ftp access to a select number of users, each with their own password? (I'd rather not have to put users in /etc/passwd and /etc/group)

Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) authentication from within Proftpd is provided via the mod_ldap module, which is not compiled in by default. For information on compiling in additional modules go back and read the chapter on installing Proftpd.

As of version 1.2 most of the annoying bugs have been removed and the code is of suitable quality to provide a stable authentication backend. The module became part of the distribution as of version 1.1.

What is LDAP

LDAP is a distributed, hierarchical directory service access protocol which is used to access repositories of users and other network− related entities. Because LDAP is often not tightly integrated with the host operating system, information such as users may need to be kept both in LDAP and in an operating system supported nameservice such as NIS. By using LDAP as the the primary means of resolving these entities, these redundancy issues are minimized and the scalability of LDAP can be exploited. (By comparison, NIS services based on flat files do not have the scalability or extensibility of LDAP or X.500.)

The object classes and attributes defined below are suitable for representing the aforementioned entities in a form compatible with LDAP and X.500 directory services.

Example 8−6. A typical configuration fragment

```
LDAPServer                      "localhost"
LDAPPrefix                      "dc=horde,dc=net"
LDAPDN                          "cn=thedn,dc=horde,dc=net"
LDAPDNPass                      "ldap_dnpass"
LDAPNegativeCache       on
```

Ldap notes

I try to compile Proftpd 1.2.0pre9 with the ldap support. According to info on the homepage of the ldap module, the mod_ldap.c is in the /modules directory and I run configure with the −−with−modules=mod_ldap but make always complains about missing lber.h and ldap.h (which are found in the OpenLDAP package). Does that means that I have to compile (or copy some files from?) OpenLDAP on the computer? My aim is to use a remote LDAP server, not a locally installed one (and I don't want, if possible, to install a LDAP server on this machine). How can I do? Sorry if my question seems a bit simple to the "gurus"! :−)

mon avis you don't need to install a full ldap server on your system,= but you need to have a set of ldap library (openldap, netscape...) and the corres= ponding include headers. I recommend you to install the openldap and build the li= braries only.

That's correct. For OpenLDAP, you can build the client header files, libraries, and utilities _only_ by saying: ./configure −−disable−slapd −−disable−slurpd when you build OpenLDAP. =C0 mon avis you don't need to install a full ldap server on your system,= but you need to have a set of ldap library (openldap, netscape...) and the corres= ponding include headers. I recommend you to install the openldap and build the li= braries only.

I try to compile Proftpd 1.2.0pre9 with the ldap support. According to info on the homepage of the ldap module, the mod_ldap.c is in the /modules directory and I run configure with the −−with−modules=mod_ldap but make always complains about missing lber.h and ldap.h (which are found in the OpenLDAP package). Does that means that I have to compile (or copy some files from?) OpenLDAP on the computer? My aim is to use a remote LDAP server, not a locally installed one (and I don't want, if possible, to install a LDAP server on this machine). How can I do? Sorry if my question seems a bit simple to the "gurus"! :−) I've a proftpd authenticating users against a ldap server. The users are not unix users, then I can't use the normal quotta system. Does have proftpd an internal system to limit the size of a directory?? How can I control the size of each user directory?? thanks,

On Wed, Jan 12, 2000 at 11:05:15AM +0100, Juli=E1n Romero wrote: > I've a proftpd authenticating users against a ldap server. > The users are not unix users, then I can't use the normal quotta system. Hmmmm − considering that I've not dealt with the mod_ldap stuff this may well be a silly question. Does each of your users have a unique UID on the system ? Or are you using DefaultRoot and a single UID for all users ? If you've got a unique UID for each user then you *can* still use the quota system as it actually uses UIDs for the work − the user name to UID map is performed by the quota commands. That said, the quota commands often accept UIDs in the place of usernames. It has to be said that the standard quota stuff (at least under Solaris) is painful to use in automatic systems − its often easier to dig into the quota system and code your own programs to control the quota system than use the system provided ones.

Thanks to Jim, i succeeded in linking mod−ldap with LDAP−C SDK librai= ries (−lpthread −lldapssl30 is a good way)... But now, when I try to run proftpd, I get an error message which says= : − Fatal: Group: Unknown group 'nogroup'. ...I tried to add "ftp−master", "nobody", "nogroup" entries to my LDA= P server...but nothing changes! Thanx for any help...Peter, could you send me part of your proftpd.co= nf, or ldif entries you had to add on your LDAP server?

here's my proftpd.conf (very basic one, i thought...) −−−−−−−−−−−−−−−−−−−− # This is a basic ProFTPD configuration file (rename it to # 'proftpd.conf' for actual use. It establishes a single server # and a single anonymous login. It assumes that you have a user/group # "nobody" and "ftp" for normal operation and anon. ServerName "Serveur test" ServerType standalone DefaultServer on #PAMConfig ftp #AuthPAMAuthoritative off LDAPServer test.rouen.men.fr LDAPDNInfo xxxxxxxxxxxxxxxx LDAPDoAuth on "ou=3D..." LDAPDoUIDLookups off LDAPDoGIDLookups off LDAPNegativeCache on Port 21 Umask

022 MaxInstances 30 User nobody Group nobody <Directory /*> AllowOverwrite on </Directory> <Anonymous ~> User ftp Group ftp UserAlias anonymous ftp MaxClients 10 DisplayLogin welcome.msg DisplayFirstChdir .message <Directory *> <Limit WRITE> DenyAll </Limit> </Directory> <Directory incoming> <Limit READ WRITE> DenyAll </Limit> <Limit STOR> AllowAll </Limit> </Directory> </Anonymous> −−−−−−−−−−−−−−−−−−−−−− % % Thanks to Jim, i succeeded in linking mod−ldap with LDAP−C SDK librairies % (−lpthread −lldapssl30 is a good way)... % % But now, when I try to run proftpd, I get an error message which says: % − Fatal: Group: Unknown group 'nogroup'. % % ...I tried to add "ftp−master", "nobody", "nogroup" entries to my LDAP % server...but nothing changes! % % LDAPServer test.rouen.men.fr % LDAPDNInfo xxxxxxxxxxxxxxx % LDAPDoAuth on "ou=..." % LDAPDoUIDLookups off % LDAPDoGIDLookups off % LDAPNegativeCache on In theory, you can have the users referenced in your proftpd.conf User/Group config directives in the LDAP database. I really haven't done any testing with a situation like that, though, but it *should* work. I usually have my User/Group users listed in /etc/passwd and /etc/group and just use the LDAP directory for user authentication. If you choose to reference LDAP−only users/groups in your User/Group config directives, you'll need to set both LDAPDoAuth and LDAPDoGIDLookups to on.

I am trying to build mod_ldap module (2.5.2) with proftpd on Solaris 2.6. I use Netscape DirectoryServer API libldapssl30.so. When I do make with the package it says one error: cannot find "llber" library. I remember if use Netscape directory server API 3.0, we need not llber library. Where to get llber (which looks for openldap on linux)? Does any one succeed to use proftpd authenticate with ldap on Solaris ? DO you get the same compiling problem? When I do ./configure −−with module=mdo_ldap, for some system check the answer is no, does that matter?

I have a problem in running Proftpd from command line. I configure it as default run by "inet" NOT standalone. I add few entries in proftpd.conf but when I run it, error message say Fatal: unknown configuration directive 'LDAPDNInfo', any first directive start with 'LDAP' get the same error. I searched the source code no place read in "proftpd.conf" file , what is wrong with my mod_ldap module? Please drop me few lines if you have any idea where this config file is read in and how it proftpd talk to Unix Solaris nation unix.pam.so 1 module and pam talk to mod_ldap? Atcuallt where is the mod_ldap lib installed? In other email group, I succeeded allow wu_ftp to talk to standard pam_ldap and nss_ldap module for ftp user to authenticate with a remote LDAP server. I know here Proftp has its native LDAP support. Can anyone help me describe the code architecture? I am new to proftp.

A quick question is that, you user login as uid and pass then search on the proconfigured base in mod_ldap.conf file which is static. But uid may not unique in a whole LDAP server, for example a ISP company providing service for many domains. The search base is root. How you avoid this problem? OR, if use login as uid@domain.com can you parse it into "uid" and "domain.com" then do the ldap search with dn in mod_ldap.conf plus "domian.com" as new search base? OR your ProFTP can prompt user for a "domain" in addition to "user:" "password"? Which way is easier to deal with this problem?

I have problems with authentication when using mod_ldap. I'm using Solaris2.6, Netscape Directory Server, 1.2.0pre10 and mod_ldap−2.5.2. ProFTPD is running standalone. When trying to login I get: −−−−−−−−−− 220 ProFTPD 1.2.0pre10 Server (ProFTPD) [hostname] Name (IP−address:user): profuser 421 Service not available, remote server has closed connection Login failed. −−−−−−−−−− The "profuser" is registered in the NDS and has the objectclass posixaccount. In the errorlog of the NDS it looks like bind is ok but what happens after that??? errorlog: −−−−−−−−−−− [28/Feb/2000:15:28:28 +0100] − new connection on 44 [28/Feb/2000:15:28:28 +0100] − listening for connections on 0 [28/Feb/2000:15:28:28 +0100] − activity on 44r [28/Feb/2000:15:28:28 +0100] − read activity on 44 [28/Feb/2000:15:28:28 +0100] − add_pb [28/Feb/2000:15:28:28 +0100] − listening for connections on 0 [28/Feb/2000:15:28:28 +0100] − get_pb [28/Feb/2000:15:28:28 +0100] − do_bind [28/Feb/2000:15:28:28 +0100] − BIND dn="cn=Directory Manager" method=128 version=2 [28/Feb/2000:15:28:28 +0100] − => get_ldapmessage_controls

[28/Feb/2000:15:28:28 +0100] − <= get_ldapmessage_controls no controls [28/Feb/2000:15:28:28 +0100] − do_bind: version 2 method 0x80 dn cn=Directory Manager [28/Feb/2000:15:28:28 +0100] − => slapi_pw_find value: "password" [28/Feb/2000:15:28:28 +0100] − <= slapi_pw_find matched "password" using scheme "clear" [28/Feb/2000:15:28:28 +0100] − => send_ldap_result 0:: [28/Feb/2000:15:28:28 +0100] − flush_ber() wrote 14 bytes to socket 44 [28/Feb/2000:15:28:28 +0100] − <= send_ldap_result [28/Feb/2000:15:28:28 +0100] − listener got signaled [28/Feb/2000:15:28:28 +0100] − listening for connections on 0 [28/Feb/2000:15:28:28 +0100] − activity on 44r [28/Feb/2000:15:28:28 +0100] − read activity on 44 [28/Feb/2000:15:28:28 +0100] − add_pb [28/Feb/2000:15:28:28 +0100] − listening for connections on 0 [28/Feb/2000:15:28:28 +0100] − get_pb [28/Feb/2000:15:28:28 +0100] − PR_Recv(2588624) 0 (EOF) [28/Feb/2000:15:28:28 +0100] − listener got signaled −−−−−−−−−−−− In proftpd.conf I have these values: −−−−−−−−−−−− LDAPServer localhost LDAPDNInfo "cn=Directory Manager" password LDAPDoAuth on "ou=users,ou=customers,o=organization" LDAPDoUIDLookups off LDAPDoGIDLookups off LDAPNegativeCache on LDAPHomedirOnDemand off LDAPDefaultAuthScheme clear −−−−−−−−−− Is there anyone who knows what happens here after the Bind??? I added the "allowedservices" attribute but it makes no difference. How is it used? Have you edited mod_ldap.c? Did you do something else to get it going?

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−− > What attribute is "allowedservices"? I don't have that attribute > in my Directory Server. Is it a standard attribute or something > you've created? > Something I added in each LDAP user definition Same problem I got. Verify that your user as the following attributes (LDAP definition): userpassword homedirectory allowedservices (set it to FTP) It's the minimum requirements. Joakim Br=E5n=E4s (QRA) wrote: > What attribute is "allowedservices"? I don't have that attribute > in my Directory Server. Is it a standard attribute or something > you've created? > Something I added in each LDAP user definition −− Laurent PIERRE T=E9l:01 47 33 82 84 Fax:01 47 33 76 98 E−mail: laurent.pierre@alcove.fr ** Alc=F4ve lib=E8re votre informatique... Web: http://www.alcove.fr **

Just a wild guess − what about escapeing the whitespace ? like : LDAPDNInfo cn=3DDirectory\ Manager,dc=3Ddatelec,dc=3Dcom [ ] > % I guess this new release had correctly fixed the solaris bugs. > % > % I can connect to proftpd without this "signal 11" error. Nice job ! > > Good, I'm glad to hear it. > > % But I have now another issue : > % > % My cn for my Netscape Directory server is : Directory Manager > % > % So my proftpd.conf file seems like : > % > % > LDAPServer ldap_shagga > % > LDAPDNinfo cn="Directory Manager",dc=datelec,dc=com [ ] > [misc. snippage] > % But I guess your module doesn't parse correctly the cn field cause in my ldap > % server logs I get this error : > % > % > [16/Feb/2000:12:47:20 +0100] conn=7371 op=0 BIND dn="cn="Directory" method=128 version=2 > > Config−file parsing is taken care of by proftpd; oops ... sorry :) > it basically splits config > file parameters on whitespace. I think that the quotes in the middle of the > paramter are confusing it. What happens if you try: > > LDAPDNInfo "cn=Directory Manager,dc=datelec,dc=com" [ ] I've tried : same error... So I've to change the name of my Cn...... Exact ?

I guess this new release had correctly fixed the solaris bugs. I can connect to proftpd without this "signal 11" error. Nice job ! But I have now another issue : My cn for my Netscape Directory server is : Directory Manager So my proftpd.conf file seems like : > LDAPServer ldap_shagga > LDAPDNinfo cn="Directory Manager",dc=datelec,dc=com micr0sc0n > LDAPDoAuth on "dc=People,dc=datelec,dc=com" > LDAPDoUIDLookups off > LDAPDoGIDLookups off > But I guess your module doesn't parse correctly the cn field cause in my ldap server logs I get this error : > [16/Feb/2000:12:47:20 +0100] conn=7371 fd=164 slot=164 connection from 192.168.120.164 to 192.168.120.160 > [16/Feb/2000:12:47:20 +0100] conn=7371 op=0 BIND dn="cn="Directory" method=128 version=2 > [16/Feb/2000:12:47:20 +0100] conn=7371 op=0 RESULT err=32 tag=97 nentries=0 etime=0 > [16/Feb/2000:12:47:20 +0100] conn=7371 op=−1 fd=164 closed − B1 > Normally that should be (like a good connection): > [16/Feb/2000:12:47:59 +0100] conn=7372 fd=164 slot=164 connection from 192.168.120.165 to 192.168.120.160 > [16/Feb/2000:12:47:59 +0100] conn=7372 op=0 BIND dn="cn=Directory Manager" method=128 version=2 > [16/Feb/2000:12:47:59

+0100] conn=7372 op=0 RESULT err=0 tag=97 nentries=0 etime=0 >

> I released mod_ldap v2.5.1 on Saturday. It adds support for authenticated > binds and also fixes all known bugs up until this point in time (most > notably the mod_ldap−segfaults−under−solaris bug). Authenticated binds > allows mod_ldap to support any password encryption scheme that your LDAP > server supports; it will bind to your LDAP server with the credentials > listed by LDAPDNInfo and fetch all user information except for userPassword. > It will then re−bind to the LDAP server as the FTP user who is attempting > to log in with the user−supplied password. If the bind succeeds, the user > is allowed access. http://horde.net/~jwm/software/mod_ldap/ > > I know that a bunch of people on the list are using mod_ldap, so I figured > it would be of enough interest to post here. If there's a more appropriate > place for this (or you'd rather I didn't announce new versions here), please > let me know.

% I guess this new release had correctly fixed the solaris bugs. % % I can connect to proftpd without this "signal 11" error. Nice job ! Good, I'm glad to hear it. % But I have now another issue : % % My cn for my Netscape Directory server is : Directory Manager % % So my proftpd.conf file seems like : % % > LDAPServer ldap_shagga % > LDAPDNinfo cn="Directory Manager",dc=datelec,dc=com [ ] [misc. snippage] % But I guess your module doesn't parse correctly the cn field cause in my ldap % server logs I get this error : % % > [16/Feb/2000:12:47:20 +0100] conn=7371 op=0 BIND dn="cn="Directory" method=128 version=2 Config−file parsing is taken care of by proftpd; it basically splits config file parameters on whitespace. I think that the quotes in the middle of the paramter are confusing it. What happens if you try: LDAPDNInfo "cn=Directory Manager,dc=datelec,dc=com" [ ]

I'm trying to use proftpd with the ldap module (v2.0). I've made a beautiful compilation of proftpd with some others modules (the problem is the same without quota and ratio) > shagga (root) /tmp/proftpd−1.2.0pre10 > ./proftpd −l > Compiled−in modules: > mod_core.c > mod_auth.c > mod_xfer.c > mod_site.c > mod_ls.c > mod_unixpw.c > mod_log.c > mod_pam.c > mod_ratio.c > mod_ldap.c > mod_quota.c > My configuration file is something like that : > ServerName "Internal FTP Server" > ServerType StandAlone > DefaultServer on > Port 21 > > User nobody > Group nogroup > > MaxInstances 30 > > TimeoutStalled 300 > > DisplayLogin welcome.msg > DisplayFirstChdir .message > > RootLogin on > > AuthPAMAuthoritative off > > LDAPServer ldap_shagga > LDAPDNinfo cn=admin,dc=datelec,dc=com password > LDAPDoAuth on "dc=People,dc=datelec,dc=com" > LDAPDoUIDLookups off > LDAPDoGIDLookups off > When I start proftpd (proftpd −d 5 −n) and i try to connect to the server, I get this message : > shagga − ProFTPD 1.2.0pre10 standalone mode STARTUP > shagga (snardone.datelec.ch[192.168.120.165]) − connected − local : 192.168.120.164:21 > shagga (snardone.datelec.ch[192.168.120.165]) − connected − remote : 192.168.120.165:3451 > shagga (snardone.datelec.ch[192.168.120.165]) − ProFTPD terminating (signal 11) > Not too much debug infos !! Other important point : when I sniff if something come out off my ftp server I can see : NOTHING ! I guess something is wrong in my configuration file (I'm really not a LDAP "guru"). Thanks in advance, Stephan

> Yes, I admit that mod_ldap needs some serious debugging info added; the next > release is pretty frozen right now, but definitely in the next release. > After mod_ldap is called to parse its config file entries, it logs a summary > of all its config paramters, if you run proftpd normally (letting it fork > and without debugging), do you see something like this in your syslogs? Can > you look in your LDAP server's logs to see if mod_ldap is querying the LDAP > database yet? Also, what operating system are you using? Even if I try to start proftpd normally I can't see anything additional debug. I also do not see any connections from proftpd to my ldap (netscape Directory 4.1). I'm running proftpd under Solaris 2.6 (Sun). > > > % Other important point : when I sniff if something come out off my ftp server I can see : NOTHING ! > % > % I guess something is wrong in my configuration file (I'm really not a LDAP "guru"). > % > % PS: that's may be due to the compilation with netscape SDK.... > > I compiled pre10 on this machine (Slack Linux 4), and mod_ldap works fine > with your config (without AuthPAMAuthoritative, I don't have access to a > PAMified machine). But I'd like to find out if you see the config−summary > syslogged anywhere before I'm lead to believe that it's an SDK problem. So,

I'm now looking for an Sun solaris 2.6 binaries plus libraries (ldap and lber).... I am seeing the same thing as Stephan (signal 11, proftpd closes connection). My proftpd config is the same as well. I'm on a solaris 7 box that has PAM and I have tried the proftpd directive "AuthPAMAuthoritative" set to "on" and "off" with the same result. I have carried out John's instruction to Stephan (run proftpd in standalone mode without debugging and check syslog for mod_ldap config parsing entries. However, I do not see anything. What I do see logged when a ftp client connection is made is the following: /var/adm/messages <snip> Jan 27 07:35:08 thumbsuck proftpd[20813]: thumbsuck.mweb.co.za (net−61−51.mweb.c o.za[196.2.61.51]) − ProFTPD terminating (signal 11) Jan 27 07:35:08 thumbsuck proftpd[20813]: thumbsuck.mweb.co.za (net−61−51.mweb.c o.za[196.2.61.51]) − ProFTPD terminating (signal 11) Jan 27 07:36:51 thumbsuck proftpd[20816]: thumbsuck.mweb.co.za (localhost[127.0. 0.1]) − ProFTPD terminating (signal 11) Jan 27 07:36:51 thumbsuck proftpd[20816]: thumbsuck.mweb.co.za (localhost[127.0. 0.1]) − ProFTPD terminating (signal 11) <snip> Seems odd that proftpd is logging what appears to 2 duplicate lines for each client connection. Note that I tried from two different clients. I also do not see any connections from proftpd to my ldap (openldap). Hopefully this may assist in pinning down this problem? Paul Gamble. On Wed, Jan 26, 2000 at 08:25:48PM +0100, Stephan Nardone wrote: % % I'm trying to use proftpd with the ldap module (v2.0). % % I've made a beautiful compilation of proftpd with some others modules % (the problem is the same without quota and ratio) % % > shagga (root) /tmp/proftpd−1.2.0pre10 > ./proftpd −l % > Compiled−in modules: % > mod_core.c % > mod_auth.c % > mod_xfer.c % > mod_site.c % > mod_ls.c % > mod_unixpw.c % > mod_log.c % > mod_pam.c % > mod_ratio.c % > mod_ldap.c % > mod_quota.c I believe you need something like MySQL or PostgreSQL to store persistent ratios across FTP sessions; mod_ldap doesn't support storing ratio information (yet, I'll have to look to see what's involved). % My configuration file is something like that : % [snip] % > LDAPServer ldap_shagga % > LDAPDNinfo cn=admin,dc=datelec,dc=com password % > LDAPDoAuth on "dc=People,dc=datelec,dc=com" % > LDAPDoUIDLookups off % > LDAPDoGIDLookups off This looks fine. % When I start proftpd (proftpd −d 5 −n) and i try to connect to the server, I get this message : % % % > shagga − ProFTPD 1.2.0pre10 standalone mode STARTUP % > shagga (snardone.datelec.ch[192.168.120.165]) − connected − local : 192.168.120.164:21 % > shagga (snardone.datelec.ch[192.168.120.165]) − connected − remote : 192.168.120.165:3451 % > shagga (snardone.datelec.ch[192.168.120.165]) − ProFTPD terminating (signal 11) % % Not too much debug infos !! Yes, I admit that mod_ldap needs some serious debugging info added; the next release is pretty frozen right now, but definitely in the next release. After mod_ldap is called to parse its config file entries, it logs a summary of all its config paramters, if you run proftpd normally (letting it fork and without debugging), do you see something like this in your syslogs? Can you look in your LDAP server's logs to see if mod_ldap is querying the LDAP database yet? Also, what operating system are you using? % Other important point : when I sniff if something come out off my ftp server I can see : NOTHING ! % % I guess something is wrong in my configuration file (I'm really not a LDAP "guru"). % % PS: that's may be due to the compilation with netscape SDK.... I compiled pre10 on this machine (Slack Linux 4), and mod_ldap works fine with your config (without AuthPAMAuthoritative, I don't have access to a PAMified machine). But I'd like to find out if you see the config−summary syslogged anywhere before I'm lead to believe that it's an SDK problem. On Thu, Jan 27, 2000 at 07:48:45AM +0200, Paul Gamble − MWeb wrote: > I am seeing the same thing as Stephan (signal 11, proftpd closes > connection). My proftpd config is the same as well. I'm on a solaris 7 box It may not help much, but its probably worth pointing out that signal number 11 is a segmentation violation (SIGSEGV, see /usr/include/sys/signal.h) which indicate that some code within ProFTPD is doing bad things to memory. −− On Wed, Jan 26, 2000 at 08:25:48PM +0100, Stephan Nardone wrote: % % I'm trying to use proftpd with the ldap module (v2.0). % % I've made a beautiful compilation of proftpd with some others modules % (the problem is the same without quota and ratio) % % > shagga (root) /tmp/proftpd−1.2.0pre10 > ./proftpd −l % > Compiled−in modules: % > mod_core.c % > mod_auth.c % > mod_xfer.c % > mod_site.c % > mod_ls.c % > mod_unixpw.c % > mod_log.c % > mod_pam.c % > mod_ratio.c % > mod_ldap.c % > mod_quota.c I believe you need something like MySQL or PostgreSQL to store persistent ratios across FTP sessions; mod_ldap doesn't support storing ratio information (yet, I'll have to look to see what's involved). % My configuration file is something like that : % [snip] % > LDAPServer ldap_shagga % > LDAPDNinfo cn=admin,dc=datelec,dc=com password % > LDAPDoAuth on "dc=People,dc=datelec,dc=com" % > LDAPDoUIDLookups off % > LDAPDoGIDLookups off This looks fine. % When I start proftpd (proftpd −d 5 −n) and i try to connect to the server, I get this message : % % % > shagga − ProFTPD 1.2.0pre10 standalone mode STARTUP % > shagga (snardone.datelec.ch[192.168.120.165]) − connected − local : 192.168.120.164:21 % > shagga (snardone.datelec.ch[192.168.120.165]) − connected − remote : 192.168.120.165:3451 % > shagga (snardone.datelec.ch[192.168.120.165]) − ProFTPD terminating (signal 11) % % Not too much debug infos !! Yes, I admit that mod_ldap needs some serious debugging info added; the next release is pretty frozen right now, but definitely in the next release. After mod_ldap is called to parse its config file entries, it logs a summary of all its config paramters, if you run proftpd normally (letting it fork and without debugging), do you see something like this in your syslogs? Can you look in your LDAP server's logs to see if mod_ldap is querying the LDAP database yet? Also, what operating system are you using? % Other important point : when I sniff if something come out off my ftp server I can see : NOTHING ! % % I guess something is wrong in my configuration file (I'm really not a LDAP "guru"). % % PS: that's may be due to the compilation with netscape SDK.... I compiled pre10 on this machine (Slack Linux 4), and mod_ldap works fine with your config (without AuthPAMAuthoritative, I don't have access to a PAMified machine). But I'd like to find out if you see the config−summary syslogged anywhere before I'm lead to believe that it's an SDK problem. Hi all, I am trying to use Apache's mass virtual hosting features to create a mass virtual hosting server for web data. Trouble is, to upload their data, users need to use ftp to do it. I am looking for an ftp server daemon wchich will let me do the following: − authenticate username/password in LDAP − chroot access to their home directory − NO POSIX ACCOUNT NEEDED in the LDAP server (easier to maintain, more secure) Can proftpd (with LDAP patches) do this?

P On Thu, Dec 02, 1999 at 03:28:04PM +0100, Graham Leggett wrote: % I am trying to use Apache's mass virtual hosting features to create a % mass virtual hosting server for web data. Trouble is, to upload their % data, users need to use ftp to do it. % % I am looking for an ftp server daemon wchich will let me do the % following: % % − authenticate username/password in LDAP Sure, mod_ldap can do this. % − chroot access to their home directory This is a part of ProFTPD itself, and has no problems with mod_ldap as far as I can see. % − NO POSIX ACCOUNT NEEDED in the LDAP server (easier to maintain, more % secure) Currently, mod_ldap uses the posixAccount objectclass; if you really don't want to use it, you can modify this behavior, but it will require modification of the mod_ldap source to change the names of the attributes that the module is looking for from the LDAP database. I'm thinking of making this compile−time configurable in the next release of mod_ldap[1]; a couple other people have mentioned that they don't want to use the posixAccount objectclass. [1] mod_ldap v2.0 will be released any day now; I've got 95% of the docs done, just gotta get the web site updated. I'll think about adding a non−posixAccount objectclass to my todo for the next release. If anybody wants an advance copy of v2.0, please let me know.

% Ideally what I am looking for is something that can match the % VirtualDocumentRoot directive in the Apache mod_vhost_alias module. % % Here you define a template of some kind that tells Apache where to find % the document root directory based on the DNS name of the website. % % It would be great if ProFTPd could do this also, either getting the DNS % name from an attribute in LDAP, or by using the username+SomeDNSSuffix % to correspond.

Hm, that would be interesting. Maybe a config option to vary the LDAPPrefix based on the IP address the remote user connected to. I'll have to check it out.

% The reason why I don't want to use the posixAccount objectclass is % because I cannot seem to find any widely available LDAP editors that % allow me to edit an object using it. What editors have you looked at, and what objectclasses have they supported? I'm still considering making objectclass a compile−time option, I just need some other objectclasses to support. :−)

% In addition, the need for posix user and group ids is a pain, something % has to assign them, and ensure these numbers are unique. This is too % much work just for ftp.

mod_ldap 2.0 changes that; it's the first release that will let you run in a pure virtual environment (an "ftp toaster" kind of deal). You can assign a single default UID/GID in your proftpd.conf and also create home directories on demand (when the user logs in for the first time). (Thanks to Bert Vermeulen <bert@be.easynet.net> and Krzysztof Dabrowski <brush@pol.pl> for ideas/patches in this area.)

% > Hm, that would be interesting. Maybe a config option to vary the LDAPPrefix % > based on the IP address the remote user connected to. I'll have to check it % > out. % % The Apache mass virtual hosting places many sites under one IP address, % so determining the hostname this way won't work -- but it will work in % the case everyone is given their own IP address.

Okay; I didn't consider that possibility. In that case, something like that for the FTP protocol in general won't work; there's no way to do virtual hosting without an IP address for each FTP virtual host. I've heard that there's been some draft work on changing this situation, I don't have any URLs handy, but I think that some have been posted to the list in the past.

Yup, it's at http://horde.net/~jwm/software/proftpd-ldap/. It works well for me, and I've had reports of v2.0 working well at other (some large) sites. Let me know how things go.

Why use LDAP over SQL?

> >- Because LDAP is a standard, SQL is not. > > Excuse me? I think you're misinformed here, as SQL is a standard. What > various companies have done with propietary "extensions" is another issue, > but you can always choose not to use them and stick with core. But still , > I think I understand what you're getting at: portability.

SQL thinks it's a standard, but I'm talking in practical terms. Each vendor seems to have it's own variation on syntax, as well as access libraries, otherwise you have to install and correctly configure ODBC, a real pain.

The lack of a standard SQL schema is also a problem. The way application A stores it's user information is usually completely different to the way that application B does it, because there is no "right" way of doing it. Assuming it's even possible, making application A and B share the same schema is usually lots of work. Yuck.

> I have been considering using LDAP, which is what prompted my inquiry in > the first place. Feedback I've gotten from a few people implementing it is > that it works great, but does not scale as well as SQL, is more resource > intensive, and that for large user bases (e.g. couple hundred thousand) is > much slower. But I've not yet delved into this thoroughly enough to make a > sound evaluation.

LDAP scales much better than SQL because of the way the database is designed. You can spread your data logically across multiple machines, allowing different people to have different access to data sets (such as the US people being able to edit their userids, and the Europe people being able to edit thier userids, but neither can edit the other's, if you want it like that), while at the same time keeping the tree looking like a single logical data set. You can also (as we do here) mirror your data across many LDAP servers, so if one server goes down it won't take out your applications.

> >- LDAP's replication, scalability and fault tolerance support is simpl= e > >to configure and use, SQL's is vendor specific and unnecessarily > >complicated. > > I have been considering using LDAP, which is what

prompted my inquiry i= n > the first place. Feedback I've gotten from a few people implementing i= t is > that it works great, but does not scale as well as SQL, is more resourc= e > intensive, and that for large user bases (e.g. couple hundred thousand)= is > much slower. But I've not yet delved into this thoroughly enough to ma= ke a > sound evaluation.

I'm speaking for a commercial LDAP implementation, Netscape–iPlanet Direc= tory Server 4.11. It's fast like hell! If you use the personalisation features of my.netscape.com, you can see that's it fast. And my. netscape has over 20million users in their ldap servers and each user has around 400 attri= butes.

PcWeek measured on a 4 CPU NT box over 5000 authentication / second with = this LDAP server. As Paul Tavernier wrote it really uses cool caching, and it's one of the = most stabe product I've ever seen. But if you have a lot's of write operation LDAP is not about handling the= m very fast.

1. performance is better for read operations (what an authentication is) 2. price 3. easier to implement failover than with eg an Oracle.

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

Other authentication methods

...

NIS/YP

Be sure to read the documentation on the PersistentPasswd configuration directive.

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.

Encrypted passwords

No support yet.

SecureID

No support yet.

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 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

Chapter 9. DefaultRoot and other issues

Locking users into a directory (chroot)

Preventing users from moving round the filesystem is a must for many system admistrators. Proftpd achieves this functionality using the chroot() system call. This call moves the system root directory to the specified location. Anonymous connections do this by default setting the chroot() to the directory specified in the <Anonymous> directive. For more normal users the DefaultRoot directive is required

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:

Example 9−1. Simple DefaultRoot setup

```
#
# A simple DefaultRoot setup
# limiting all users to their $HOME
#
<VirtualHost myhost.mynet.foo>
DefaultRoot ~
</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).

The chroot() system call simply moves the root (or "/") directory to a specified point within the filesystem. When implemented properly this has the effect of jailing a user into a particular branch of the filesystem directory structure. The security advantages of this approach are easily seen and it is a common method used by programmers and system administrators worldwide to enhance their local security models.

Security Implications

This approach should not be considered a high security model it has a number of flaws, not least of which is that chroot jails can be broken out of. Breaking a chroot is not a trivial task but it's nowhere near to being impossible and a competant cracker should be able to breach the security offered by chroot. This said it is still a valuable tool in the armoury of the admin.

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

This prevention method was developed by Carole Fennelly and her partner. Have a look at the August 1999 Security column of SunWorld Online for the article − see http://www.sunworld.com/sunworldonline/swol−08−1999/swol−08−security−2.html

It's worth noting that almost all FTP servers retain root privileges throughout their life, though they may revert to lower levels where possible. This is because the daemon needs to keep its root privs around when a user is logged in for things like opening sockets on ports less than 1024.

Non−root server issues

The chroot() system call will not work under a non−root ftp server process, the call requires root privileges. 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.

Required files

Some OSs require files to always exist in an environment for certain things to work. For example you need the following files available for chroot()ed work under Solaris 2.5.1:

/dev/tcp
/dev/ticotsord
/dev/udp
/dev/tcp

So to actually make Proftpd work in a chrooted environment it may be necessary to create $HOME/etc/ $HOME/dev/ and similar directories and create certain files. While files are required will vary from system to system and are generally outside the scope of this guide

---

svc.conf

For systems which require the svc.conf file, tyically Digital Unix systems, it is essential that a copy of this file is placed within the chroot ($HOME/etc/svc.conf) or all attempts to authenticate will fail with error messages similar to those below.

```
331 Password required for test.
Password:
230 User test logged in.
ftp> dir
200 PORT command successful.
getsvc: stat of /etc/svc.conf failed
ftp> pwd
getsvc: stat failed: No such file or direc
getsvc: stat of /etc/svc.conf failed
getsvc: stat failed: No such file or direc
150 Opening ASCII mode data connection for
ftp> pwd
226 Transfer complete.
257 "/" is current directory.
ftp> dir
```

```
200 PORT command successful.
getsvc: stat of /etc/svc.conf failed
ftp>
```

Example 9−2. Sample svc.conf file

```
# WARNING: This file is MANDATORY !
#
# Setup recommendation: As you add distributed services to database
#       entries, it is recommended that "local" is the first service.
#       For example:
#                       passwd=local,yp
#
# Note: White space allowed only after commas or newlines.
#
# File Format
# −−−−−−−−−−−
# database=service,service
#
# The database can be:
#       aliases
#       group
#       hosts
#       netgroup
#       networks
#       passwd
#       protocols
#       rpc
#       services
# The service can be:
#       local
#       yp
#       bind (hosts ONLY)
#
aliases=local
group=local
hosts=local,bind,yp
netgroup=local
networks=local
passwd=local
protocols=local
rpc=local
services=local
SECLEVEL=BSD   # for backwards compatibility ONLY
```

Finer grained control

There are situations where different classes of user should be limited in different ways. For example, developers working on a site should only be able to see the section they are responsible for, whereas the sysadmins and supervisors need to have a wider view on the server. This can be acomplished either by setting the $HOME of each user to the location on the disk which is most appropriate, or more commonly by using system groups.

Example 9−3. DefaultRoot, modified by system group

```
#
```

```
# A more complex setup where all users are locked into
# their home except those in group 'staff' who are
# locked into /u2/allweb
#
<VirtualHost myhost.mynet.foo>
DefaultRoot ~ !staff
DefaultRoot /u2/allweb staff
</VirtualHost>
```

Symlinks and chroot()

Contributor: Rod Whitworth <rodw at witworx dot com>

There have been many questions on the ProFTPD user mailing list about why symlinked directories are not visible to chrooted users (this includes <Anonymous> users as well as users restricted using DefaultRoot. This document is intended to clarify the issues and discuss some ways of achieving what is commonly desired.

These issues are not specific to ProFTPD, but rather to the workings of a Unix system. First, a brief review of how links work, and why chroot(2) poses such a problem. Then a look at ways around the issue.

How Links Work

There are two types of links in Unix: hard and symbolic.

A hard link is a file that is, for all intents and purposes, the file to which it is linked. The difference between a hardlink and the linked file is one of placement in the filesystem. Editing the hardlink edits the linked file. One limitation of hard links is that linked files cannot reside on different filesystems. This means that if /var and /home are two different mount points in /etc/fstab (or /etc/vfstab), then a file in /var/tmp cannot be hardlinked with a file in /home:

```
> pwd
/var/tmp
  > ln /home/tj/tmp/tmpfile tmplink
ln: cannot create hard link `tmplink' to `/home/tj/tmp/tmpfile': Invalid cross-device link
```

A symbolic link (also referred to as a "symlink") is a file whose contents contain the name of the file to which the symbolic link points. For example:

```
lrwxrwxrwx   1 root     root           11 Mar  2  2000 rmt -> /sbin/rmt
```

The file rmt contains the nine characters /sbin/rmt. The reason symbolic links fail when chroot(2) is used to change the position of the root (/)of the filesystem is that, once / is moved, the pointed–to file path changes. If, for example, if chroot(2) is used to change the filesystem root to /ftp, then the symlink above would be actually be pointing to /ftp/sbin/rmt. Chances that that link, if chroot(2) is used, now points to a path that does not exist. Symbolic links that point to nonexistent files are known as dangling symbolic links. Note that symbolic links to files underneath the new root, such as symlinks to a file in the same directory:

```
> pwd
/var/ftp
  > ls -l
-rw-r--r--   1 root     root            0 Jan 16 11:50 tmpfile
  lrwxrwxrwx   1 root     root            7 Jan 16 11:50 tmplink -> tmpfile
```

will be unaffected; only paths that point outside/above the new root will be affected.

Filesystem Tricks

A typical scenario is one where "DefaultRoot ~" is used to restrict users to their home directories, and where the administrator would like to have a shared upload directory, say /var/ftp/incoming, in each user's home directory. Symbolic links would normally be used to provide an arrangement like this. As mentioned above, though, when chroot(2) is used (which is what the DefaultRoot directive does), symlinks that point outside the new root (the user's home directory in this case) will not work. To get around this apparent limitation, it is possible on modern operating systems to mount directories at several locations in the filesystem.

To have an exact duplicate of the /var/ftp/incoming directory available in /home/bob/incoming and /home/dave/incoming, use one of these commands:

```
    * Linux (as of the 2.4.0 kernel):

mount --bind /var/ftp/incoming /home/bob/incoming
  mount --bind /var/ftp/incoming /home/dave/incoming

    * BSD (as of 4.4BSD):

mount_null /var/ftp/incoming /home/bob/incoming
  mount_null /var/ftp/incoming /home/dave/incoming

    * Solaris:

mount -F lofs /var/ftp/incoming /home/bob/incoming
  mount -F lofs /var/ftp/incoming /home/dave/incoming
```

The same technique can be used for <Anonymous> directories, which also operate in a chroot()ed environment.

As usual, more information can be found by consulting the man pages for the appropriate command for your platform. The commands for other flavors of Unix will be added as needed.

In order to have these tricks persist, to survive a system reboot, the /etc/fstab (or /etc/vfstab) file may need to have these mounts added. Consult your local fstab(5) (or vfstab(4) for Solaris) man pages for more information.

Chapter 10. Anonymous Servers

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.

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.

Create a new unix group on your system named `anonftp'. Please each user who will have per−user anonymous FTP in this group. Create an `anon−ftp' and `anon−ftp/incoming' directory in each user's home directory. 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>
```

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.

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>

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.

Additional anonymous accounts

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):

Example 10−1. Access control using LIMIT

```
<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.

Secure upload facilities

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 XCWD>
        AllowAll
     </Limit>
     <Limit READ DELE MKD RMD XMKD XRMD>
        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.

Chapter 11. Using AuthUserFiles

Originally by TJ Saunders

The FTP protocol is old, stemming from the days of Telnet, before security came to be the relevant issue it is today. One of the protocol's biggest flaws, in today's security−conscious world, is the transmission of passwords "in the clear", unencrypted, easily visible to network sniffers. There are several ways of attempting to deal with this flaw.

ProFTPD allows for the definition of "virtual" users: users who do not have accounts on the host machine, whose account information is defined in other sources. The passwords for these users are then specific only to FTP access, and thus do not expose shell access (ssh, hopefully) to unauthorized users. These alternative account information sources include SQL tables (via mod_sql), LDAP servers (via mod_ldap), CDB files (via mod_auth_cdb), and other system files (via the AuthUserFile and AuthGroupFile configuration directives). The proftpd server can be configured to use multiple account information sources simultaneously as well, allowing for flexible support of a range of environments.

This document focuses on the use of the AuthUserFile and AuthGroupFile directives. Several questions often arise about the use of these directives. In general, the answers to those questions apply to other authentication/account sources, too.

Formats

Any configured AuthUserFile is used instead of /etc/passwd, not in addition to it; similarly for AuthGroupFile and /etc/group. The format of an AuthUserFile is the same as /etc/passwd (man passwd(5)), and the format of an AuthGroupFile is the same as /etc/group (man group(5)). There is an ftpasswd script available that can be used to create and update these files.

It is important to note here that not all flavors of Unix use these formats; one notable exception is FreeBSD. With FreeBSD, user account information is stored in binary database files as opposed to ASCII files. The C library on this platform also lacks the functions necessary for making the AuthUserFile and AuthGroupFile directives work properly. In this case, proftpd uses an internal implementation of the missing functions to read AuthUserFiles and AuthGroupFiles. This internal implementation requires that AuthUserFiles have the traditional passwd(5) format:

```
username:password:uid:gid:gecos:homedir:shell
```

and that AuthGroupFiles have this format:

```
groupname:grouppasswd:gid:member1,member2,...memberN
```

The ftpasswd script mentioned creates files in these required formats.

Choice of IDs

The choice of IDs for your users is important; the operating system does not deal with user processes in terms of their user names, it knows only of the numeric identity of a process. In general, it is best if each user has a unique positive user ID (negative IDs are an ugly hack, and proftpd will complain if they are used).

However, in some mass−hosting environments such as ISPs, the number of users is greater than the number of IDs available. In these cases, you can give each user the same ID; additional steps should then be taken to make sure that each user is isolated from affecting other users' files. These additional steps are to make sure each user has a unique home directory, and then to restrict each user to their respective home directories by having

```
DefaultRoot ~
```

in your proftpd.conf.

There is also currently an issue with using the same usernames in different authentication schemes (e.g. having the same username in both /etc/passwd and a mod_sql SQL table); read this thread for more information:

http://www.proftpd.org/proftpd−l−archive/full/msg16600.html

Shadow passwords

There really is no need for an AuthShadowFile directive. The purpose of a shadow file is separate sensitive information (e.g. passwords) from other account information (username/UID/GID, etc). Programs like /bin/ls often reference the passwd file in order to display user/group names rather than numbers; these programs do not really need that sensitive information. Rather than relying on programs like /bin/ls to ignore the sensitive information, libraries were developed to split the information into /etc/passwd, /etc/shadow (and similarly for /etc/group, but very few administrators use group passwords anymore). Some operating systems, most notably FreeBSD, though, chose a different form of information separation. Since FreeBSD maintains account information in binary database files, the shadow libraries mentioned above are not used. Instead, FreeBSD returns the sensitive information to the calling program only if it has sufficient (i.e. superuser) privileges.

When proftpd uses an AuthUserFile, it is looking for all of the account information, including the password. And since AuthUserFiles are specific to proftpd, there is no need to split any passwords out of an AuthUserFile into an AuthShadowFile. As the documentation states, an AuthUserFile need not reside inside a chroot() filesystem, which means that users can be effectively isolated from having access to that AuthUserFile. At that point, the only consideration is making sure that the permissions on the AuthUserFile are sufficient for the server to have access, but no other users.

Permissions

As the AuthUserFile and AuthGroupFile files are meant to be drop−in replacements for their system cousins, there are a few caveats. /etc/passwd and /etc/group are normally world−readable on modern Unix systems. This allows programs like /bin/ls to map system ID numbers to more legible names; sensitive information in the /etc/passwd and /etc/group is normally stored elsewhere, in restricted shadow files. The proftpd server thus assumes that it will not need special privileges to read an AuthUserFile or an AuthGroupFile. The process will access any AuthUserFiles and AuthGroupFiles with the credentials of the user and group configured via the User and Group directives. The files may contain sensitive information, so they should not have as open of permissions as /etc/passwd and /etc/group. The most paranoid setting will have user−read−only permissions for those files, and have the files be owned by the user configured for the relevant server via the User directive. Hopefully the server administrator has created a new account on the system just for the ftpd daemon.

ID−to−name mapping

A consequence of which to be aware when using an AuthUserFile is the difference between that AuthUserFile's mapping of system IDs to names, and the mapping in /etc/passwd. This may catch some system administrators unawares when they go to check the ownership of files uploaded by some user whose account is defined in an AuthUserFile, and find those files being reported as being owned by different users and/or groups by /bin/ls. Keep in mind that /bin/ls is using /etc/passwd, not the AuthUserFile. This issue crops up with any alternative account information source, not just AuthUserFiles.

If you are using the same UID/GID for your users, e.g. in a mass hosting environment, one trick you might like to do is make all of the files, as listed by the server, appear to be owned by the logged in user. This is done using the DirFakeUser and DirFakeGroup directives, like this:

```
# make listed files appear to be owned by the logged−in user
  DirFakeUser on ~
  DirFakeGroup on ~
```

These directives are purely cosmetic, and in no way change the real ownership of files. This may cause some confusion on the client side in some cases, if the user sees a file that is reported to be owned by them, and the permissions on the file show user access is allowed, and yet the client is unable to access the file. HideNoAccess can help in situations like this.

Chapter 12. Configuration for NAT

Contributed by Tobias Ekbom

Basic information

A NAT is a system that acts like a proxy, but on "packet level". When a computer on your local network connects to a computer on the Internet, the NAT replaces the "from" information of packets with it's own address, making your local network invisible to the Internet.

For server systems, NAT can improve security and enable multiple servers to be accessed as a single IP.

This is done by allowing certain ports forwarded "inwards" to the local network. However, the part of the FTP protocol known as "Passive" mode is not by default compatible with NAT solutions. But NAT functionality is possible with ProFTPD from versions 1.2rc2, and this document shows you how.

For details on NAT configuration, read the Linux IP−masq HOWTO (http://www.linuxdoc.org/HOWTO/IP−Masquerade−HOWTO.html) or search for information concerning your OS of choice.

Configuring ProFTPD

First configure your ProFTPD install so that it works right from inside the NAT. There are example configuration files included with the source.

Then add the directive "MasqueradeAddress" in your etc/proftpd.conf file to define the public name or IP address of the NAT:

```
MasqueradeAddress       ftp.mydomain.com
        −or−
MasqueradeAddress       123.45.67.89
```

Now your ProFTPD will hide it's local address and instead use the public address of your NAT.

However, one BIG problem exists.

The passive FTP connections will use ports from 1024 and up, which means that you must forward all ports 1024−65535 from the NAT to the FTP server! And you have to allow lots of (possibly) dangerous ports in your firewalling rules!

Have no fear, simply use the PassivePorts directive in your etc/proftpd.conf to control what ports ProFTPD uses:

```
PassivePorts 60000 65535        # These ports should be safe...
```

Now start the FTP daemon and you should see something like

```
123.45.67.89 − Masquerading as '123.45.67.89' (123.45.67.89)
```

Configuring Linux

This example is for Linux kernel version 2.2.x with ipchains and ipmasqadm. The examples below assume that your FTP server has local address 192.168.1.2.

First we need to enable NAT for our FTP server. As root user:

```
echo "1">/proc/sys/net/ipv4/ip_forward
ipchains -P forward DENY
ipchains -I forward -s 192.168.1.2 -j MASQ
```

Now we load the autofw kernel module and forward ports 20 and 21 to the FTP server:

```
insmod ip_masq_autofw
ipmasqadm autofw -A -r tcp 20 21 -h 192.168.1.2
```

Then we forward ports for Passive FTP. In our etc/proftpd.conf file we restriced passive ports to 60000–65535, so that's what we'll use here:

```
ipmasqadm autofw -A -r tcp 60000 65535 -h 192.168.1.2
```

Now you can try to login to your FTP server from a computer on the Internet!

Security

Setting up a ProFTPD install that allows PASV mode connections requires that a range of ports is forwarded from the NAT to the local network. This could be a security hazard, but since you can specify what port range to use, you are still able to setup relatively tight firewalling rules.

To be sure that you have no other processes listening on the ports you have specified for Passive FTP, use a port scanner such as nmap:

```
nmap -sT -I -p 60000-65535 localhost
```

If the result says something like

```
All 5536 scanned ports on localhost (127.0.0.1) are: closed
```

then you should be safe.

Chapter 13. Configuring ProFTPD for FTP over SSH

Chapter by TJ Saunders

Basic premise

The File Transfer Protocol is an old protocol, and like many of the older protocols it was developed in a time when security was not of such a paramount concern. One aspect of FTP that reflects this is the transmission of passwords, used to prove to the server that the client is who they say they are, "in the clear", unencrypted, visible to any packet sniffer. Protocols like SSH, and its scp program, are replacing FTP as a means to securely transfer files among hosts.

However, many people still prefer to use their standard FTP clients. What would be easiest would be a way to allow such clients to function while transparently providing the encryption necessary for today's networks. Can this be done? Yes – after a fashion. This document aims to describe how to tunnel FTP over an SSH channel, providing for secure transmission of the user's password.

Client Configuration

The trick to encrypting the password is to make use of ssh's local port forwarding capability. It works by creating a sort of proxy on the local host that listens to some high–numbered port. Any traffic sent to that port is encrypted, and forwarded to the configured remote address and port. (Note: the prior instructions' flaw was that the local port forwarding request was sent to localhost, rather than to the remote server, which effectively set up the encrypted channel between localhost and itself). Here's how to set up a local port forward:

```
ssh -Llocal-port:remote-addr:remote-port user@host
```

This says to listen on port local–port on localhost, and to send that encrypted traffic to host's remote–addr at port remote–port. To use this trick to secure an FTP session (to be specific, the control channel, through which passwords are transmitted, will be encrypted; the data channel will not), it would look like:

```
ssh -f -L3000:ftpserver:21 ftpserver 'exec sleep 10' && ftp localhost 3000
```

Note that the choice of local–port is arbitrary. Using port 3000 is not a requirement. This trick also requires that the ftp client use passive mode data transfers, so make sure to use a client that understands FTP passive mode.

Remember that only the control connection is encrypted, not the data connection: any data you transfer (eg directory listings, files uploaded or downloaded) are still sent "in the clear". Your password (and the other FTP commands sent by the client) is not.

Server Configuration

The server side of the connection needs some configuration to make the ssh tunneling work as well. First, the client need a valid account on the remote host is necessary in order to support the ssh tunnel. A valid shell is not strictly necessary for these ssh tunnels, though; something like:

```
#!/bin/sh
  sleep 10
```

would suffice. This would prevent the FTP users from logging in, yet give them enough time to establish a port forwarded ssh connection. With this sort of quasi−shell (although, strictly speaking, there are better, more restrictive shells than this example, as it could be escaped from), one can ssh over any command:

```
ssh −l test −f −L3000:ftpserver:21 ftpserver true && ftp localhost 3000
```

You'll also need to use the AllowForeignAddress configuration directive to your configuration file:

```
AllowForeignAddress on
```

or proftpd will reject the passive transfer connections and log:

```
SECURITY VIOLATION: Passive connection from {host} rejected
```

Note that the use of the server host's DNS name, or its IP address, in the setting up of the ssh tunnel assumes that the routing of traffic to the host's own IP address will be short−circuited in the kernel and thus not actually be transmitted on the wire. On modern kernels this is a fair assumption, but may not always be the case. For the truly paranoid, use of localhost or 127.0.0.1 as the remote−addr parameter in the ssh tunnel command would cause traffic on host to be sent over its loopback address, and not over the network. However, this will prevent data transfers from working altogether. The author is presently developing a patch for scenarios like this; if interested, please contact him directly via email.

III. Advanced configuration

Table of Contents

Chapter 14. Access controls

It's a rare day in the life of the systems administrator when he doesn't want to limit the access a user or network has to a resource. Whether the limits are prohibitions on access or limits on the amount of use that can be made they are a fact of life on anything but the simplest of configurations.

Access limitation

Controlling timeouts

There are a number of methods for controlling how long the daemon waits for connections to complete, and how long a connection is held open by the daemon while waiting for traffic. The times given in the various Timeout* directives are in seconds.

```
TimeoutNoTransfer              900
TimeoutIdle                    900
```

Setting the various timeouts too high can result in problems because of connections being held open too long. Setting the timeout to zero is definately recommended against, infinate timeouts are bad without a very good reason.

```
TimeoutIdle
Syntax: TimeoutIdle seconds
Default: TimeoutIdle 600
Context: server config
Compatibility: 0.99.0 and later
```

The TimeoutIdle directive configures the maximum number of seconds that proftpd will allow clients to stay connected without receiving any data on either the control or data connection. If data is received on either connection, the idle timer is reset. Setting TimeoutIdle to 0 disables the idle timer completely (clients can stay connected for ever, without sending data). This is generally a bad idea as a "hung" tcp connection which is never properly disconnected (the remote network may have become disconnected from the Internet, etc) will cause a child server to never exit (at least not for a considerable period of time) until manually killed

Abusive users

Your attempt to post to the ProFTPD mailing list at i've got a problem with people who "hammer" my ftp site when its quite busy..........i have my max connections per IP set at 1, but when people hammer, they try connecting several times per second! this sometimes allows them to connect many times, often using up all the users for the account. i have looked through the documentation and have been unable to find something to eliminate this. is there anything? something i can configure to, let's say "if user connects X times in X seconds, then ban. or ignore for X minutes" etc. any help would be appreciated. eric

You mean you have MaxClientsPerHost set to 1 already ? If so then this and a user from the same IP address can connect more than once then this is presumably a bug (check the logs to make sure that they are coming from the same IP address) or do you mean that they connect and disconnect quickly ?

Access classes

How do I go about using the class directive in proftpd. Under wu−ftpd, I could limit function base on define class (anonymous, guest, etc) How would you do this under proftpd? (ie anonymous could only do this, real user from this ip could do this, where real user from this other IP could do this!) Class Syntax: Class "name" limit|regex|ip value Default: None Context: server config Compatibility: 1.2.0pre9 and later Controls class based access. Class base access allows each connecting IP to be classified into a separate class. Each class has its own maximum number of connections. limit sets the maximum number of connections for that class name, regex sets a hostname regex (POSIX) for inclusion in the class and ip sets an IP/netmask based inclusion. The default class is called default. Example: Classes on Class local limit 100 Class default limit 10 Class local regex .*foo.com Class local ip 172.16.1.0/24 This creates two classes, local and default, with local being everything in *.foo.com and 172.16.1.* combined. Classes Syntax: Classes on|off Default: Off Context: server config Compatibility: 1.2.0pre9 and later Controls class based access. Enables class based access control. see: Class

Ok, been playing again :) Example of a real life classes based config.

Example 14−1. Configuration using classes

```
ServerName                      "Frostbite FTPserver"
ServerType                      standalone
DeferWelcome                    on
Port                            21
Umask                           002
User                            ftp
Group                           ftp
TransferLog                     /var/spool/syslog/proftpd/xferlog.legacy
DefaultRoot                     /ftp/ftp.linux.co.uk
TimeoutLogin                    120
TimeoutIdle                     600
TimeoutNoTransfer               900
TimeoutStalled                  3600
ScoreboardPath                  /var/run/proftpd
LogFormat                       default "%h %l %u %t \"%r\" %s %b"
LogFormat                       auth    "%v [%P] %h %t \"%r\" %s"
LogFormat                       write   "%h %l %u %t \"%r\" %s %b"
UseReverseDNS                   off
MultilineRFC2228                on
AllowFilter ".*/[a-zA-Z0-9 ]+$"
Port 0

<Global>
        DisplayLogin            welcome.msg
        DisplayFirstChdir       readme
        AllowOverwrite          yes
        AccessGrantMsg          "Welcome to Tux's kingdom oh chilly %u"
        DisplayConnect /ftp/ftp.linux.co.uk/login.msg
        IdentLookups         off
        ExtendedLog             /var/spool/syslog/proftpd/access.log WRITE,READ write
        ExtendedLog             /var/spool/syslog/proftpd/auth.log AUTH auth
        ServerIdent             on "Linux.co.uk server"
        AllowForeignAddress     on
        PathDenyFilter          "(\.htaccess)|(\.ftpaccess)$"
</Global>
```

```
# -----------------------------------------------------
# ftp.linux.co.uk ("Linux.co.uk FTP Archive")
# Contact : zathras@linux.co.uk
#
<VirtualHost 195.200.4.15>
ServerAdmin             zathras@linux.co.uk
ServerName              "Linux.co.uk FTP Archive"
TransferLog             /var/spool/syslog/xfer/ftp.linux.co.uk
MaxLoginAttempts        3
RequireValidShell       no
DefaultRoot             /ftp/ftp.linux.co.uk
User                    linux
Group                   linux
AllowOverwrite          yes
DefaultServer           yes
LoginPasswordPrompt     off "Wibble"
#
# Allow 50 users from Local network, local WAN
# Limit everyone else to 20 connections
#
Classes on
Class local limit 50
Class default limit 20
Class local regex .*ftech.co.uk
Class local ip 195.200.0.0/19
Class local ip 212.32.0.0/17
Class local ip 192.168.0.0/16

<Anonymous /ftp/ftp.linux.co.uk>
        User                    ftp
        Group                   ftp
        UserAlias               anonymous ftp
        RequireValidShell       no
        ##MaxClients              200
        MaxClientsPerHost       5 "Please don't be a hog and open any more sessions"
        AccessGrantMsg          "Welcome to Tux's kingdom oh chilly anonymous user"
        AllowForeignAddress     on
        #
        # Global upload, no download or browsing.
        #
        <Directory pub/incoming/*>
                AllowOverwrite off
                <Limit STOR CWD XCWD CDUP MKD>
                        AllowAll
                </Limit>
                <Limit READ DELE WRITE DIRS>
                        DenyAll
                </Limit>
        </Directory>
</Anonymous>
</VirtualHost>
```

Stopping permission changes

As of 1.2.0rc1 it is possible to prevent end users from altering the permissions on files in directories they have modify rights to. The AllowChmod directive, which defaults to deny, can be used to allow chmod on a server, global, virtualhost or directory basis. Giving a much finer degree of control over what the end users are capable of.

Bandwidth control

The bandwidth control mechanisms inside Proftpd have changed dramatically during the 1.2.0 development and release cycle. The original 'Bandwidth' directive has been removed and replaced with a nubmer of 'Rate*' directives. These only work on a per session basis with no scope for limiting on a VirtualHost basis or a netblock basis. This functionality is planned for the 1.3.x development branch.

Example 14−2. Simple throttling config

```
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.

Limiting the total usage by a VirtualHost

How can i achieve bandwidth restriction to depend on current user. one should have 1000 bytes per second write, the other 100bytesps for example. RateWriteBPS is global, isn't it ?

I have a delimma that I need opinions and ideas on. At K−State our Internet1 bandwidth is getting pretty hefty. They are trying cut back on resources until they can get a grip on the napster problem we're having. One of the things that they wanted done to my public mirror server (see sig) was rate limiting. I don't terribly mind it but I don't want to do it to Internet2 Universities and other participants. My possible solution was to send all Internet1 users to one hostname, and all Internet2 users to a Virtualhost. The Internet1 would have standard rate limiting features (could someone give me an opinion on numbers for this please? I haven't used it before) and the Interet2 virtualhost would be unlimited. That's a reasonable idea. I'm trying to get a list of IP blocks for all groups on Internet2 from the 'Net2 people themselves. Then I thought of another thing. Can it be done within the same host? Could I do a allow, deny for both groups within the same host−−one of them gets the good speeds, the other gets limited? How? Would it be better to create an internet2 user on my system that 'Net2 people could login in with and then have it check to see if they really are 'Net2 people (according to the IP block)? I'm having trouble deciding what to try and/or which to use. Any thoughts or advice? LinuxPPC 2000 will be coming out this weekend and will be hosted by me. I need to get something in place before that happens.

Example 14−3. Rate limiting

```
<Anonymous ~ftp>
    # ...etc....
        RateWriteBPS                    16384           # all writes at max 16K/s
```

```
        <Directory slow>
                RateReadBPS             1024            # 1K/s max
                RateReadFreeBytes       64000           # less than 64KB at full speed
                RateReadHardBPS         on              # after 64KB xfer _forced_ down to 1K/s
        </Directory>
        <Directory pub/win95>
                RateReadBPS             8192            # 8K/s max
                RateReadFreeBytes       256000          # until 256KB files at full speed, then 8K/s
        </Directory>
</Anonymous>
```

And a comment: if the normal cases one should not use "RateReadHardBPS on", it is cruel to the users. :)

I'm currently running 1.2.0pre10 in inetd mode, and the RateReadBPS directive works well for users who are individually defined within my proftpd.conf. However, I would like to restrict total outgoing FTP bandwidth, and it looks like this should be possible with RateReadBPS. >From section 6.16 of the FAQ:

"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."

A section of my proftpd.conf might look like this: <Global> MaxClients 6 RateReadBPS 10000 </Global> It looks like total bandwidth should be limited to ~60 KB/s (after restarting inetd). However, this is not the case. Any suggestions?

On Mon, Feb 28, 2000 at 05:22:12PM −0500, dboyles@r75h121.res.gatech.edu wrote: > "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." > > A section of my proftpd.conf might look like this: > > <Global> > MaxClients 6 > RateReadBPS 10000 > </Global> > > It looks like total bandwidth should be limited to ~60 KB/s (after > restarting inetd). However, this is not the case. Any suggestions? inetd (with or without tcpd) may be the problem. Try running proftpd in standalone mode as a single daemon.

Hi, I'm having a problem getting the RateReadBPS limit to work. I've = put it in the 'global', in the 'directory' and in the body. No matter = where I put it, it doesn't seem to work. Can someone please give me an = example, or something? This is not an anon site, and my users all have = different logins, but go to the same dir tree, where they are jailed.

Thanx!! ServerType inetd DisplayConnect /home/ftp/.ftpmess=20 DefaultServer on maxclients 22 "Sorry, max number of users has been = reached" maxclientsperhost 1 "Sorry, only 1 connection per user is = allowed" Group ftp User ftp AllowStoreRestart on ExtendedLog /var/log/ftp.log all=20 LogFormat default "%h %l %u %t \"%r\" %s %b" Umask 0000 TimeoutNoTransfer 600 TimeoutIdle 600 TimeoutStalled 600 AccessGrantMsg "User access granted for %u." <Global> <Limit LOGIN> AllowGroup ftp </Limit> DefaultRoot /home/ftp </Global> <Directory /*> AllowOverwrite off </Directory>

I'm having a problem getting the RateReadBPS limit to work. I've = put it in the 'global', in the 'directory' and in the body. No matter = where I put it, it doesn't seem to work. Can someone please give me an = example, or something? This is not an anon site, and my users all have = different logins, but go to the same dir tree, where they are jailed.

Thanx!! ServerType inetd DisplayConnect /home/ftp/.ftpmess=20 DefaultServer on maxclients 22 "Sorry, max number of users has been = reached" maxclientsperhost 1 "Sorry, only 1 connection per user is =

allowed" Group ftp User ftp AllowStoreRestart on ExtendedLog /var/log/ftp.log all=20 LogFormat default "%h %l %u %t \"%r\" %s %b" Umask 0000 TimeoutNoTransfer 600 TimeoutIdle 600 TimeoutStalled 600 AccessGrantMsg "User access granted for %u." <Global> <Limit LOGIN> AllowGroup ftp </Limit> DefaultRoot /home/ftp </Global> <Directory /*> AllowOverwrite off </Directory>

"Doesn't work" simply means that proftpd acts as if there were no RateReadBPS lines in the config file. To say in other words: the download xfer rate is the speed of the LAN independently of the login type (user or anonymous) and of the number behind the RateReadBPS directive. A very intersting thing: the download time is very high when running the proftpd with the "−n −d 5" switches. In this case the download xfer rate is constant: 0.71 kBytes/sec, and it's independent of the RateReadBPS setting, and the download is performed in the following way: one block is transmitted, then there is a 5−minute−wait−state, then the rest of the file is transmitted at the speed of the LAN, instead of waiting smaller amounts of time between blocks of transfers.

This is a basic ProFTPD configuration file (rename it to=20 # 'proftpd.conf' for actual use. It establishes a single server # and a single anonymous login. It assumes that you have a user/group # "nobody" and "ftp" for normal operation and anon. ServerName "ProFTPD Default Installation" ServerType inetd DefaultServer on # Port 21 is the standard FTP port. Port 21 # Umask 022 is a good standard umask to prevent new dirs and files # from being group and world writable. Umask 022 # To prevent DoS attacks, set the maximum number of child processes # to 30. If you need to allow more than 30 concurrent connections # at once, simply increase this value. Note that this ONLY works # in standalone mode, in inetd mode you should use an inetd server # that allows you to limit maximum number of processes per service # (such as xinetd) MaxInstances 32 MaxClientsPerHost 1 "The client number trying to connect from the same ho= st is limited to 1!" TimeoutStalled 600 # Set the user and group that the server normally runs at. User nobody Group nobody AuthUserFile /etc/passwd # Normally, we want files to be overwriteable. <Directory /*> AllowOverwrite on </Directory> # A basic anonymous configuration, no upload directories. <Anonymous ~ftp> User ftp Group ftp # We want clients to be able to login with "anonymous" as well as "ftp" UserAlias anonymous ftp # Limit the maximum number of anonymous logins MaxClients 4 RateReadBPS 8192 # We want 'welcome.msg' displayed at login, and '.message' displayed # in each newly chdired directory. DisplayLogin welcome.msg DisplayFirstChdir .message # Limit WRITE everywhere in the anonymous choot <Limit WRITE> DenyAll </Limit> </Anonymous>

I'm using RateReadBPS on a Redhat 6.0 system with pre10 RPMs and it works fine. I have kernel 2.2.14. here is a snippet of my config file that limits anonymous users to 12000 kbs: <Anonymous /somefilepath/somedir > User ftp Group ftp UserAlias anonymous ftp RateReadBPS 12000 RequireValidShell off MaxClients 15 <Limit LOGIN> AllowAll </Limit> <Limit WRITE> DenyAll </Limit> </Anonymous>

Does this limit each seperate anonymous user to 12000bytes D/L or all anonymous users as a whole?

> But I really think you should consider other points (security for example >), and upgrade AS SOON AS POSSIBLE. > > pre10 has MANY advantages over pre1, including built−in bandwidth > control, as you desire.

The reason i didnt uprgrade yet is the following: I am a very newbie to linux. I have SuSE 6.0 installed with very little packages selected (Almost only network things, because i only want to use the P100/16MB Ram as a FTP−Server in text mode without graphics) My problem is, that i'm not able to install proftp−opre10. I know that i have to install it with RPM. I downloaded core and inetd RPMs, but if want to install them, i get the following dependencies:

> But I really think you should consider other points (security for example >), and upgrade AS SOON AS POSSIBLE. > > pre10 has MANY advantages over pre1, including built−in bandwidth > control, as you desire.

The reason i didnt uprgrade yet is the following: I am a very newbie to linux. I have SuSE 6.0 installed with very little packages selected (Almost only network things, because i only want to use the P100/16MB Ram as a FTP−Server in text mode without graphics)

My problem is, that i'm not able to install proftp−opre10. I know that i have to install it with RPM. I downloaded core and inetd RPMs, but if want to install them, i get the following dependencies: pam >= 0.59 fileutils libpam.so.0 libc.so.6 (GLIBC_2.0) # I installed libc, didnt help libc.so.6 (GLIBC_2.1) libcrypt.so.1 (GLIBC_2.0) Yast installs the packages without any error messages, but then the programs wont run. Does anyone of you have mercy and tell me where i can find those packages ? Very big thanks in advance, Schinken

Quota controls

System Quota

The users on my Redhat 6.1 system have a disk quota set for management purposes. I would like proftpd to run /usr/bin/quota right after the welcome.msg displays and/or let the users run quota while logged into ftp. Anyone know of a way to do this? I noticed that proftpd returns an error after the user has exceded his hard quota, so maybe the quota can be displayed without the quota program?

'fraid that's not possible, ProFTPd doesn't allow any external programs to be run like "site <some command>" and I think this includes things like this 'pre−exec' (like Samba can do) also.

mod_quota

mod_quota is maintained in parallel to the main code tree, the most recent source is generally available from ftp://ftp.urbanrage.com/pub/c/mod_quota.c

I'm having a problem with proftpd & mod_quota. my configuration: Quotas On DefaultQuota 1048576 QuotaExempt 500,501 QuotaType hard QuotaCalc On The problem is that user 500 & 501 also have a 1MB Quota. Haw can I disable Quotas on certain users ? What am I'm doing wrong You aren't doing anything wrong, I just coded quota exempt wrong (it was exempting only during quota calc rather than also shutting off quotas when that user logs in).

This is my fault. I put in documentation on GroupQuota but I haven't finished the code that supports it. Do not use it yet, it is not implemented (only documented). I updated mod_quota.c to use open instead of fopen so that it could explicitly set the permissions on the file. It was brought to my attention that depending on your umask the .quota file could have permissions that could give access that you do not want to the .quota file.

Access controls

Access Prohibitions

Preventing general access to part or all of a site or blocking a particular user from reading some secure documents, these are all common requirements. All these are provided within the <Limit> directive, this permits access control based on

•

User

- Group
- Source IP
- Source Network (netmask or CIDR)

Limiting the network resources

Bandwidth limiting, this cannot be done on a per server or per VirtualHost basis, there are development plans to try and implement this but within the current architechure it is impossible.

Limiting is only possible at the moment on a per connection basis where the maximum download or upload rate can be specified, in addition to this a "grace" allocation can be given. The effect of this extra allocation is to effectively remove the throttling on small files which are downloaded within the allocation and only throttle larger files.

stuff

ratios, upload vs download limiting. mod_ratio. can interface with SQL to maintain state between sessions

Limit

(Problem #1 :) Currently I have the following: <Directory /home> <Limit LIST NLST> DenyAll </Limit> </Directory> <Directory /home/*/> <Limit LIST NLST> AllowAll </Limit> </Directory> What I mean it to do is permit /home to be listed in /, not permit the contents of /home to be listed, permit ppl to cd into the directories of /home and for people to be able to list the contents of the home dirs themselves. This ALMOST works except: 1. When in a home dir, NLST does not work. It's as if the second <Directory> never got read. Am I misusing the wildcard? I can't think of any other way of denying LIST and NLST access to just –ONE– directory level... 2. Doing something like: LIST –d ***** in /home circumvents the limit and permits listing of /home Main reason I have for blocking this is that /home contains 10,000+ dirs and listing this tends to suck CPU LOTS and ofcourse most people happen to use graphical browsers and then they get lost and so on... Problem #1.1: Actually, looking at the top list, LIST –d ***** has caused the server to spin out. It's currently sucking cpu like mad even though I've disconnected and quite the ftp client. This is obviously bad as it could make a system's load skyrocket to the point of unusability and therefore be a nice DoS attack. Problem #2 :) I have 1 user who cannot log in. We are using pam authentication and even if I just put pam_permit for the auth block it still denies him access. I've checked his shell (/bin/sh) and it's in /etc/shells. I can log in as him with telnet using a similar auth sequence as proftpd (pam). Can anyone think of what might be causing this/I might want to look for? I have no limits on logging in in the proftpd.conf. Just <Directory> based limits on reading and writing and listing. I run Proftpd 1.2pre7 in a glibc 2.1.1 system and have compiled it with gcc 2.95.1 on Linux 2.2.12. linux capabilities, pam and readme modules compiled in.

mod_ratio

To install, copy this file into modules/ and do: ./configure −−with−modules=mod_ratio This module is inactive unless configured, which can be done with an Anonymous, Directory, or VirtualHost block in proftpd.conf, or with a .ftpaccess file. (Ratios must be turned on elsewhere for a directive in .ftpaccess to take effect.) If compiled with −DMOD_MYSQL_RATIOS, this module can get and set session stats using mod_mysql, so the only directive needed is "Ratios on". This acts like a weak "UserRatio" −− any directive described below can override it. It also makes multiple concurrent uploads/downloads possible, with persistent credits. See mod_mysql docs for setup details. Most ratio directives take four numbers: file ratio, initial file credit, byte ratio, and initial byte credit. Setting either ratio to 0 disables that check: FooRatio bar [frate] [fcred] [brate] [bcred] The directives are HostRatio (matches FQDN −− wildcards are allowed in this one), AnonRatio (matches password entered in an anon login, usually an email address), UserRatio (accepts "*" for 'any user'), and GroupRatio. Matches are looked for in that order. Some examples: 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 FileRatioErrMsg "Come on you can send more files than that...." ByteRatioErrMsg "This file is %i big, you know...." LeechRatioMsg "Access: Unlimited" Setting "Ratios on" without configuring anything else will enable leech mode: it logs activity and sends status messages to the ftp client, but doesn't restrict traffic. Ratio module activity is recorded to syslog at DEBUG0; it usually shows up in /var/log/debug, like this: foo in /: CWD /teen :−15/3450 +0/0 (my 5 15 5 150000) =0/146550 [NO F] This example is for someone who (1) has downloaded 15 files totalling 3450k, (2) has uploaded nothing, (3) has a ratio of 5:1 files and 15:1 bytes, (4) has 0 files and 146k credit remaining, (5) got the ratio from the MySQL record ("my") and (6) is changing directory from / to /teen. Note that if this module is turned on globally, any user can create a personal ratio area with a .ftpaccess file. One way to prevent this is with: PathDenyFilter "\.ftpaccess$" The authors of this module, this ProFTPD software, and this OS and kernel disclaim all warranties and are not responsible for what random users of this module may do. If you have ideas on how to improve this module, please contact

---

Controlling permission changes

By default proftpd does not allow the client to change the system permissions of any file, this behaviour can be overridden using the AllowChmod directive. This directive is valid in the server, VirtualHost and Directory contexts allowing a fine grained control over the connecting clients.

.ftpaccess files

The .ftpaccess file is used to give a measure of control to the individual administrator of each VirtualHost without having to disturb the sysadmin everytime a change is required. Each .ftpaccess file can be considered to be a 'floating' section of the <Directory> configuration block.

The file uses the same structure as the main daemon configuration, though there are a significant number of directives which have no meaning within the .ftpacess context.

Example 14−4. .ftpaccess file

```
#
# Simple .ftpaccess file to control which IPs
# can access this directory structure
```

```
#
<Limit>
        Allow 212.32.5.0/26
        Allow 158.152.0.0/16
        DenyAll
</Limit>
#
# end
#
```

There is no way to disable the use of .ftpaccess files, however for servers with no shell access a simple file filter which blocks .ftpaccess should suffice to prevent users from using the functionality. There are also implications for letting untrained and inexperienced end users from generating their own files as they may lock themselves out of their space or leave it less secure than they intend.

Chapter 15. Debugging Problems

Originally by TJ Saunders

Users of ProFTPD will often encounters problems. It happens with all software, not just ProFTPD. How, then, does the user track down the cause of the problem, and fix it? This is the art of debugging. When users post these problems are the mailing lists, it is extremely helpful to include the following bits of information to help find the answer. Even better is when the user follows these steps and determines the solution for themselves.

Know the version

Various problems afflict various versions of the code, so when tracking down problems, it is good to know the version being used:

```
proftpd −vv
```

It is possible that the problem you are encountering is due to some bug that may already be fixed in a more current version, fixed in the CVS repository, or has a bug report with an attached patch. Searching http://bugs.proftpd.org will often yield useful information, depending on the keywords used in the search.

Know the modules

System administrators who compile and install the server from the souce code distribution will probably already know this, but administrators who install using RPMs or other package formats may not know the specifics of the contained pre−built binary. To list the modules compiled into the server:

```
proftpd −l
```

Knowing the modules helps to pinpoint the source of error messages (e.g. mod_tls and certificate files).

Perform syntax checks

When making changes to the configuration file, it is often helpful to make sure that your changes are valid. The easiest way to do this is to do an informative syntax check:

```
proftpd −td5
```

The −t option directs the server to parse the configuration file but stop before actually starting its operations as a server. The −d5 will cause the server to display debugging messages during this testing of the configuration file. Another useful command is:

```
proftpd −c /path/to/new/config/file −td5
```

which lets you test the syntax of some new configuration file before it is put into production.

Common problems

One common question is: "I changed the configuration file, but the new configuration is not being seen!" The solution depends on your configured ServerType. Almost certainly it will be standalone, as inetd−mode servers pick up configuration changes almost instantly (the server is started from the ground up for each connection). For configuration changes to be seen by a standalone server, you need to either stop, then start the server (the hard way), or to send the HUP signal the the daemon process.

Another common question involves use of ProFTPD's <Limit> directive to restrict certain FTP commands. These limits always function in addition to the normal filesystem permissions, not instead of them. If having problems writing, deleting, or updating files, check your directory and file permissions first.

Locate log files

A common response on the mailing lists to a posted question is: "What do your server logs say?" Locating the server's log files can be troublesome, depending on your configuration. If the SystemLog configuration directive is in effect, you know exactly where the server's log file is. If not, then by default the server uses syslog for logging. The location of syslog'd log files is set in your system's /etc/syslog.conf file. You may need to read your system's man pages for syslog.conf or syslogd to understand the format of that file. Note that the server will log using a syslog facility of daemon for most of its messages; during authentication, messages are logged using the authpriv facility.

Collect debug information

The code has built−in debug information reporting capabilities − the trick is in enabling that reporting, and tracking down where it is sent. The easiest way to get the debugging information is to start the server from the command line using:

```
proftpd −nd5
```

This will display lots of information on the connected terminal's screen, both as the server starts up and during the servicing of any clients. If clients are having trouble logging in or authenticating, the debug messages reported by the server when a client connects are much more useful than knowing the messages displayed by the client, as the client does not know why it cannot log in. If asked to send debugging information to the mailing list, you can send the relevant snippets (if you know what the relevant debug messages are), or you can capture the debug output to a file:

```
proftpd −nd5 2>&1 >& /path/to/debug/file
```

and send that file, compressed, along with your post.

The above method works if you have ServerType standalone in your configuration file. If you run the server in inetd mode instead, and are unable or unwilling to make the changes necessary to run the server in standalone mode for the sake of debugging, then use of the SystemLog configuration directive is necessary for capturing the debug information. Add this directive to your configuration file, and add −d5 to your /etc/inetd.conf's ftp line, or to the server_args tag in your xinetd configuration file for the server. Be sure to restart inetd/xinetd so that your configuration changes will take effect.

Note that use of the SystemLog directive is not necessarily confined to inetd mode servers. If you are interested in letting your standalone server run unattended and want to have that debugging information in the log file, use SystemLog and add −d5 (or whatever your preferred debug level is) to the server startup script.

Once you have the debug output and various other information, and are still in need of help, search the FAQ, Userguide, and mailing list archives for material related to the problem. If you're unable to find anything helpful in these sources, post your question to the appropriate mailing list. Be sure to include the version used, your proftpd.conf, and possibly any debug information.

The following document describes how to ask good questions that are likely to be answered:

```
http://www.tuxedo.org/~esr/faqs/smart-questions.html
```

Chapter 16. Common Problems

"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.

"bind: unable to bind to port"

Proftpd is unable to take control of port 21 (or whatever port has been defined) at startup. This is normally down to one of two reasons.

Clash with inetd

Proftpd has been configured to in standalone mode but inetd hasn't been reconfigured to not provide the ftp service. Ensure that the line starting with "ftp" has been removed or commented out and send a SIGHUP to the inetd process.

Another copy of Proftpd is running

Check the process listing for the server to ensure that another proftpd process is not already running. If it is either kill this process or send the master daemon a SIGHUP to make it reload it's configuration if desired.

"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
```

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)

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−07.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.

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.

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.

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>

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.

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

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.

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.

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:

Example 16−1. xinetd configuration

```
service ftp
{
        flags           = REUSE
        socket_type     = stream
        instances       = 50
        wait            = no
        user            = root
        server          = /usr/sbin/proftpd
        bind            = <the-ip-you-wish-to-bind-to>
        log_on_success  = HOST PID
        log_on_failure  = HOST RECORD
}
```

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.

Is is possible to shutdown a single VirtualHost?

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

Unable to resolve IP

This is not normally a problem with proftpd, but with the configuration of the hosting server or the DNS servers it relies on. The error message simply means the daemon was unable to map the hostname given in the configuration to an IP address. Check the /etc/resolv.conf (or equivalent) and the spelling of the hostname in the proftpd.conf file, also try tools such as nslookup to check the behaviour of the local DNS servers.

Chapter 17. More complex Configuration Issues

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

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.

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.

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>

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.

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).

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.

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> ...

Chapter 18. Running ProFTPD As A Nonroot User

Chapter by TJ Saunders

Occasionally, one might want to run ProFTPD on a system where root privs are not available to you as a user. It is still possible to setup a functioning FTP server without root privileges. There are a few catches and special considerations for this, however.

Here are the configuration directives that you will need to use in order to run the server without root privileges:

Port
: This needs to be a number greater than 1023. Lower number ports require root privileges in order for the process to bind to that address. This will also mean that clients wishing to contact your server will need to know the port on which it is listening. Most FTP clients connect to the standard FTP port (21).

AuthUserFile, AuthGroupFile
: In order to authenticate users, by default the server looks in /etc/passwd for account information, and in /etc/shadow for the password. Comparing stored passwords requires root privileges, which this nonroot–running daemon will not have. You can get around this requirement by supplying your own passwd (and possibly group) files via the AuthUserFile and AuthGroupFile directives. Make sure the permissions on your custom files allow for the daemon to read them (but hopefully not other users).

AuthPAM
: PAM authentication requires root privileges. This directive will need to be set off.

WtmpLog
: Logging to wtmp files requires root privileges. While it is not strictly necessary for this directive to be set to off, failure to do so will result in server log messages like:

```
host.domain.net (localhost[127.0.0.1]) - wtmpx /var/adm/wtmpx: Permission denied
```

User, Group
: The ability to switch the identity of the server process to those configured by the User and Group directives requires, of course, root privileges. It is best to configure User to be your username, and Group to be the name of your primary group (which is usually the first group listed by the groups command).

Note that other configuration directives will be affected by the lack of root privileges: DefaultRoot will not work, nor will <Anonymous> sections, nor UserOwner. Basically any operation that requires root privileges will be disabled.

IV. WorkShop

Table of Contents

This is primarily cut and paste from the mailing list to get the text into my working files. Once this is done I can look at starting to clean up the information into the user guide proper

Chapter 19. Cleaned sections

Cleaned − part A

Filtering upload/download paths

There are occasions when it is desirable or essential that access to certain files or paths is limited, or that steps are taken to prevent uploading of certain kinds of material. The most common method of achieving these ends is to use the PathAllowFilter and PathDenyFilter directives.

Example 19−1. Filter example

```
#
# Block alteration of .ftpaccess
# Prevent uploading of mp3 files.
#
PathDenyFilter "(^\.ftpaccess$)|(\.mp3$)"
```

File overwriting

The default configuration of the daemon prevents the overwriting of files on the server. To disable this behaviour set "AllowOverwrite 1"

Logs report 'signal 11'

If 'ProFTPD terminating (signal 11) appears in your logs it's an indication that there is a serious problem with your insallation. A signal 11 (or SEGV) is a segmentation fault, usually caused by either incompatible libraries or a bug in the daemon. If recompiling from a clean source distribution doesn't resolve the problem it's probably worth reporting it as a bug.

Unknown group errors

A very simple problem, with an equally simple solution. Proftpd requires that a user and group are specified for it to run the daemon as after a sucessful login. These names are resolved to their numeric values by the appropriate system calls when the configuration is loaded or tested (using the −t option). Failure of these to resolve is a non−recoverable solution and is almost always caused by the group not existing in the appropriate user directory (ie /etc/passwd or /etc/shadow).

The solution is to either create the user/group account or to reconfigure Proftpd to use another user/group account. Which of these is the best solution will depend on your local conditions. The user Proftpd runs as does not require a valid password or a usable shell (/bin/true will suffice).

---

proftpd.filter

Hi, there: I tried to setup proftpd.conf with allowFilter in the proftpd.conf #====== ServerName mumble ServerType inetd DeferWelcome on Umask 002 User proftpd Group proftpd TransferLog

/var/log/proftpd/xferlog.log DefaultRoot ~ users,!staff TimeoutLogin 120 TimeoutIdle 600 TimeoutNoTransfer 900 TimeoutStalled 3600 ScoreboardPath /var/run/proftpd LogFormat default "%h %l %u %t\"%r\" %s %b" LogFormat auth "%v[%P] %h %t \"%r\" %s" LogFormat write "%h %l %u %t \"%r\" %s %b" UseReverseDNS off AllowFilter ".*/[a−zA−Z0−9]+$" #========== When I tried to login at username after I hit enter, I got "Forbidden command argument". I am using 1.2.0pre9. Any ideas, After I marked AllowFilter out, everything is fine. I tried to use Allowfilter too but couldn't get it to work no matter what I tried. I ended up adding a denyfilter "%" instead. −−−−Original Message Follows−−−− From: "michael liu" <mliu@rmsys.net> Reply−To: proftpd@proftpd.org To: "Proftpd@Proftpd. Net" <proftpd@proftpd.org> Subject: [ProFTPD] allowFilter Date: Fri, 7 Apr 2000 17:40:29 −0500 Hi, there: I tried to setup proftpd.conf with allowFilter in the proftpd.conf #====== ServerName mumble ServerType inetd DeferWelcome on Umask 002 User proftpd Group proftpd TransferLog /var/log/proftpd/xferlog.log DefaultRoot ~ users,!staff TimeoutLogin 120 TimeoutIdle 600 TimeoutNoTransfer 900 TimeoutStalled 3600 ScoreboardPath /var/run/proftpd LogFormat default "%h %l %u %t\"%r\" %s %b" LogFormat auth "%v[%P] %h %t \"%r\" %s" LogFormat write "%h %l %u %t \"%r\" %s %b" UseReverseDNS off AllowFilter ".*/[a−zA−Z0−9]+$" #========== When I tried to login at username after I hit enter, I got "Forbidden command argument". I am using 1.2.0pre9. Any ideas, After I marked AllowFilter out, everything is fine. Anyone know a good string for AllowFilter? I tried the one in the docs on www.proftpd.org but then every command is invalid. i'm using 1.2.0pre8 on redhat 6.1. i have sucessfully used the PathDenyFilter in the <Global> section with the example: PathDenyFilter "(\.ftpaccess)|(\.htaccess)$" now i am trying to limit commands with DenyFilter. i admit to not understanding regular expressions, but using the above as a sort of guide, i am still baffled. i've tried the following variations without success: DenyFilter "proxy" DenyFilter proxy DenyFilter "(proxy)|(pwd)$" DenyFilter "proxy$" DenyFilter "(proxy)$" if anyone could shed some light on this i would be very pleased. 1. proftpd−1.2p10 working OK. 2. Can I hidde on virtual root ftp servers FrontPage directory _vti_* ? When I add in proftpd.conf directive: PathDenyFilter "(\.htaccess)|(\.ftpaccess)$" working OK. but add: PATHDenyFilter "(\.htaccess)|(\.ftpaccess)|(_vti_*)$" directory _vti_* not hidding. == Wiesiek Glod e−mail: wkg@x2.pl old wkg@halicz.com.pl

Chapter 20. Initial ponderings from the list

stuff_a

showing all files

How can i can show files starting with a "." ? Surprisingly easy method: LsDefaultOptions "−a"

Setting defaults for all VirtualHosts

Many of the directives which are valid in the VirtualHost context are also valid in the <Global> context. Proftpd will take the <Global> values as the default but will allow them to be overridden on a <VirtualHost> by <VirtualHost> basis

Data connection problems

Why I get "Can't build data connection : Connection refused" error when I send the list or dir commands to proftpd from Windows Command prompt? normally down to firewalls and pasv/acttive connections > Does someone knows how to set up proftpd as a passive ftp server? I'm having trouble +with proftp behind a firewall, upload speeds are very low. > Could the problem be that it's running active instead of passive?? It is not the job of the server to configure this. It is the client's job. Set your FTP client to use PASV.

Installation

History: I need FTP server on my Linux machine. Reading about it on many Linux sites i found recommedations that proftpd is more secure than wuftpd, that configuration is easier etc etc. OK, let's listen to more expireanced linux admins and install proftpd. I found site (good one), read all i could and decided that's the one, let's install it.

1. No installation instructions. OK, never mind, it must be peace of cake on RH6.1 with RPM's. Let's find them. ftp://ftp.proftpd.org/pub/proftpd/RPMS/

2. There are 2 directories on above address: i386 and i686. OK, i386 must be the one for my poor Pentium 133, the other one must be for PII machines. Let's use i386 one.

3. 6 files in i386 directory, ...core... ...inetd... and ...standalone... for versions 9 and 10 OK, I want standalone version 10 so I definatelly need that small 4,21 file ... and I presume that core file for version 10 is needed as well ...

Too many maybe's. I am not far from giving up (wuftpd). Was it SO hard to write few sentences describing installation issues for this case scenario??????

Also, there is no mailing list archive, so I am posting this question after reading faq and user guide.

Please could someone decribe me installation procedure for RH6.1 (RMP) v10 (reply to this message via e−mail)? You need the main package: proftpd−core−1.2.0pre10−1.i386.rpm Plus you need *either* the −inetd or −standalone package, depending upon whether you want to start proftpd through inetd or have it run as a standalone daemon.

Jim P.S. I encountered a problem with 1.2.0pre10 not recognizing secondary group permissions, but skimming through the mailing list archives uncovered a patch from MacGuyver that fixed it. http://www.proftpd.org/proftpd–l–archive/00–01/msg00371.html I included the patch in a revised RPM if you are interested in trying it. http://hammer.prohosting.com/~onjapan/rpms/ | 1. No installation instructions. Doesn't the INSTALL file count? | Also, there is no mailing list archive, so I am posting this question after | reading faq and user guide. Actually, there are two in addition to the bugzilla system: http://www.proftpd.org/proftpd–l–archive/ http://www.proftpd.org/proftpd–devel–archive/ | Please could someone decribe me installation procedure for RH6.1 (RMP) v10 The RPM spec file is in contrib/dist/rpm/. All that said, I'm sure there is room for improvement.

---

uploading issues

how can i make two diferent users were the one can only upload in upload_here and the other can do everything in directory ftp? What I have so far: User nobody Group nogroup DefaultRoot /ftp {Directory /ftp} {Limit WRITE} Deny All {/Limit} {/Directory} {Directory /ftp/upload_here} {Limit CWD STOR RETR MKD RMD XCWD XMKD XRMD} Allow All {/Limit} {/Directory}

---

proftpd.binding

Using the default "basic.conf" file with a 1.2.0pre10 installation on Slackware Linux 7 (2.2.14), proftpd has problems with all of the aliased interfaces on the box. The daemon will start up and listen on port 21 on all of the interfaces, however if I connect to any of the aliased interfaces, the log shows a bind(): Permission Denied and the connection closes. The primary and localhost interface will accept connections with no problems. The daemon is running as nobody:nogroup, and those users exist on the system. Incedintally, if I run the daemon as root, proftpd works fine on ALL configured interfaces, however running the daemon as root is not desired. I have had no problems running any versions previous to pre10 on the same machine and on other machines as well. Anyone knows why am I getting this error?? Mar 13 16:44:40 gabriel proftpd[2997]: gabriel.domain.com – attempted bind to 0.0.0.0, port 21 Mar 13 16:44:40 gabriel proftpd[2997]: gabriel.domain.com – bind() failed in inet_create_connection(): Address already in use Mar 13 16:44:40 gabriel proftpd[2997]: gabriel.domain.com – Check the Serv erType directive to ensure you are configured correctly. I comment out the ftp in my inetd.conf. I am using the proftpd as a services On Fri, 17 Dec 1999, Alderman, Sean wrote: > Just compiled and installed pre9 w/ the TYPE A N patch... Running in > standalone mode works great...Running from inetd gives the following in > /var/log/messages – > Dec 17 15:22:51 dmz1 proftpd[7163]: dmz1.freitrater.com – bind() failed in > inet_create_connection(): Address already in use Normally caused by one of the following o Another proftpd running o ftp being configured in inetd and running the daemon in standalone o Something else listening on port 21

See the SocketBindTight directive. When off (the default), sockets are bound to 0.0.0.0:port. When on, specific IPs are used (as specified by VirtualHost) and the "main" IP is guessed. You can also use the port 0 trick to prevent binding the the main guessed IP.

---

proftpd.auth

On Mon, Apr 03, 2000 at 06:51:49PM –0700, Irwan Hadi wrote: > Is it possible to make proftpd use it's own username + password and not use > I don't want use the system account because the more user in /etc/passwd > (system account) the more the system can be compromised. AuthUserFile DefaultRoot or SQL/LDAP authentication DefaultRoot At 14:01 06/04/2000 +0100, **hamster@vom.tm**, has written a message, and here is the reply : >On Mon, Apr 03, 2000 at 06:51:49PM –0700, Irwan Hadi wrote: >AuthUserFile >DefaultRoot > >or > >SQL/LDAP authentication >DefaultRoot Thanks fpr you reply, but if there is already somebody here who has done similar like what I like, I hope that you can give the steps to me. Because I'm in a hurry to setup the ftp server as the deadline for it is next week.

In the basic /usr/local/etc/proftpd.conf you will need to add this line: AuthUserFile <File path> Where <File Path> is the pathname of the file to use instead of /etc/passwd. Note: the auth file has to have the same format as /etc/passwd. More info: http://www.proftpd.org/docs/configuration.html#AuthUserFile You will probably also want to use: AuthGroupFile <File Path> Much the same, format is the same as the /etc/group file. More Info: http://www.proftpd.org/docs/configuration.html#AuthGroupFile In the basic configuration file, you may want to comment out the anonymous entry... That is the easiest way to do it... You can also use Ldap, MySQL, but neither are for people in a rush. :−) Information on the alternatives can be found in: http://www.proftpd.org/docs/proftpdfaq−8.html At 12:00 06/04/2000 −0400, Michael Grabenstein wrote: > > In the basic /usr/local/etc/proftpd.conf you will need to add >this line: > >AuthUserFile <File path> > > Where <File Path> is the pathname of the file to use instead of >/etc/passwd. > >Note: the auth file has to have the same format as /etc/passwd. > >More info: >http://www.proftpd.org/docs/configuration.html#AuthUserFile First of all I want to thank you for your reply, but my question is what is the meaning of "the same format ?" so I make a list of username:password::::/homedir/ how about the password ? can it be encrypted or not ? if it *can* be encrypted, with which tool should I encrypt it then.

Irwan Hadi wrote: > First of all I want to thank you for your reply, but my question is what is > the meaning of "the same format ?" > so I make a list of > username:password::::/homedir/ > how about the password ? can it be encrypted or not ? > if it *can* be encrypted, with which tool should I encrypt it then. > Yes that would be the format... I use Perl to encrypt the password, or if you already have a /etc/passwd to start with, then just copy it... An alternative easy way to do this is to encrypt a password and keep the encrypted version around. Like Change your password to 'ABC123' then as you create users in the alternate passwd file, paste the encrypted form of that password into the new logon entry. And instruct the new user to change their password as soon as they first FTP to the system, or change it for them via FTP and give them the new password. :−) BTW: once you have the encrypted version of 'ABC123' feel free to change your password back. :−) Attached is a simple Perl script that will encrypt a plain text password sent to it... Mark, please feel free to add this to the FAQ. TIA. I don't believe proftp has a way of using plain text passwords in the password file, but Mark can correct me if I am wrong. :−)

#−−− Start Cut after this line #!/usr/bin/perl use Getopt::Std; use vars qw($opt_h $opt_p $opt_s); getopt ("hp:s:"); my ($salt); if ((defined($opt_h)) || (! defined($opt_p))) { print "Usage: $0 −hps\n"; print "\t−h −− This Usage message\n"; print "\t−p <password> −− The password to encrypt\n"; print "\t−s <salt> −− The salt to use, optional\n\n"; exit (166); } if ($opt_s =~ /(\w+)/) { $salt = $1; } else { $chr = chr(int(rand(26)+65)); $salt = $chr; $chr = chr(int(rand(26)+97)); $salt .= $chr; } print crypt($opt_p, $salt) . "\n"; exit (0); # −− Stop here. Don't get the signature at the bottom...

> First of all I want to thank you for your reply, but my question is what is > the meaning of "the same format ?" > so I make a list of > username:password::::/homedir/ > how about the password ? can it be encrypted or not ? > if it *can* be encrypted, with which tool should I encrypt it then. > Yes that would be the format... I use Perl to encrypt the password, or if you already have a /etc/passwd to start with, then just copy it... An alternative easy way to do this is to encrypt a password and keep the encrypted version around. Like Change your password to 'ABC123' then as you create users in the alternate passwd file, paste the encrypted form of that password into the new logon entry. And instruct the new user to change their password as soon as they first FTP to the system, or change it for them via FTP and give them the new password. :−) BTW: once you have the encrypted version of 'ABC123' feel free to change your password back. :−) Attached is a simple Perl script that will encrypt a plain text password sent to it... Mark, please feel free to add this to the FAQ. TIA. I don't believe proftp has a way of using plain text passwords in the password file, but Mark can correct me if I am wrong. :−)

#−−− Start Cut after this line #!/usr/bin/perl use Getopt::Std; use vars qw($opt_h $opt_p $opt_s); getopt ("hp:s:"); my ($salt); if ((defined($opt_h)) || (! defined($opt_p))) { print "Usage: $0 −hps\n"; print "\t−h −− This Usage message\n"; print "\t−p <password> −− The password to encrypt\n"; print "\t−s <salt> −− The salt to use, optional\n\n"; exit (166); } if ($opt_s =~ /(\w+)/) { $salt = $1; } else { $chr = chr(int(rand(26)+65)); $salt = $chr; $chr = chr(int(rand(26)+97)); $salt .= $chr; } print crypt($opt_p, $salt) . "\n"; exit (0); # −− Stop here. Don't get the signature at the bottom...

At 09:29 07/04/2000 −0400, **Michael Grabenstein**, has written a message, and here is the reply : >Irwan Hadi wrote: > I use Perl to encrypt the password, or if you already have a /etc/passwd to >start with, then just copy it... > Attached is a simple Perl script that will encrypt a plain text password >sent >to it... Umm, sorry to bother you again, but how about the shell of the users ? should it be set to /bin/bash or /bin/ftponly (which is another name of /bin/false) ? On Tue, Apr 04, 2000 at 04:15:34PM −0700, Irwan Hadi wrote: > Umm, sorry to bother you again, but how about the shell of the users ? > should it be set to /bin/bash or /bin/ftponly (which is another name of > /bin/false) ? The shell can be whatever you want, however it has to be in either /etc/shells or the RequireValidShell directive has to be set to "off" On Tue, Apr 04, 2000 at 09:17:07AM −0700, Irwan Hadi wrote: > username:password::::/homedir/ > how about the password ? can it be encrypted or not ? > if it *can* be encrypted, with which tool should I encrypt it then. Is must be crypted, there is a script in the contrib directory (genuser.pl IIRC) to do this.

Hello .. Just to clarify something for me ..... If I use mySQL (with mySQL users) as the authentication method for users of my proftpd server, then I will not need to add them as users in the system password file. I want all users to this server to have to login, but I'd prefer not to have to add them to the password file. Am I way off base on this ? or ?? Nope, sounds pretty correct. You do have to add a couple of things to the /etc/ group and passwd file, but not all of the users. In /etc/passwd, you need to add a user for the user that proftpd will run under (or use nobody...). In /etc/group, you need to add a group for the user that will run proftpd (or use nobody, again...). Plus in /etc/group, you need the group you will be assigning to all the users on the system. (or list of groups...) Then in MySQL's user table you need to have an entry for the user you will be connecting with. The line in proftpd.conf: MySQLInfo localhost hamster ABC123 proftpd Means you need a user in the user table of the mysql DB for user id 'hamster' with password 'ABC123'. Also from the line above the DB name is proftpd... My user's don't have upload, so everything looks good. This only gets sticky if you want your users to upload... When they upload the files are assigned the user id number assigned in MySQL, but if that does not exist in /etc/passwd then 'ls' shows only the uid number. If you make the uid number the same as a user that exists in the /etc/passwd, then it looks normal with the added benefit of that user owning the file. :−) You could have a "generic" user in the /etc/passwd that can not log in and have all MySQL user id's assigned that uid. The home directory comes form MySQL, so they can all have different homes with no problems... Greetings: I am trying to get 1.20pre10 running on Solaris 7, and, using the basic configuration file shipped, can only log in as anonymous (or ftp) but never an actual user of the system. I have a shadow file (o' course) and compiled with −−enable−shadow and −−enable−shadow−autodetect options .. the only changes to the basic config file were in using inetd and, well, allowoverwrite off. I have since added a preemptive, if unnecessary, <limit login> allowall </limit> and remove the anonymous block (now I have no ftp access, duh!). Any ideas about what I am missing? I have several users on my Linux system. I am trying to allow them all to be able to have logins for FTP. For example, one customer can create the FTP account webmaster which logs into /home/customer1/public_html with the password poiuy, while another customer can create the FTP account webmaster which logs into /home/customer2/public_html with the password lkjhg. I looked through the configurations and AuthUserFile looked like the best way to do it. So I setup a test one. In the proftpd.conf vhost for game−guys.com, I setup AuthUserFile /home/game−guys/game−guys. In that game−guys, I would like to have several logins and passwords (encrypted of course) which can only login to game−guys.com on the server. My question is, what should go in /home/game−guys/game−guys, and how should I add users to it and set the password? All three commands, useradd, passwd and htpasswd don't seem to want to work properly. Does anyone have any ideas? Thanks, help would be appreciated. :)

On Wed, Mar 22, 2000 at 09:35:15AM −0500, Alderman, Sean wrote: > You might want to check the archives. I believe someone had built a perl > script and posted it to the list to create encrypted username/password pairs > for custom proftpd auth files. genuser.pl in the contrib/ directory. Syntax is "htpasswd.pl userid password". Output is "userid:encryptedPassword". You might need to change the path to your perl. #!/usr/bin/perl $user = $ARGV[0]; $pass1 = $ARGV[1]; my($salt)=seedchar().seedchar(); $pass = crypt($pass1, $salt); print STDOUT "$user:$pass\n"; sub seedchar { ('a'..'z','A'..'Z','0'..'9','.','/','"')[rand(64)]; } > Syntax is "htpasswd.pl userid password". Output is > "userid:encryptedPassword". You might need to change the path to your perl. Well, I tried using htpasswd, but that does not go to the same format as /etc/passwd. ProFTPD will only read the /etc/passwd format, correct?

I was wondering, what utility do you use to generate the encrypt shadow passwd??

I have several users on my Linux system. I am trying to allow them all to be able to have logins for FTP. For example, one customer can create the FTP account webmaster which logs into /home/customer1/public_html with the password poiuy, while another customer can create the FTP account webmaster which logs into /home/customer2/public_html with the password lkjhg. I looked through the configurations and AuthUserFile looked like the best way to do it. So I setup a test one. In the proftpd.conf vhost for game−guys.com, I setup AuthUserFile /home/game−guys/game−guys. In that game−guys, I would like to have several logins and passwords (encrypted of course) which can only login to game−guys.com on the server. My question is, what should go in /home/game−guys/game−guys, and how should I add users to it and set the password? All three commands, useradd, passwd and htpasswd don't seem to want to work properly. Does anyone have any ideas? Thanks, help would be appreciated. :) On Wed, Mar 22, 2000 at 09:35:15AM −0500, Alderman, Sean wrote: > You might want to check the archives. I believe someone had built a perl > script and posted it to the list to create encrypted username/password pairs > for custom proftpd auth files. genuser.pl in the contrib/ directory. > genuser.pl in the contrib/ directory. > > Mark

Okay, I ran genuser with ftp1 as my username and lala as my password. It came up as this: ftp1:9l/MJ4vLeAAlU So everything after that colon can be put in the passwd file, and it will work? Thanks for all of your help! On Thu, Mar 23, 2000 at 06:09:56PM −0500, Vincent Paglione wrote: > ftp1:9l/MJ4vLeAAlU > > So everything after that colon can be put in the passwd file, and it will > work? Thanks for all of your help! What you need to do from this point is generate a /etc/passwd compatible file ie.
ftp1:9l/MJ4vLeAAlU:103:65534::/var/run/identd:/bin/false
ftp2:9l/MJ4vLeAAlU:103:65534::/var/run/identd:/bin/false
ftp3:9l/MJ4vLeAAlU:103:65534::/var/run/identd:/bin/false
ftp4:9l/MJ4vLeAAlU:103:65534::/var/run/identd:/bin/false and save this as your $CONF/authpasswdfile and then reference it from the proftpd.conf I've got my own homebrew system running on the core ftp vhost server which takes a condensed version of the proftpd.conf and builds it into the full configuration and generates the passwd/group files. I'll toss it up there if anyone is interested (but it's nasty evil perl with no documentation :)

Okay, I have everything with my AuthUserFile setup. THANKS everyone who helped me. I just have one more request. In /etc/passwd, if I wanted to make additional FTP accounts for a user, I would make the UID the same as the original account so that the sub−ftp account could write/overwrite the data in the main accounts directory, and once it was uploaded, the main account could write/overwrite it too. Do you know how I can accomplis this with multiple passwd's?

I was using Fetch 3.0.03 (MacOS) to transfer 10's of thousands of files (over 1GB total data) and about half way through I received: Mar 25 01:48:15 sneex proftpd[539]: Internal error: non−PASV mode, yet data connection already exists?!? Anyone seen this or have comments?

On Sat, 25 Mar 2000, Vincent Paglione wrote: >Okay, I have everything with my AuthUserFile setup. THANKS everyone who >helped me. I just have one more request. > >In /etc/passwd, if I wanted to make additional FTP accounts for a user, I >would make the UID the same as the original account so that the sub−ftp >account could write/overwrite the data in the main accounts directory, and >once it was uploaded, the main account could write/overwrite it too. > >Do you know how I can accomplis this with multiple passwd's? >From the sound of things what you want to do is create a group, say fnord, and make all of the relevant users have fnord as their primary group then play with umask to give everyone the requisite access. This is a far tidier solution than creating multiple accounts with the same UID, which while technically possible is messy. Have a play with groupadd(8), addgroup(8), and group(5) and see how you go.

> and save this as your $CONF/authpasswdfile and then reference it from > the proftpd.conf This is the only part I did not udnerstand. I was hoping to save the passwd file somewhere like /etc/users/userpasswd. What is this $CONF/authpasswdfile?

I noticed that there may be a bug in using AuthUserFile. When you create a new passwd file on FreeBSD 3.4, it only reads the first 3 lines of the passwd file. Any user that is after the 3rd line is not read, and proftpd says that user is not found. Anybody have any idea.

if you don't want to have PAM−support, try to compile without PAM, otherwise compile with PAM. Configure looks like configure −−with−modules=mod_pam if you want to have PAM−Support, or −−without−modules=mod_pam (?) if you don't want to have support for PAM.

tstoev@compsci.lyon.edu on 09.02.2000 06:42:18 Bitte antworten an proftpd@proftpd.org @ Internet An: proftpd@proftpd.org @ Internet Kopie: Thema: [ProFTPD] AuthPAMAuthoritative I have tried to use the AuthPAMAuthoritative directive and it does not seem to work, because it seems like PAM is always the authority. That is on FreeBSD 3.4 and RedHat 6.0. Does anybody have an idea.

I have tried to use the AuthPAMAuthoritative directive and it does not seem to work, because it seems like PAM is always the authority. That is on FreeBSD 3.4 and RedHat 6.0. Does anybody have an idea.

I have a question, i am Using a special AuthUserFile which i think is = correctly created! (username:crypt(password,salt)) But when i try to login with a user, given in this AuthUserFile, it = doesn't work. I have already added the Directive=20 RequireValidShell off but it does not work, what can i do?? is there a way to find the mistake = ??

I wish to only have FTP access to to "fake", non shell users, since my shell users login with ssh, and they cannot use the same username password pair in an unencrypted FTP session. The server running FTP only has a single IP and will only be listening in on PORT 21, so there won't be any virtual FTP hosts. ProFTPd is configured in as a standalone daemon, no inetd.

To that end, I have created an alternative passwd file, using the apache htpasswd command, and a group file. ProFTPd is configured to run as user nobody, and does a chroot for to the www root directory which it owns. Just for testing purposes, I have made these alternative passwd & group files, plus the directories they are in, readable by all user ids.

I have added the following directives to proftpd.conf: AuthUserFile /opt/proftpd/etc/passwd AuthGroupFile /opt/proftpd/etc/group PersistentPasswd off As mentioned, I only want proftpd to use /opt/proftpd/etc/passwd and *NOT* the server's /etc/passwd file. Unfortunately, when I use this configuration, no one can log in. Reading the FAQ, I try to add the directive: AuthPAMAuthoritative off Unfortunately when I do so, I get the following error when I start up ProFTPd: − Fatal: unknown configuration directive 'AuthPAMAuthoritative'. Running "proftpd −l" to get a list of modules reveals: mod_core.c mod_auth.c mod_xfer.c mod_site.c mod_ls.c mod_unixpw.c mod_log.c

Unfortunately the AuthPAMAuthoritative directive is *ONLY* read by the "mod_pam" module, which is missing. So when I try to recompile ProFTPd, with the configure "−−with−modules=mod_pam" option, I get the following compiler error when I run gmake: mod_pam.c:39: security/pam_appl.h: No such file or directory

No "pam_appl.h" file is included with ProFTPd, and it is not included in "/usr/include/security". (I am running NetBSD 1.4.1 on ix86 and sparc, neither of which have anything related to PAMs. No pam_appl.h, pam.conf, or pam_unix.so files. "apropos pam" finds nothing appropriate.)

What can I do? I simple want ProFTPd to use an alternative passwd and group file, just like my apache does. I have went through all of the ProFTPd documentation, FAQ, and mailing list archive without any solution.

On Mon, Jan 31, 2000 at 07:29:13AM −0500, Alicia da Conceicao wrote: > I wish to only have FTP access to to "fake", non shell users, since > my shell users login with ssh, and they cannot use the same username > password pair in an unencrypted FTP session. The server running FTP [...] > I have added the following directives to proftpd.conf: > > AuthUserFile /opt/proftpd/etc/passwd > AuthGroupFile /opt/proftpd/etc/group > PersistentPasswd off [...] > this configuration, no one can log in. Reading the FAQ, I try to > add the directive: > > AuthPAMAuthoritative off > > Unfortunately when I do so, I get the following error when I start > up ProFTPd: > > − Fatal: unknown configuration directive > 'AuthPAMAuthoritative'. [..] > with the configure "−−with−modules=mod_pam" option, I get the following > compiler error when I run gmake: > > mod_pam.c:39: security/pam_appl.h: No such file or directory Given that you don't appear to have PAM installed on your machine you don't need to concern yourself with the "AuthPAMAuthoritative" directive.

>> I have added the following directives to proftpd.conf: >> AuthUserFile /opt/proftpd/etc/passwd >> AuthGroupFile /opt/proftpd/etc/group >> PersistentPasswd off >> ... >> mod_pam.c:39: security/pam_appl.h: No such file or directory > > Given that you don't appear to have PAM installed on your machine you > don't need to concern yourself with the "AuthPAMAuthoritative" > directive. Dear Mark: If that is the case, then why doesn't the AuthUserFile work? No one can login using the alternative passwd and group files I created with apache htpasswd. I assumed that AuthPAMAuthoritative might be the cause of the problem, since the FAQ mentioned it. My goal is to restrict FTP access to users who do not have entries in the server /etc/passwd file. All FTP users must be specified in /opt/proftpd/etc/passwd. For security reasons, users with shell access will be *NOT* be allowed to use FTP (they can use ssh/scp instead). Am I doing any thing work?

I have some problems with 'AuthUserFile' / 'AuthGroupFile'. I set them to an absolute path but I cannot login. I created my own passwd with the following line: userxyz:x:501:101:Webadmin:/var/http/userxyz:/bin/bash and my own group file: wwwuser:x:101: What about /etc/shadow? A test with an own passwd (with the crypted password in it) of userxyz:fsdf76s23:501:101:Webadmin:/var/http/userxyz:/bin/bash didn't work, too... I am using SuSE Linux 6.3 on x86.

On Fri, Jan 28, 2000 at 04:38:26PM +0100, Chris Loos wrote: > Hi, > I have some problems with 'AuthUserFile' / 'AuthGroupFile'. > I set them to an absolute path but I cannot login. > I created my own passwd with the following line: > userxyz:x:501:101:Webadmin:/var/http/userxyz:/bin/bash > and my own group file: > wwwuser:x:101: > What about /etc/shadow? > A test with an own passwd (with the crypted password in it) of > userxyz:fsdf76s23:501:101:Webadmin:/var/http/userxyz:/bin/bash > didn't work, too... Check the FAQ.... AuthPAMAuthoritive off (check the spelling of the directive) PersistantPasswd off (IIRC)

of course I checked the FAQs but the only hint I found was theses two comments you wrote. But after using "AuthPamAuthoritve off" and "PersistantPasswd off" inetd isn't able to start proftpd − seems that the ftpd crashed or stops itself immediately.

Weird, I'm using AuthUserFile extensively on one machine (virtualhosting and I want the user/password details to be unique to the virtual) with no problems. The only difference is I run in standalone, can you try that approach and see what happens? Can you run in debug mode? (ie proftpd –n –dx, where x = a number between 1 and 9)

Problem: Valid user accounts are not able to log in. System: Sun SPARC, running Solaris 7. Hardware details available on request. Symptoms: (From perl Net::FTP, Debug mode)... Net::FTP: Net::FTP(2.53) Net::FTP: Exporter Net::FTP: Net::Cmd(2.16) Net::FTP: IO::Socket::INET Net::FTP: IO::Socket(1.1603) Net::FTP: IO::Handle(1.1505) Net::FTP=GLOB(0xc9268)<<< 220 members.friendfactory.com Net::FTP=GLOB(0xc9268)>>> user whoami Net::FTP=GLOB(0xc9268)<<< 331 Password required for whoami. Net::FTP=GLOB(0xc9268)>>> PASS .... Net::FTP=GLOB(0xc9268)<<< 230 User whoami logged in. Net::FTP=GLOB(0xc9268)>>> QUIT Net::FTP=GLOB(0xc9268)<<< 221 Goodbye. Net::FTP=GLOB(0xc604c)<<< 220 members.friendfactory.com Net::FTP=GLOB(0xc604c)>>> user whatsyrname Net::FTP=GLOB(0xc604c)<<< 331 Password required for whatsyrname. Net::FTP=GLOB(0xc604c)>>> PASS .... Net::FTP=GLOB(0xc604c)<<< 530 Login incorrect. Net::FTP=GLOB(0xc604c)>>> QUIT Net::FTP=GLOB(0xc604c)<<< 421 Login Timeout (300 seconds): closing control connection. >From ftpdlog: pluto.driftwood.com 207.229.89.167 nobody [23/Jan/2000:15:14:22 –0800] "USER whoami" 331 – pluto.driftwood.com 207.229.89.167 whoami [23/Jan/2000:23:14:22 +0000] "PASS (hidden)" 230 – pluto.driftwood.com 207.229.89.167 nobody [23/Jan/2000:15:14:23 –0800] "USER whatsyrname" 331 – pluto.driftwood.com 207.229.89.167 nobody [23/Jan/2000:15:14:24 –0800] "PASS (hidden)" 530 –

Notes on above: 1) The output is from a perl script which goes cycling through random sets of known usernames and passwords in order to do performance testing on our new authentication server. The names of the users have been changed to protect the innocent. 2) Note that the timezone in the ftpdlog changes from –0800 to +0000 when there is a sucessful login. Note also that the username registers sucessfully. 3) This problem has repeated itself using Solaris /usr/bin/ftp, ncftp, and perl Net::FTP. As such, I don't think it's a client issue per se. 4) On Net::FTP (the only one which I have done extensive testing on) we have gotten about 80% reproducability on a sample of 2000 attempted connections. The other 20% of the queries validate normally. 5) I originally thought that the problem may be related to a disparity with the time clocks between the client and server machines. (a mystic longshot, given that RFC–959 doesn't exchange date/time stamps per se). An earlier test eradicated this problem by synchronizing the system clocks. Any ideas on what the errors of my ways might be?

I am intending to use proftpd to set up an ftp server (and thint it is a Good Thing) Configuration is a Linux box, RedHat 6.1 kernel 2.2.12–20 Intending to use simply /etc/passwd and shadow for authentication to begin with. Therefore I'm using PAM, and have configured /etc/init.d/ftp as per the README.PAM file Problem is that at authentication time the PAM module is tryint to make connections back to the calling machine on Port 113, which is the port for the auth protocol. Has anyone come across this one please, and how do we stop it doing this? It is not what we want the ftp server to do, and is making authentication take a long time. Sorry if this is a real simple RTFM.

John Hearns wrote: > > Problem is that at authentication time the > PAM module is tryint to make connections back to the > calling machine on Port 113, which is the > port for the auth protocol. I answer my own question by finding the IdentLookups directive. I hang my head in shame – I should have all my merit badges ceremonially stripped off and be drummed out of the sys admin brownies, to be banished to scratching a poor existence loading Windows printer drivers. Apologies for a wasted post to the list – I'm not a baby sys admin who's unwrapped his first box of Linux CDs (honest!). I only asked for help after watching loads of firewall log traces and a lot of head scratching. One tip though – I finally got clued into my problem by finding documentation on the Apache IdentityCheck directive, which the IdentLookups directive is similar to.

I've been working with proftpd for a while and I still don't quite understand how authentication works. The object is to have users listed in /etc/passwd authenticated via system methods which works but I would like to have an additional password file used for guest users that are confined to their home dir. Can anyone suggest how to do this or point me to some documentation. I'm using the config file that gets installed when you run make install with 1.2.0pre8 with the addition of the two lines below. AuthUserFile /usr/local/etc/test.pwd AuthPAMAuthoritative off

Can someoe explain these two directives please? What I would like to know is the following: 1. Must they exactly follow the format of /etc/passwd and /etc/group?=20 2. Which crypt must be used for the password – crypt or MD5? 3. Under which user will the VirtualHost execute? 4. How do they influence a chroot'd <VirtualHost>? 5. How is an <Anonymous> section inside a <VirtualHost> influenced? 6. If (1) is true, what is the significance of the UIDs and GIDs?

Note to Mark: Maybe we should clarify the documentation on the AuthUserFile directive? > 2. Which crypt must be used for the password – crypt or MD5? The password check is done via the crypt() call...so if your system happens to map that to an MD5 version of crypt(), then it's MD5. There's a script in the contrib directory called genuser.pl that will generate valid usename:password crypt–ed pairs for you. > 3. Under which user will the VirtualHost execute? Pardon? Under whatever user you've specified via the User directive of course. > 4. How do they influence a chroot'd <VirtualHost>? They don't really. Whatever you've listed as the home directories is used in determined a user–chroot jail as appropriate. > 5. How is an <Anonymous> section inside a <VirtualHost> influenced? Huh? > 6. If (1) is true, what is the significance of the UIDs and GIDs? > UIDs and GIDs are your method to control access on the system. Presumably you have these allocated in some fashion. ProFTPD will honor whatever you specify.

Can somebody please tell me how to create the AuthUser file? I can't seem to find out how I should encode the passwords in that file.

I am building a ProFTP ratio server. I was able to get mod.ratio installed and working properly and I think I understand the rest of the configuration that I need to do. Except, I want a basic anonymous user, a user1/user1 (username/password) user with better access and ration, and a user2/user2 with full access no ratio. As I understand it my conf file would look something like this... AuthUserFile /usr/ftp/etc/passwd <Anonymous ~ftp> User ftp Group ftp UserAlias anonymous ftp MaxClients 10 "Sorry, the maximum number of allowed users are already connected (%m) " MaxClientsPerHost 1 "Sorry, you may not connect more than one time." RequireValidShell off DisplayLogin welcome.msg DisplayFirstChdir .message <Limit WRITE> AllowUser User1 AllowUser User2 DenyAll </Limit> <Limit STOR> AllowAll </Limit> Ratios on UserRatio * 0 0 1 0 UserRatio user1 0 0 10 0 UserRatio user2 0 0 0 0 </Anonymous> How do I generate the AuthUserFile so that this will work?? Thanks in advance!

I'm using pre9, with an anon section like this: <Anonymous /virtual/ftp> User virtual Group virtual UserAlias joe virtual AuthAliasOnly on </Anonymous> This works as expected –– I can't login anonymously with "virtual", but I can with "joe". When I do this: <Anonymous /virtual/ftp> User virtual Group virtual UserAlias joe virtual AuthAliasOnly off </Anonymous> There is no change –– I still can't login anonymously with "virtual", but I should be able to. Now, if I do this: <Anonymous /virtual/ftp> User virtual Group virtual UserAlias joe virtual # AuthAliasOnly off </Anonymous>

...it _does_ allow me to login anonymously with "virtual". In other words, "AuthAliasOnly off" doesn't work. If I want the functionality that it provides, I have to comment it out or remove it completely.

Yes, you can. Go to httpd://www.proftpd.org Read the Documentation. Using an alternate password is documented very well. Tsanko Stoev Lyon College > Can I use a different password file (other than /etc/passwd) with in the > same domain???

My mail server is down so i gotta use hotmail...ugh. Anyway... I fixed the anonymous login problem by adding a "RequireValidShell off" into proftpd.conf. Now my problem is that valid users of the machine cannot login into the proftpd service but can with ssh and telnet. Anyone know any reasons as to why that is happening? Thanks in advance! −Andrew

I tried that already but it still does not work. >From: Matt Critcher <MCritch@lifeplususa.com> >Reply−To: proftpd@proftpd.org >To: "'proftpd@proftpd.org'" <proftpd@proftpd.org> >Subject: RE: [ProFTPD] Login Problems >Date: Wed, 15 Mar 2000 08:33:14 −0600 > >You probably dont have an entry in /etc/pam.d/ for ftp > >you have to put a file there called ftp that contains something similar the >following: > >#%PAM−1.0 >auth required pam_listfile.so item=user sense=deny >file=/etc/ftpusers onerr=succeed >auth sufficient pam_userdb.so icase db=/tmp/dbtest >auth required pam_pwdb.so shadow nullok try_first_pass >auth required pam_shells.so >account required pam_pwdb.so >session required pam_pwdb.so > >or something like this. its all i can remember without being on my machine >(stuck to the hells of windows at work). in any case there is a file >called >README.PAM that comes with the src for proftpd that has the correct >contents.

You probably dont have an entry in /etc/pam.d/ for ftp you have to put a file there called ftp that contains something similar the following: #%PAM−1.0 auth required pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed auth sufficient pam_userdb.so icase db=/tmp/dbtest auth required pam_pwdb.so shadow nullok try_first_pass auth required pam_shells.so account required pam_pwdb.so session required pam_pwdb.so or something like this. its all i can remember without being on my machine (stuck to the hells of windows at work). in any case there is a file called README.PAM that comes with the src for proftpd that has the correct contents.

I'm having the exact same problem. I have a binary and config file that allows logins on my 6.0 machines, but when I copy them to a 6.1 box, I cannot login as a normal user... I get this message (proftpd −n): localhost (10.80.80.10[10.80.80.10]) − PAM(bobo): Authentication failure. localhost (10.80.80.10[10.80.80.10]) − USER bobo (Login failed): Incorrect password.

---

proftpd.chmod

I'm using RedHat 6.1 and proftpd 1.2.0pre10 (and I was trying with pre2) and cannot change file permissions (under no circumstances). The funny thing is that using a Windows client like CuteFTP this client somehow seems to know that from somewhere since the option change file attributes is disabled. When trying to do it with the ftp client from linux I always get permission denied...

I'm not even using Anonymous Blocks...! What I have is a Directory Block with users home Dirs and within that just a Limit block that say Everything is allowed − at least that's what my conf= ig has come down to when trying to solve this problem... :)=20 "Wimmer, Tobias" wrote: >=20 > I'm not even using Anonymous Blocks...! >=20 > What I have is a Directory Block with users home Dirs and within that j= ust a > Limit block that say Everything is allowed − at least that's what my co= nfig > has come down to when trying to solve this problem... :) >=20 Could it be that the permission denied comes from the Unix file system? In that case it should be visible by running strace on the proftpd process that is handling the session. When I log into a shell with this user, setting file permissions does wor= k,but I'll give it a try with strace... > > Hi, > > Simple Question − (Hard answer?): > > I'm using RedHat 6.1 and proftpd 1.2.0pre10 (and I was trying with pre2) and > cannot change file permissions (under no circumstances). The funny thing is > that using a Windows client like CuteFTP this client somehow seems to know > that from somewhere since the option change file attributes is disabled. > When trying to do it with the ftp client from linux I always get permission > denied... > > Anyone any ideas? I noticed this too, on SuSE 6.1, proftpd pre10; debug level 5 just says Apr 6 20:31:35 novix proftpd[1158]: novix (moniek[10.1.0.1]) − received: SITE CHMOD 611 tim.htm Apr 6 20:31:35 novix proftpd[1158]: novix (moniek[10.1.0.1]) − in dir_check(): path = '/tmp/tim.htm', fullpath =

'/home/jei/tmp/tim.htm'. After some experimenting I noticed my test user/directory were configured inside an <anonymous> block, after i got them out of there the chmod did work. My current guess is that "Anonymous" has built−in restrictions (no overwrite, rename, chmod, ..) that cannot be lifted by <limit ..> blocks. I peek in the source now and then, but 35k+ lines is a lot to look at. I have some trouble when I try to allow chmod file from my user. I have looking for directives in the doc, but I didn't find anything. So what is the way to allow my user to chmod their file in they account ?

---

proftpd.ls

if I'm ftping from a remote server. ftp> ls 200 PORT command successful. 421 Service not available, remote server has closed connection edward> cat syslog Feb 28 14:00:17 edward proftpd[776]: edward (localhost[127.0.0.1]) − attempted bind to 127.0.0.1, port 20 Feb 28 14:00:17 edward proftpd[776]: edward (localhost[127.0.0.1]) − bind() failed in inet_create_connection(): Permission denied Feb 28 14:00:17 edward proftpd[776]: edward (localhost[127.0.0.1]) − Check the ServerType directive to ensure you are configured correctly.

Mark was referring to the user you were logged in as when you actually launched the ProFTPD server. −−−−−Original Message−−−−− From: Norio Kashiwagi [mailto:kasiwagi@kakoi.co.jp] Sent: Monday, February 28, 2000 10:19 AM To: proftpd@proftpd.org Subject: Re: [ProFTPD] Can't do "ls" command > > Feb 28 14:00:17 edward proftpd[776]: edward (localhost[127.0.0.1]) > > − bind() failed in inet_create_connection(): Permission > > denied > > Did you start ftp as root? Yes. −−− proftpd.conf −−− # Set the user and group that the server normally runs at. #uSEr nobody User root Group nogroup

Thanks for help, Chris

/etc/inetd.conf is ftp stream tcp nowait root /usr/local/sbin/proftpd/in.proftpd in.proftpd But ws−ftp's "Passive transfers" is no problem. ProFTPD is Passive mode only?

> > Feb 28 14:00:17 edward proftpd[776]: edward (localhost[127.0.0.1]) > > − bind() failed in inet_create_connection(): Permission > > denied > > Did you start ftp as root? Yes. −−− proftpd.conf −−− # Set the user and group that the server normally runs at. #uSEr nobody User root Group nogroup

−− To unsubscribe, send mail to proftpd−request@proftpd.org with "unsubscribe" in the subject field of the message. Please read the documentation and the FAQ before posting a question −− chances are it's already been answered. http://www.proftpd.org −− The Official ProFTPD web site. http://bugs.proftpd.org −− Bug reporting and feature requests. http://www.proftpd.org/docs/ −− The latest ProFTPD documentation and FAQ. From proftpd−request@tos.net Mon Feb 28 15:02:33 2000 Received: from firewall.vom.tm ([212.32.5.30] helo=flyhmstr.vom.tm) by weasel.vom.tm with esmtp (Exim 3.12 #1) id 12PRgv−0007aS−00 for mark@weasel.vom.tm; Mon, 28 Feb 2000 15:02:33 +0000 Received: from starbase.tos.net ([209.212.188.150]) by flyhmstr.vom.tm with esmtp (Exim 3.11 #1 (Debian)) id 12PRgt−0006DF−00 for <hamster@vom.tm>; Mon, 28 Feb 2000 15:02:32 +0000 Received: (from listserv@localhost) by starbase.tos.net (8.9.3/8.9.3) id JAA08446; Mon, 28 Feb 2000 09:02:18 −0600 Resent−Date: Mon, 28 Feb 2000 09:02:18 −0600 Date: Mon, 28 Feb 2000 14:53:41 +0000 From: Mark Lowes <hamster@vom.tm> To: proftpd@proftpd.org Subject: Re: [ProFTPD] Can't do "ls" command Message−ID: <20000228145341.A29107@weasel.vom.tm> References: <002101bf81f9$3f83b820$843365c1@kakoi.co.jp> Mime−Version: 1.0 Content−Type: text/plain; charset=us−ascii User−Agent: Mutt/1.0.1i In−Reply−To: <002101bf81f9$3f83b820$843365c1@kakoi.co.jp>; from kasiwagi@kakoi.co.jp on Mon, Feb 28, 2000 at 11:36:44PM +0900 Resent−Message−ID: <7cBfHD.A.yAC.Owou4@starbase.tos.net> Resent−From: proftpd@proftpd.org Reply−To: proftpd@proftpd.org X−Mailing−List: <proftpd@proftpd.org>

archive/latest/3697 X−Loop: proftpd@proftpd.org Precedence: list Resent−Sender: proftpd−request@proftpd.org Resent−Bcc: X−Filter: proftpd Status: RO Content−Length: 872 Lines: 27 On Mon, Feb 28, 2000 at 11:36:44PM +0900, Norio Kashiwagi wrote: > Feb 28 14:00:17 edward proftpd[776]: edward (localhost[127.0.0.1]) > − bind() failed in inet_create_connection(): Permission > denied Did you start ftp as root? Mark

−− The Flying Hamster <hamster@suespammers.org> http://hamster.wibble.org/ Do not meddle in the affairs of hamsters. Just don't. It's not worth it. − Ailbhe on #afp −− To unsubscribe, send mail to proftpd−request@proftpd.org with "unsubscribe" in the subject field of the message. Please read the documentation and the FAQ before posting a question −− chances are it's already been answered. http://www.proftpd.org −− The Official ProFTPD web site. http://bugs.proftpd.org −− Bug reporting and feature requests. http://www.proftpd.org/docs/ −− The latest ProFTPD documentation and FAQ. From proftpd−request@tos.net Mon Feb 28 15:25:03 2000 Received: from firewall.vom.tm ([212.32.5.30] helo=flyhmstr.vom.tm) by weasel.vom.tm with esmtp (Exim 3.12 #1) id 12PS2h−0007bQ−00 for mark@weasel.vom.tm; Mon, 28 Feb 2000 15:25:03 +0000 Received: from starbase.tos.net ([209.212.188.150]) by flyhmstr.vom.tm with esmtp (Exim 3.11 #1 (Debian)) id 12PS2e−0006I8−00 for <hamster@vom.tm>; Mon, 28 Feb 2000 15:25:00 +0000 Received: (from listserv@localhost) by starbase.tos.net (8.9.3/8.9.3) id JAA09041; Mon, 28 Feb 2000 09:24:40 −0600 Resent−Date: Mon, 28 Feb 2000 09:24:40 −0600 Message−ID: <004801bf81ff$1c09a160$843365c1@kakoi.co.jp> From: "Norio Kashiwagi" <kasiwagi@kakoi.co.jp> To: <proftpd@proftpd.org> References: <002101bf81f9$3f83b820$843365c1@kakoi.co.jp> <20000228145341.A29107@weasel.vom.tm> Subject: Re: [ProFTPD] Can't do "ls" command Date: Tue, 29 Feb 2000 00:18:41 +0900 MIME−Version: 1.0 Content−Type: text/plain; charset="iso−8859−1" Content−Transfer−Encoding: 7bit X−Priority: 3 X−MSMail−Priority: Normal X−Mailer: Microsoft Outlook Express 5.00.2314.1300 X−MimeOLE: Produced By Microsoft MimeOLE V5.00.2314.1300 Resent−Message−ID: <NLL9UC.A.fJC.ZEpu4@starbase.tos.net> Resent−From: proftpd@proftpd.org Reply−To: proftpd@proftpd.org X−Mailing−List: <proftpd@proftpd.org> archive/latest/3698 X−Loop: proftpd@proftpd.org Precedence: list Resent−Sender: proftpd−request@proftpd.org Resent−Bcc: X−Filter: proftpd Status: RO Content−Length: 840 Lines: 24

> > Feb 28 14:00:17 edward proftpd[776]: edward (localhost[127.0.0.1]) > > − bind() failed in inet_create_connection(): Permission > > denied > > Did you start ftp as root? Yes. −−− proftpd.conf −−− # Set the user and group that the server normally runs at. #uSEr nobody User root Group nogroup

I can't get mirror to work well with symlinks and proftpd. After testing a while I found that proftpd seems to make no difference between the two commands: ls −lR ls −lLR where the last should make proftpd show up the real files, not the symlinks which point to the files. Here is a simple test: cd /var/tmp mkdir −p d0/d1/d2/d3 mkdir d0/real_dir date >d0/real_dir/test ln −s ../../../real_dir d0/d1/d2/d3 ln −s ../../../real_dir/test d0/d1/d2/d3/data_link Then, proftpd gives no difference with the above two ftp−commands: ... d0: drwxr−xr−x 4 root root 1024 Feb 29 06:26 . drwxrwxrwx 5 root root 3072 Feb 29 08:35 .. drwxr−xr−x 3 root root 1024 Feb 29 06:26 d1 drwxr−xr−x 2 root root 1024 Feb 29 06:26 real_dir d0/d1: drwxr−xr−x 3 root root 1024 Feb 29 06:26 . drwxr−xr−x 4 root root 1024 Feb 29 06:26 .. drwxr−xr−x 3 root root 1024 Feb 29 06:26 d2 d0/d1/d2: drwxr−xr−x 3 root root 1024 Feb 29 06:26 . drwxr−xr−x 3 root root 1024 Feb 29 06:26 .. drwxr−xr−x 2 root root 1024 Feb 29 06:26 d3 d0/d1/d2/d3: drwxr−xr−x 2 root root 1024 Feb 29 06:26 . drwxr−xr−x 3 root root 1024 Feb 29 06:26 .. lrwxrwxrwx 1 root root 22 Feb 29 06:26 data_link −> ../../../rea l_dir/test lrwxrwxrwx 1 root root 17 Feb 29 06:26 real_dir −> ../../../real _dir d0/real_dir: drwxr−xr−x 2 root root 1024 Feb 29 06:26 . drwxr−xr−x 4 root root 1024 Feb 29 06:26 .. −rw−r−−r−− 1 root root 29 Feb 29 06:26 test ... Of course, you may set ShowSymlinks off, but this is NOT the desired mode.

Thanks, Andreas Wehler

+−−−−−− "Dr. Andreas Wehler" wrote (Tue, 29−Feb−00, 08:44 +0100): | | I can't get mirror to work well with symlinks and proftpd. | After testing a while I found that proftpd seems to make | no difference between the two commands: | ls −lR | ls −lLR | where the last should make proftpd show up the real files, | not the symlinks which point to the files. That's right. The mod_ls module doesn't recognize the −L option. | Of course, you may set ShowSymlinks off, but this is NOT the | desired mode. Try the attached patch, made against the current CVS sources (mod_ls.c 1.21 2000/01/23). This is my first look at the mod_ls.c code, and I didn't spend much time on it, so there is some doubt in my mind about it, particularly the push_cwd/pop_cwd bits. So, don't be shy with feedback, positive or negative. If it looks OK, I can submit the patch. It's probably too late to get by the 1.2.0 release code freeze, though. While we are thinking about this stuff, I noticed that a couple of other common ls options are missing that might be reasonable to add, e.g. −A, −p, and −s. It also occurred to me that there might be legitimate use for a method to disable the −R option, from a resource conservation point of view. This could be either a directive, or a sentinel file (like the wu−ftpd .notar), or both. A more comprehensive directive, say "LsDisableOptions", might disable ls options selectively. Thoughts? Speaking of .notar, mod_tar doesn't appear to use that convention. Does anyone miss that feature?

Charles Seeger wrote: > > +−−−−−− "Dr. Andreas Wehler" wrote (Tue, 29−Feb−00, 08:44 +0100): > | > | I can't get mirror to work well with symlinks and proftpd. > | After testing a while I found that proftpd seems to make > | no difference between the two commands: > | ls −lR > | ls −lLR > | where the last should make proftpd show up the real files, > | not the symlinks which point to the files. > > That's right. The mod_ls module doesn't recognize the −L option. > > | Of course, you may set ShowSymlinks off, but this is NOT the > | desired mode. > > Try the attached patch, made against the current CVS sources > (mod_ls.c 1.21 2000/01/23). Thank you very much! It works like a charme. And this within hours. What sort of commercial software company may reach this level of support?

Thanks. Andreas Wehler −− CCS Informationssysteme GmbH Tel.: (+49) 211 − 52740 − 228 Dr.−Ing. Andreas Wehler Fax.: (+49) 211 − 52740 − 280 http://www.ccs−web.com −− To unsubscribe, send mail to proftpd−request@proftpd.org with "unsubscribe" in the subject field of the message. Please read the documentation and the FAQ before posting a question −− chances are it's already been answered. http://www.proftpd.org −− The Official ProFTPD web site. http://bugs.proftpd.org −− Bug reporting and feature requests. http://www.proftpd.org/docs/ −− The latest ProFTPD documentation and FAQ. From proftpd−request@tos.net Sat Feb 12 00:07:39 2000 Received: from firewall.vom.tm ([212.32.5.30] helo=flyhmstr.vom.tm) by weasel.vom.tm with esmtp (Exim 3.12 #1) id 12JQ67−0000GC−00 for mark@weasel.vom.tm; Sat, 12 Feb 2000 00:07:39 +0000 Received: from starbase.tos.net ([209.212.188.150]) by flyhmstr.vom.tm with esmtp (Exim 3.11 #1 (Debian)) id 12JQ65−0001KI−00 for <hamster@vom.tm>; Sat, 12 Feb 2000 00:07:38 +0000 Received: (from listserv@localhost) by starbase.tos.net (8.9.3/8.9.3) id OAA09338; Fri, 11 Feb 2000 14:51:54 −0600 Resent−Date: Fri, 11 Feb 2000 14:51:54 −0600 From: "Lan Tran" <lan@recol.com> To: <proftpd@proftpd.org> Date: Fri, 11 Feb 2000 15:44:38 −0500 Message−ID: <NDBBLOICLKHIHOKDAEGKAEGFCBAA.lan@recol.com> MIME−Version: 1.0 Content−Type: text/plain; charset="iso−8859−1" Content−Transfer−Encoding: 7bit X−Priority: 3 (Normal) X−MSMail−Priority: Normal X−Mailer: Microsoft Outlook IMO, Build 9.0.2416 (9.0.2910.0) In−Reply−To: <56183983FF5AD31185F70060B06DF1683FF394@dot.ctcts.com> Importance: Normal X−MimeOLE: Produced By Microsoft MimeOLE V5.00.2314.1300 Subject: [ProFTPD] Can't do "ls" Resent−Message−ID: <JDSsr.A.bMC.dRHp4@starbase.tos.net> Resent−From: proftpd@proftpd.org Reply−To: proftpd@proftpd.org X−Mailing−List: <proftpd@proftpd.org> archive/latest/3385 X−Loop: proftpd@proftpd.org Precedence: list Resent−Sender: proftpd−request@proftpd.org Resent−Bcc: X−Filter: proftpd Status: RO Content−Length: 689 Lines: 22

I can do a "ls" if I ftp from localhost. However, "ls" does not work if I'm ftping from a remote server. ======= > | After testing a while I found that proftpd seems to make > | no difference between the two commands: > | ls −lR > | ls −lLR > | where the last should make proftpd show up the real files, > | not the symlinks which point to the files. > > That's right. The mod_ls module doesn't recognize the −L option. > > |

Of course, you may set ShowSymlinks off, but this is NOT the > | desired mode. > > Try the attached patch, made against the current CVS sources > (mod_ls.c 1.21 2000/01/23). Thank you very much! It works like a charme. And this within hours. What sort of commercial software company may reach this level of support?

I can do a "ls" if I ftp from localhost. However, "ls" does not work if I'm ftping from a remote server. ftp> ls 200 PORT command successful. It hangs like this forever. Running on RH60 proftpd−1.2pre10 Kernel 2.2.14. Any ideas?

Thanks. −− To unsubscribe, send mail to proftpd−request@proftpd.org with "unsubscribe" in the subject field of the message. Please read the documentation and the FAQ before posting a question −− chances are it's already been answered. http://www.proftpd.org −− The Official ProFTPD web site. http://bugs.proftpd.org −− Bug reporting and feature requests. http://www.proftpd.org/docs/ −− The latest ProFTPD documentation and FAQ. From proftpd−request@tos.net Mon Feb 14 01:34:15 2000 Received: from firewall.vom.tm ([212.32.5.30] helo=flyhmstr.vom.tm) by weasel.vom.tm with esmtp (Exim 3.12 #1) id 12KAP1−0006ds−00 for mark@weasel.vom.tm; Mon, 14 Feb 2000 01:34:15 +0000 Received: from starbase.tos.net ([209.212.188.150]) by flyhmstr.vom.tm with esmtp (Exim 3.11 #1 (Debian)) id 12KAOz−00081n−00 for <hamster@vom.tm>; Mon, 14 Feb 2000 01:34:14 +0000 Received: (from listserv@localhost) by starbase.tos.net (8.9.3/8.9.3) id TAA11707; Sun, 13 Feb 2000 19:33:12 −0600 Resent−Date: Sun, 13 Feb 2000 19:33:12 −0600 From: L.M.D.Cranswick@dl.ac.uk (L. Cranswick) Message−Id: <200002140125.BAA07792@xrdsv1> Subject: Re: [ProFTPD] Can't do "ls" To: proftpd@proftpd.org Date: Mon, 14 Feb 2000 01:25:21 +0000 (GMT) In−Reply−To: <NDBBLOICLKHIHOKDAEGKAEGFCBAA.lan@recol.com> from "Lan Tran" at Feb 11, 2000 03:44:38 PM X−Mailer: ELM [version 2.5 PL2] MIME−Version: 1.0 Content−Type: text/plain; charset=us−ascii Content−Transfer−Encoding: 7bit Resent−Message−ID: <v−puK.A.−xC.Tm1p4@starbase.tos.net> Resent−From: proftpd@proftpd.org Reply−To: proftpd@proftpd.org X−Mailing−List: <proftpd@proftpd.org> archive/latest/3425 X−Loop: proftpd@proftpd.org Precedence: list Resent−Sender: proftpd−request@proftpd.org Resent−Bcc: X−Filter: proftpd Status: RO Content−Length: 1379 Lines: 47

> I can do a "ls" if I ftp from localhost. However, "ls" does not work > if I'm ftping from a remote server. > > ftp> ls > 200 PORT command successful. > > It hangs like this forever. Running on RH60 proftpd−1.2pre10 Kernel 2.2.14. > Any ideas? Sorry if this has been answered − just quickly getting through backlog of few 200 or so E−mails. Do you have a firewall − or tunnelling (if being used) configured correctly? This could be blocking the output?

On Mon, Feb 14, 2000 at 01:25:21AM +0000, L.M.D.Cranswick@dl.ac.uk wrote: > > I can do a "ls" if I ftp from localhost. However, "ls" does not work > > if I'm ftping from a remote server. > > > > ftp> ls > > 200 PORT command successful. > > > > It hangs like this forever. Running on RH60 proftpd−1.2pre10 Kernel 2.2.14. > > Any ideas? This was one of the problems I experienced with proftpd under FreeBSD 4.0. The problem, from what I can tell (MacGyver has yet to respond to my Emails. Did you die, Mac?), seemed to be related to the fact that proftpd has #ifdefs which look for the definition of 'FREEBSD3', generated during configure time, taken from uname (most likely). Since 'FREEBSD4' wasn't listed, well, you can see the problem. The code that used this was in inet.c, if my memory serves me right. The patch was very small. Check your logfiles. The error I was receiving made zero sense; proftpd was claiming port 21 was already bound (ab− surd, since the daemon was in standalone mode ;−)). Can't really help you much more than this, as you're using Linux. Best of luck.

> I can do a "ls" if I ftp from localhost. However, "ls" does not work > if I'm ftping from a remote server. Well, my gut feeling (sorry, just played who wants to be a millionaire on abc.com) is to put this in your proftpd.conf: LsDefaultOptions −al That will add −al, but you probably want that. If adding −al doesn't work out, tell me.

Nope, doesn't work. I'm behind a firewall. Tried setting to passive mode: "We only support stream mode, sorry." I have to recompile the source?

−−−−−Original Message−−−−− From: Vincent Paglione [mailto:mogom@jtan.com] Sent: Friday, February 11, 2000 3:58 PM To: proftpd@proftpd.org Subject: Re: [ProFTPD] Can't do "ls" > I can do a "ls" if I ftp from localhost. However, "ls" does not work > if I'm ftping from a remote server. Well, my gut feeling (sorry, just played who wants to be a millionaire on abc.com) is to put this in your proftpd.conf: LsDefaultOptions −al That will add −al, but you probably want that. If adding −al doesn't work out, tell me.

On Fri, Feb 11, 2000 at 04:15:47PM −0500, Lan Tran wrote: > Nope, doesn't work. I'm behind a firewall. Tried setting to passive mode: > "We only support stream mode, sorry." I have to recompile the source? it sounds like you can't run a FTP server ... what is happening is that a client can connect to port 21 to send the commands through, but data transfers (either coming from a non−restricted port > 1023, or from port 20) going to the client are blocked on one end or the other. you'll want to talk to the people in charge of your local firewall and see what their setup is. typically I'd allow: incoming to server on port 20−21, >1023 outgoing from server to all client ports (if you have to limit it, ports >1023)

On Fri, Feb 11, 2000 at 03:44:38PM −0500, Lan Tran wrote: > ftp> ls > 200 PORT command successful. > > It hangs like this forever. Running on RH60 proftpd−1.2pre10 Kernel 2.2.14. > Any ideas? it sounds like there's a firewall in the way blocking the request. try passive mode and see what happens.

---

proftpd.sql

I'm trying to replace my current ftpd/realusers with proftpd/sqlpw auth and I get the following problem with mod_sqlpw set to non−authoritative: if real user tries to login to ftp with his name, that also exists in the SQL DB but with different password and homedir, he gets the uid,gid and home of the SQL user! Is this the way it should work? If no, is there someone who fixed this? I'm too lazy to make the work already done by somebody ;) To rephrase your question: You are saying that if a user logs in who's login id exists in both /etc/passwd and in the MySQL DB and the user logs in gets the home directory of the user in the MySQL DB. I am assuming that the passwords are the same in both /etc/passwd and MySQL DB. That sounds right to me... Though I don't think we have a good description of what happens if you set SQLauthoritative to "off" and build proftpd with mod_sqlpw... I hope Mark may have some more insight here...

At 15:36 10/04/2000 +0400, **Roman Korolyov**, has written a message, and here is the reply : >Hi! >I'm trying to replace my current ftpd/realusers with proftpd/sqlpw auth >and I get the following problem with mod_sqlpw set to non−authoritative: Sorry this is not an answer, but after checking my archive I found this −−−−− ProFTPD and mod_sqlpw create a security hole
−−

SUMMARY Compiling the mod_sqlpw module into ProFTPD makes it possible for local users to view the passwords of users who have connected to the ftp server. When the module is used, it writes information to wtmp. Unfortunately, it writes the password to wtmp where the username should be. The passwords can be seen when a command such as 'last' is used locally.

DETAILS Solution: Adding the following to your ProFTPd configuration file should solve this problem: <Global> Wtemplog off </Global> Wtmplog details below: WtmpLog Syntax: WtmpLog on|off|NONE Default: WtmpLog on Context: server config, <VirtualHost>, <Anonymous>, <Global> Compatibility: 1.1.7 and later The WtmpLog directive controls proftpd's logging of ftp connections to the host system's wtmp file (used by such commands as `last'). By default, all connections are logged via wtmp.

ADDITIONAL INFORMATION The mentioned vulnerability has been discovered by: <mailto:toddc@NET–LINK.NET> Todd C. Campbell.

Got the latest CVS a few days ago, and the wtmp output of proftpd (with mod_sqlpw & mysql) is still really wired : sometimes is the password (!) logged, and the other times, a kind of random string (*J***J**).

*J***J** ftp 195.130.185.44 Sat Apr 8 01:02 – 01:07 (00:04) *J***J** ftp megazh–d–218.agr Sat Apr 8 00:17 – 00:18 (00:00) *J***J** ftp 195.130.185.92 Fri Apr 7 23:30 – 23:39 (00:08) *J***J** ftp 195.130.185.92 Fri Apr 7 23:14 – 23:14 (00:00) *J***J** ftp 195.130.185.92 Fri Apr 7 23:12 – 23:14 (00:01) *J***J** ftp megazh–d–56.agri Fri Apr 7 21:15 – 21:15 (00:00) *J***J** ftp orion.www–hostin Fri Apr 7 14:13 – 14:14 (00:00) *J***J** ftp orion.www–hostin Fri Apr 7 14:11 – 14:12 (00:00) wxcff.97 ftp 147.78.21.30 Fri Apr 7 13:37 – 13:38 (00:01) wxcff.97 ftp 147.78.21.30 Fri Apr 7 13:34 – 13:35 (00:01) wxcff.97 ftp 147.78.21.30 Fri Apr 7 13:31 – 13:31 (00:00) Af773,r ftp sulzer.ch Fri Apr 7 11:51 – 11:52 (00:00) wxcff.97 ftp 147.78.21.30 Fri Apr 7 10:39 – 10:39 (00:00) wxcff.97 ftp 147.78.21.30 Fri Apr 7 10:38 – 10:38 (00:00)

Are these (really practical btw) modules not maintained ? Logging the password to the wtmp is just an *huge* security hog... :(

At 15:36 10/04/2000 +0400, **Roman Korolyov**, has written a message, and here is the reply : >Hi! >I'm trying to replace my current ftpd/realusers with proftpd/sqlpw auth >and I get the following problem with mod_sqlpw set to non–authoritative: Sorry this is not an answer, but after checking my archive I found this ––––– ProFTPD and mod_sqlpw create a security hole
––
––––

SUMMARY Compiling the mod_sqlpw module into ProFTPD makes it possible for local users to view the passwords of users who have connected to the ftp server. When the module is used, it writes information to wtmp. Unfortunately, it writes the password to wtmp where the username should be. The passwords can be seen when a command such as 'last' is used locally. DETAILS Solution: Adding the following to your ProFTPd configuration file should solve this problem: <Global> Wtemplog off </Global> Wtmplog details below: WtmpLog Syntax: WtmpLog on|off|NONE Default: WtmpLog on Context: server config, <VirtualHost>, <Anonymous>, <Global> Compatibility: 1.1.7 and later The WtmpLog directive controls proftpd's logging of ftp connections to the host system's wtmp file (used by such commands as `last'). By default, all connections are logged via wtmp. ADDITIONAL INFORMATION The mentioned vulnerability has been discovered by: <mailto:toddc@NET–LINK.NET> Todd C. Campbell.

http://bugs.proftpd.org/show_bug.cgi?id=108 *** shadow/108 Thu Apr 6 12:01:58 2000 ––– shadow/108.tmp.7160 Thu Apr 6 13:58:46 2000 *************** *** 23,25 **** ––– 23,39 –––– 1) references missing flags.c program, need to remove references to flags.* 2) modmysql does not have the ".o" extension in the file. That needs ot be added... + + ––––––– Additional Comments From hamster@hamster.wibble.org 04/06/00 13:58 ––––––– + I think the "flags" stuff is a result of the wording in the FAQ, I'll amend the + FAQ to make this clearer. + + It should be + configure ––with–modules='mod_sqlpw:mod_mysql' + + not + + configure ––with–modules='mod_sqlpw:mod_mysql flags' + + flags' +

http://bugs.proftpd.org/show_bug.cgi?id=109 *** shadow/109 Thu Apr 6 12:06:53 2000 ––– shadow/109.tmp.7078 Thu Apr 6 12:06:53 2000 *************** *** 0 **** ––– 1,27 –––– + Bug#: 109 + Product: ProFTPD + Version: 1.2.0pre10 + Platform: PC + OS/Version: Linux + Status: NEW + Resolution: + Severity: normal + Priority: P2 + Component: mod_sqlpw + AssignedTo: proftpd–devel@proftpd.org + ReportedBy: mgrabenstein@mac.com + URL: + Summary: mod_sqlpw (mysql): Make.rules needs libaries and includes specified. + + In Make.rules, The definition for LIBS needs to have −lmysqlclient added to it. + LDFLAGS needs the path to your mysql library added. CPPFLAGS needs the location + of your mysql include files added. + Depending on installation they should look like: + LIBS=−lsupp −ldl −lcrypt −lm −lmysqlclient −lpam + LDFLAGS=−L/home/builds/proftpd−1.2.0pre10/lib −L/usr/local/mysql/lib/mysql + CPPFLAGS= $(DEFAULT_PATHS) $(PLATFORM) −I.. −I$(top_srcdir)/include + −I/usr/local/mysql/include/mysql + + Would be nice if there was a readme somwhere in the distribution that hinted at + these things... + Would be best if configure would do this...

http://bugs.proftpd.org/show_bug.cgi?id=108 *** shadow/108 Thu Apr 6 11:57:28 2000 −−− shadow/108.tmp.7054 Thu Apr 6 12:01:34 2000 *************** *** 11,19 **** AssignedTo: proftpd−devel@proftpd.org ReportedBy: mgrabenstein@mac.com URL: ! Summary: reference to flags.c needs to be removed to compile mod_sqlpw (mysql) To get mod_sqlpw to compile. You must first edit the Make.modules file and remove the references to flags.c. Or is it missing from the pre10 distribution ? mod_sqlpw seems work fine with ou it. :−) −−− 11,25 −−−− AssignedTo: proftpd−devel@proftpd.org ReportedBy: mgrabenstein@mac.com URL: ! Summary: mmod_sqlpw (mysql) in Make.modules, remove flag.* and add .o to modmysql To get mod_sqlpw to compile. You must first edit the Make.modules file and remove the references to flags.c. Or is it missing from the pre10 distribution ? mod_sqlpw seems work fine with ou it. :−) + + −−−−−−− Additional Comments From mgrabenstein@mac.com 04/06/00 12:01 −−−−−−− + Two problems in Make.modules: + 1) references missing flags.c program, need to remove references to flags.* + 2) modmysql does not have the ".o" extension in the file. That needs ot be + added...

http://bugs.proftpd.org/show_bug.cgi?id=108 *** shadow/108 Thu Apr 6 12:01:34 2000 −−− shadow/108.tmp.7065 Thu Apr 6 12:01:58 2000 *************** *** 11,17 **** AssignedTo: proftpd−devel@proftpd.org ReportedBy: mgrabenstein@mac.com URL: ! Summary: mmod_sqlpw (mysql) in Make.modules, remove flag.* and add .o to modmysql To get mod_sqlpw to compile. You must first edit the Make.modules file and remove the references to flags.c. −−− 11,17 −−−− AssignedTo: proftpd−devel@proftpd.org ReportedBy: mgrabenstein@mac.com URL: ! Summary: mod_sqlpw (mysql) in Make.modules, remove flag.* and add .o to modmysql To get mod_sqlpw to compile. You must first edit the Make.modules file and remove the references to flags.c.

http://bugs.proftpd.org/show_bug.cgi?id=107 *** shadow/107 Thu Apr 6 11:54:53 2000 −−− shadow/107.tmp.7006 Thu Apr 6 11:54:53 2000 *************** *** 0 **** −−− 1,24 −−−− + Bug#: 107 + Product: ProFTPD + Version: 1.2.0pre10 + Platform: PC + OS/Version: Linux + Status: NEW + Resolution: + Severity: major + Priority: P2 + Component: mod_sqlpw + AssignedTo: proftpd−devel@proftpd.org + ReportedBy: mgrabenstein@mac.com + URL: + Summary: Atempts to log in after the first failure always fail with mod_sqlpw (mysql) + + When I ftp to my host, if I type in the user id and password the first time + everything is fine and I am able to log in. + If I make a typo in the password and the log in fails. I can use the "user" + command in the ftp client to attempt logging in again. Problem is now, that no + matter what I type it will fail. + Work around: Get it right the first time, in other words: if you fail the first + time, quit ftp and re−start your ftp client. + This was noticed when ftp'ing from RedHat v6.1 and from Solaris 2.6 to my RedHat + v6.1 box...

http://bugs.proftpd.org/show_bug.cgi?id=105 *** shadow/105 Thu Apr 6 11:33:59 2000 −−− shadow/105.tmp.6990 Thu Apr 6 11:50:29 2000 *************** *** 11,17 **** AssignedTo: proftpd−devel@proftpd.org ReportedBy: mgrabenstein@mac.com URL: ! Summary: DefaultRoot does not work as advertised with mod_sqlpw OtherBugsDependingOnThis: 52[NEW] DefaultRoot does not work as advertised. It will now −−− 11,17 −−−− AssignedTo: proftpd−devel@proftpd.org ReportedBy: mgrabenstein@mac.com URL: ! Summary: DefaultRoot does not work as advertised with mod_sqlpw (mysql) OtherBugsDependingOnThis: 52[NEW] DefaultRoot does not work as advertised. It will now http://bugs.proftpd.org/show_bug.cgi?id=105 *** shadow/105 Thu Apr 6 11:33:37 2000 −−− shadow/105.tmp.6930 Thu Apr 6 11:33:37 2000 *************** *** 0 **** −−− 1,21 −−−− + Bug#: 105 + Product: ProFTPD + Version: 1.2.0pre10 + Platform: PC + OS/Version: Linux + Status: NEW + Resolution: + Severity: major + Priority: P2 + Component: mod_sqlpw + AssignedTo: proftpd−devel@proftpd.org + ReportedBy: mgrabenstein@mac.com + URL: + Summary: DefaultRoot does not work as advertised with mod_sqlpw + + DefaultRoot does not work as advertised. It will now + chroot in 1.2pre10, but the group expression field does not work. Specifing + something can disable DefaultRoot. (DefaultRoot ~ !mygroup == is not respected, + but Default root does chroot, where as DefaultRoot ~ ! mygroup == (with a space + between ! and group) seems to disable chroot.) Also DefaultRoot ~ does work (if + the group expression is blank).

http://bugs.proftpd.org/show_bug.cgi?id=108 *** shadow/108 Thu Apr 6 11:57:28 2000 −−− shadow/108.tmp.7017 Thu Apr 6 11:57:28 2000 *************** *** 0 **** −−− 1,19 −−−− + Bug#: 108 + Product: ProFTPD + Version: 1.2.0pre10 + Platform: PC + OS/Version: Linux + Status: NEW + Resolution: + Severity: normal + Priority: P2 + Component: mod_sqlpw + AssignedTo: proftpd−devel@proftpd.org + ReportedBy: mgrabenstein@mac.com + URL: + Summary: reference to flags.c needs to be removed to compile mod_sqlpw (mysql) + + To get mod_sqlpw to compile. You must first edit the Make.modules file and + remove the references to flags.c. + Or is it missing from the pre10 distribution ? mod_sqlpw seems work fine with ou + it. :−)

http://bugs.proftpd.org/show_bug.cgi?id=106 *** shadow/106 Thu Apr 6 11:50:13 2000 −−− shadow/106.tmp.6979 Thu Apr 6 11:50:13 2000 *************** *** 0 **** −−− 1,19 −−−− + Bug#: 106 + Product: ProFTPD + Version: 1.2.0pre10 + Platform: PC + OS/Version: Linux + Status: NEW + Resolution: + Severity: normal + Priority: P2 + Component: mod_sqlpw + AssignedTo: proftpd−devel@proftpd.org + ReportedBy: mgrabenstein@mac.com + URL: + Summary: inetd not working with mod_sqlpw (mysql) + + Using inetd to spawn proftpd with mod_sqlpw loaded resulted in: + "421 Service not available, remote server has closed connection" + Changing the proftpd.conf so the server was "standalone" and disabling ftp in + inetd.conf. Allows me to start a working server.

http://bugs.proftpd.org/show_bug.cgi?id=105 *** shadow/105 Thu Apr 6 11:33:37 2000 −−− shadow/105.tmp.6941 Thu Apr 6 11:33:59 2000 *************** *** 12,17 **** −−− 12,18 −−−− ReportedBy: mgrabenstein@mac.com URL: Summary: DefaultRoot does not work as advertised with mod_sqlpw + OtherBugsDependingOnThis: 52[NEW] DefaultRoot does not work as advertised. It will now chroot in 1.2pre10, but the group expression field does not work. Specifing

http://bugs.proftpd.org/show_bug.cgi?id=52 *** shadow/52 Tue Feb 15 14:26:57 2000 −−− shadow/52.tmp.6948 Thu Apr 6 11:34:00 2000 *************** *** 1,6 **** Bug#: 52 Product: ProFTPD ! Version: 1.2.0preX Platform: All OS/Version: All Status: NEW −−− 1,6 −−−− Bug#: 52 Product: ProFTPD ! Version: 1.2.0pre9 Platform: All OS/Version: All Status: NEW *************** *** 12,17 **** −−− 12,18 −−−− ReportedBy: wigstah@akitanet.co.uk URL: Summary: chroot'ing to user's home directory doesn't work when mod_mysql is in use + BugsThisDependsOn: 105[NEW] Hi,

+−−−−−− "Mark Renouf" wrote (Fri, 7−Jan−00, 18:21 −0500): | | ns1:/usr/local/mysql/var# /usr/sbin/proftpd −−version | ProFTPD Version 1.2.0pre8 | | I'm running with mod_sqlpw and mod_mysql | | wtmp is logging ftp logins VERY incorrectly | | example: a login by=20 | user: ftpuser=20 | password: mypassword | | [should show:] | | ns1:/usr/local/mysql/var# last −1 | ftpuser ftp nrwc−sh7−port89 Fri Jan 7 12:36 − 12:51 = | (00:15) | | [but instead shows like this:] | | ns1:/usr/local/mysql/var# last −1 | mypassword ftp nrwc−sh7−port89 Fri Jan 7 12:36 − 12:51 = | (00:15) | | Can anyone confirm this? Should I try pre9 ? Is this related the problem previously reported last November? AFAIK, neither pre9 nor CVS (but I haven't looked in a few days) includes a fix. It sounds similar though slightly different. +−−−−−− "Charles Seeger" wrote (Thu, 30−Dec−99, 11:25 −0500): | | o Passwords are logged to wtmp by mod_sqlpw! |

http://www.proftpd.org/proftpd−l−archive/99−11/msg00212.html | http://www.proftpd.org/proftpd−l−archive/99−11/msg00216.html | http://www.proftpd.org/proftpd−l−archive/99−11/msg00217.html | http://www.proftpd.org/proftpd−l−archive/99−11/msg00221.html | (includes a fix to mod_sqlpw.c:_checkpass function to prevent | logging the correct and wrong password in debug mode) | http://www.proftpd.org/proftpd−l−archive/99−11/msg00235.html | http://www.proftpd.org/proftpd−l−archive/99−11/msg00242.html Looks like it bears further investigation.

Uhmmmm Yes, the problem remains in pre9... The workaround is to disable that type of logging.

http://www.iusb.edu/~awalton/pro−mysql.txt that should help. −Andy −−−−− Original Message −−−−− From: "Mitch Vincent" <mitch@venux.net> To: <proftpd@proftpd.org> Sent: Saturday, March 11, 2000 2:00 PM Subject: [ProFTPD] Compile error with pre 10 > modules/mod_sqlpw.o: In function `auth_cmd_getpwnam': > /usr/source/proftpd−1.2.0pre10/modules/mod_sqlpw.c(.text+0x44a): undefined > reference to `mysql_escape_string' > modules/mod_sqlpw.o: In function `auth_cmd_auth': > /usr/source/proftpd−1.2.0pre10/modules/mod_sqlpw.c(.text+0x76c): undefined > reference to `mysql_escape_string' > *** Error code 1 > > > I'm trying to compile in mod_mysql and mod_sqlpw, however I get the above > error... Also, I couldn't find any documentation on the table layouts and > such require for mod_sqlpw and mod_mysql to work, could anyone point me to > it? >

---

proftpd.timeouts

I'm running ProFTPD 1.2.0pre10 One of the people using our FTP server is reporting problems connecting to it − I think the problems are related to timeouts. Can I ask if anyone else has had timeout problems with particular clients connecting to a proftpd server? I have set: TimeoutIdle 0 TimeoutStalled 0 TimeoutNoTransfer 3600 Yes − I do know that these may not be wise choices. (Sorry to be so vague − if I had a better handle on this I would be more precise). John Hearns

>>>>> "John" == John Hearns <john.hearns@framestore.co.uk> writes: John> I'm running ProFTPD 1.2.0pre10 One of the people using our John> FTP server is reporting problems connecting to it − I think John> the problems are related to timeouts. What problems exactly and from which FTP client (I ask because we have had many problems with IE5) ? John> Can I ask if anyone else has had timeout problems with John> particular clients connecting to a proftpd server? John> I have set: TimeoutIdle 0 TimeoutStalled 0 TimeoutNoTransfer John> 3600 John> Yes − I do know that these may not be wise choices. John> (Sorry to be so vague − if I had a better handle on this I John> would be more precise). John> John Hearns I would suggest tracing this client using the logs, perhaps setting up detailed logs of all the commands they send ?

Chapter 21. Compatibility and Integration

SQL

Authentication and persistant ratio support for the mod_ratio module are provided using SQL databases. The official documentation for this feature is currently a little thin on the ground. At the moment unless SQL support is provided for mod_ratio the ratios are only considered within a single connection with no persistance of credits recorded.

Compilation and support

To include support for sql the appropriate module has to be added prior to building the binary for the host system

```
./configure --with-module=mod_sqlpw:mod_mysql
make
make install
```

This should ensure that support is properly enabled, in addiiton to this a local MySQL (or similar) server should be installed and configured with the appropriate accesses and tables for yor setup. This is covered in later sections of this chapter.

Format of SQL tables

Example 21-1.

```
mysql> show fields from proftp;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(30) | YES  |     | NULL    |       |
| uid      | int(11)     | YES  |     | NULL    |       |
| gid      | int(11)     | YES  |     | NULL    |       |
| password | varchar(30) | YES  |     | NULL    |       |
| homedir  | varchar(50) | YES  |     | NULL    |       |
| count    | int(11)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

Example 21-2. Contents

```
mysql> select * from proftp;
+----------+------+------+----------+----------+-------+
| username | uid  | gid  | password | homedir  | count |
+----------+------+------+----------+----------+-------+
| oli      |  500 |  500 | test     | /home/om |     2 |
| oli2     |  500 |  500 | test     | /        |     1 |
+----------+------+------+----------+----------+-------+
```

(take care : uid and gid must be > 500. or change the source code of the module).

Authentication and persistant ratio support for the mod_ratio module are provided using SQL databases. The official documentation for this feature is currently a little thin on the ground. At the moment unless SQL support is provided for mod_ratio the ratios are only considered within a single connection with no persistance of credits recorded.

Compilation and support

To include support for sql the appropriate module has to be added prior to building the binary for the host system

```
./configure --with-module=mod_mysql
make
make install
```

This should ensure that support is properly enabled, in addiiton to this a local MySQL (or similar) server should be installed and configured with the appropriate accesses and tables for yor setup. This is covered in later sections of this chapter.

SQL Authentication

o Install MySQL o Compile Proftpd with the −−with−modules=mod_sqlpw:mod_mysql flags

Note: I had to alter the path slighly so the modules got mysql.h from the rightplace.

Detailing how to use MySQL is outside the scope of this document, so here's some links.

o http://www.devshed.com/Server_Side/MySQL/Administration/ o http://www.devshed.com/Server_Side/MySQL/Intro/

Quick rundown of what's needed to make a databae

o create a user for proftpd to access the database as o create permissions for this user o create new database (mine is called proftpd) o reload as required to make this live o create a table within proftpd (mine is ftp)

Example 21−3. SQL database layout

```
mysql> use proftpd;
Database changed
mysql> show tables;
+-------------------+
| Tables in proftpd |
+-------------------+
| ftp               |
+-------------------+
1 row in set (0.02 sec)

mysql> show columns from ftp ;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(60) | YES  |     | NULL    |       |
| uid      | int(11)     | YES  |     | NULL    |       |
| gid      | int(11)     | YES  |     | NULL    |       |
```

```
| password | varchar(30) | YES  |     | NULL    |       |
| homedir  | varchar(50) | YES  |     | NULL    |       |
| count    | int(11)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Example 21−4. Configuration fragment for SQL

```
--[ proftpd.conf ]--
# auth using mysql              host      login   pass    db
MySQLInfo                       localhost hamster *****   proftpd
SQLUserTable                    ftp
SQLUsernameField                username
SQLUidField                     uid
SQLGidField                     gid
SQLPasswordField                password
SQLHomedirField                 homedir
SQLLoginCountField              count
SQLAuthoritative                on
SQLPlaintextPasswords           on
--[ proftpd.conf ]--
```

Gotcha's

421 Service not availible Make sure that the home directory of the user concerned actually exists and has the right ownerships/permissions Can't connect to the database Is it running? Is it listening? Does the user proftpd is using have the right permissions?

Format of SQL tables

Example 21−5.

```
mysql> show fields from proftp;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(30) | YES  |     | NULL    |       |
| uid      | int(11)     | YES  |     | NULL    |       |
| gid      | int(11)     | YES  |     | NULL    |       |
| password | varchar(30) | YES  |     | NULL    |       |
| homedir  | varchar(50) | YES  |     | NULL    |       |
| count    | int(11)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

Example 21−6. Contents

```
mysql> select * from proftp;
+----------+------+------+----------+----------+-------+
| username | uid  | gid  | password | homedir  | count |
+----------+------+------+----------+----------+-------+
| oli      |  500 |  500 | test     | /home/om |     2 |
| oli2     |  500 |  500 | test     | /        |     1 |
+----------+------+------+----------+----------+-------+
```

(take care : uid and gid must be > 500. or change the source code of the module).

Configuration details

The following configuration is needed in the proftpd.conf file to enable sql support

Example 21−7. proftpd.conf

```
MySQLInfo                       localhost test "" test
                                # HOST login password database
MySQLUserTable                  proftp
MySQLUsernameField              username
MySQLUidField                   uid
MySQLGidField                   gid
MySQLPasswordField              password
MySQLHomedirField               homedir
MySQLLoginCountField            count
MySQLAuthoritative              on
MySQLPlaintextPasswords         on
```

SQL Logging

Example 21−8. Updated authentication table

```
mysql> show columns from ftpusers;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| username | varchar(60)  | YES  |     | NULL    |       |
| uid      | int(11)      | YES  |     | NULL    |       |
| gid      | int(11)      | YES  |     | NULL    |       |
| password | varchar(30)  | YES  |     | NULL    |       |
| homedir  | varchar(50)  | YES  |     | NULL    |       |
| count    | int(11)      | YES  |     | NULL    |       |
| fretr    | int(10)      | YES  |     | NULL    |       |
| bretr    | int(10)      | YES  |     | NULL    |       |
| bstor    | int(10)      | YES  |     | NULL    |       |
| fstor    | int(10)      | YES  |     | NULL    |       |
| ftime    | timestamp(14)| YES  |     | NULL    |       |
| faddr    | varchar(255) | YES  |     | NULL    |       |
| fhost    | varchar(255) | YES  |     | NULL    |       |
| fcdir    | varchar(255) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
14 rows in set (0.01 sec)
```

Example 21−9. File tracking table

```
mysql> show columns from logging2;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| fstor    | int(11)     | YES  |     | NULL    |       |
| fretr    | int(11)     | YES  |     | NULL    |       |
| bstor    | int(11)     | YES  |     | NULL    |       |
```

```
| bretr    | int(11)      | YES  |     | NULL    |       |
| fcdir    | varchar(255) | YES  |     | NULL    |       |
| fhost    | varchar(255) | YES  |     | NULL    |       |
| faddr    | varchar(255) | YES  |     | NULL    |       |
| ftime    | varchar(255) | YES  |     | NULL    |       |
| count    | int(11)      | YES  |     | NULL    |       |
| filename | varchar(255) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
10 rows in set (0.01 sec)
```

There's definately some cruft in the logging2 table which I need to clean out but I thought I'd make this post first >:) # auth using mysql host login pass db MySQLInfo bat.vom.tm hamster Ma3ros proftpd # SQLUserTable ftpusers SQLUsernameField username SQLUidField uid SQLGidField gid SQLPasswordField password SQLHomedirField homedir SQLLoginCountField count # # SQL Logging # SQLLogStats on # SQLLogHits "requires a table or table plus 3 fields: " "[table] filename count dir" SQLLogHits logging2 # SQLLogDirs fcdir SQLLogDirs fcdir # SQLLogHosts <host> <IP> <time> SQLLogHosts fhost faddr ftime Which results in authentication happening from the ftpusers table, and running totals of the number of files up/download and the byte counts. fcdir appears to hold the last directory change made (not sure what use it is...) and fhost, faddr, ftime appear to hold details of the last person to connect. logging2 holds a list of files downloaded and the number of times they have been collected. Notes: the logging table only works properly if it's pre−populated with filenames ie insert into logging2 (filename) values ('/full/dir/fromroot/filename'); Also with both tables the counters don't appear to work properly unless zeroed before use. Will ponder on this.

Hints

I'm trying to build Proftp pre8 from the FreeBSD ports collection with mod_mysql and having some troubles. The port's Makefile uses only the mod_ratio module by default. I thought I'd be able to build it with mod_mysql just by adding −−with−module=mod_mysql into the Makefile, but did not meet with success. Checking out the unpacked Proftpd I see links in the modules dir pointing to mod_ratio and mod_pgsql, so I tried it again with mod_pgsql instead. In both instances only the mod_ratio module was found and got made. I searched the archives and gave Johnnie's advice a go...

−with−modles=mod_sqlpw:mod_mysql:mod_pgsql:mod_ratio

Same story. So what am I missing here (besides a few brain cells)? Anybody build the FreeBSD port with mod_mysql module? Oh yeah, this is on a 3.3 box.

It doesn't work "out−of−the CVS" on my system (where mysql is installed in /usr/local/mysql). Isn't there an option on ./configure to tell where the files really are ? Currently, here are the "tricks" I'm using to make proftpd compile (using 24oct99 CVS version) :

...

./Make.rules Replaced LIBS=−lsupp −ldl −lcrypt −lm −lmysqlclient −lpam by LIBS=−lsupp −ldl −lcrypt −lm −lm /usr/local/mysql/lib/mysql/libmysqlclient.a −lpam ./modules/mod_mysql.c and ./modules/mod_sqlpw.c ===================== done an ln −s of these files from ./contrib to ./modules and replaced in _both_ files : #include <mysql.h> by #include "/usr/local/mysql/include/mysql/mysql.h" ./modules/Makefile ================== Removed the line mod_mysql.o: mod_mysql.h (there are no mod_mysql.h anymore) Finally, I compiled the whole by : ./configure −−with−modules=mod_sqlpw:mod_mysql −−prefix=/usr/local make make install Results : Oct 24 22:23:53

omega proftpd[7415]: omega.omnis.ch – ProFTPD 1.2.0pre8 standalone mode STARTUP I think it would be nice to correct ./modules/Makefile in the CVS, and why not to add symlinks from ./contrib to ./modules ?

Configuration details

The following configuration is needed in the proftpd.conf file to enable sql support

Example 21−10. proftpd.conf

```
MySQLInfo                      localhost test "" test
                               # HOST login password database
MySQLUserTable                 proftp
MySQLUsernameField             username
MySQLUidField                  uid
MySQLGidField                  gid
MySQLPasswordField             password
MySQLHomedirField              homedir
MySQLLoginCountField           count
MySQLAuthoritative             on
MySQLPlaintextPasswords        on
```

Hints

Hello: I'm trying to build Proftp pre8 from the FreeBSD ports collection with mod_mysql and having some troubles. The port's Makefile uses only the mod_ratio module by default. I thought I'd be able to build it with mod_mysql just by adding −−with−module=mod_mysql into the Makefile, but did not meet with success. Checking out the unpacked Proftpd I see links in the modules dir pointing to mod_ratio and mod_pgsql, so I tried it again with mod_pgsql instead. In both instances only the mod_ratio module was found and got made. I searched the archives and gave Johnnie's advice a go... −with−modles=mod_sqlpw:mod_mysql:mod_pgsql:mod_ratio Same story. So what am I missing here (besides a few brain cells)? Anybody build the FreeBSD port with mod_mysql module? Oh yeah, this is on a 3.3 box. Thanks bunches−−Ken It doesn't work "out−of−the CVS" on my system (where mysql is installed in /usr/local/mysql). Isn't there an option on ./configure to tell where the files really are ? Currently, here are the "tricks" I'm using to make proftpd compile (using 24oct99 CVS version) : ./Make.rules ============ Replaced LIBS=−lsupp −ldl −lcrypt −lm −lmysqlclient −lpam by LIBS=−lsupp −ldl −lcrypt −lm −lm /usr/local/mysql/lib/mysql/libmysqlclient.a −lpam ./modules/mod_mysql.c and ./modules/mod_sqlpw.c ==================== done an ln −s of these files from ./contrib to ./modules and replaced in _both_ files : #include <mysql.h> by #include "/usr/local/mysql/include/mysql/mysql.h" ./modules/Makefile ================== Removed the line mod_mysql.o: mod_mysql.h (there are no mod_mysql.h anymore) Finally, I compiled the whole by : ./configure −−with−modules=mod_sqlpw:mod_mysql −−prefix=/usr/local make make install Results : Oct 24 22:23:53 omega proftpd[7415]: omega.omnis.ch – ProFTPD 1.2.0pre8 standalone mode STARTUP I think it would be nice to correct ./modules/Makefile in the CVS, and why not to add symlinks from ./contrib to ./modules ?

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()

There are two patches available for runtime detection of sendfile() which gets round the 2.0.x problems.

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

What are these log lines in pre8?

The pre8 code has some additional debug logging going on tracking how sendfile is working. Nothing to get excited about it's probably a case of MacGyver forgetting to comment it out.

Regular expressions

ProFTPD uses POSIX−style regexps.

Chapter 22. Cookbook

Sod all here...

V. References

Table of Contents

I. Configuration Directives

This is a list of all the configuration directives

... FIX ME ...

Table of Contents

AccessDenyMsg

Name

AccessDenyMsg –– Customise the response on failed authentication

Synopsis

AccessDenyMsg [`"message"`]

`Default`
: Dependent on login type

`Context`
: server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
: mod_core

`Compatibility`
: 1.2.2 and later

Description

Normally, a 530 response message is sent to an FTP client immediately after a failed authentication attempt, with a standard message indicating the the reason of failure. In the case of a wrong password, the reason is usually "Login incorrect." It is this message can be customized with the AccessDenyMsg directive. In the message argument, the magic cookie '%u' is replaced with the username specified by the client during login.

See also

Examples

AccessDenyMsg "Guest access denied for %u."

AccessGrantMsg

Name

AccessGrantMsg –– Customise the response on successful authentication

Synopsis

AccessGrantMsg [`"message"`]

`Default`
: Dependent on login type

`Context`
: server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
: mod_core

`Compatibility`
: 0.99.0pl5 and later

Description

Normally, a 230 response message is sent to an FTP client immediately after authentication, with a standard message indicating that the user has either logged in or that anonymous access has been granted. This message can be customized with the AccessGrantMsg directive. In the message argument, the magic cookie '%u' is replaced with the username specified by the client during login.

See also

Examples

AccessGrantMsg "Guest access granted for %u."

Allow

Name

Allow –– Access control directive

Synopsis

Allow [`["from"] "all"|"none"|host|network[,host|network[,...]]`]

`Default`
: Allow from all

`Context`
: <Limit>

`Module`
: mod_core

`Compatibility`
: 0.99.0pl6 and later

Description

The Allow directive is used inside a <Limit> context to explicitly specify which hosts and/or networks have access to the commands or operations being limited. Allow is typically used in conjunction with Order and Deny in order to create sophisticated (or perhaps not–so–sophisticated) access control rules. Allow takes an optional first argument; the keyword from. Using from is purely cosmetic. The remaining arguments are expected to be a list of hosts and networks which will be explicitly granted access. The magic keyword all can be used to indicate that all hosts will explicitly be granted access (analogous to the AllowAll directive, except with a lower priority). Additionally, the magic keyword none can be used to indicate that no hosts or networks will be explicitly granted access (although this does not prevent them from implicitly being granted access). If all or none is used, no other hosts or networks can be supplied. Host and network addresses can be specified by name or numeric address. For security reasons, it is recommended that all address information be supplied numerically. Relying solely on named addresses causes security to depend a great deal upon DNS servers which may themselves be vulnerable to attack or spoofing. Numeric addresses which specify an entire network should end in a trailing period (i.e. 10.0.0. for the entire 10.0.0 subnet). Named address which specify an entire network should begin with a trailing period (i.e. .proftpd.net for the entire proftpd.net domain).

See also

Allow Order Limit

Examples

```
<Limit LOGIN>
Order allow,deny
Allow from 128.44.26.,128.44.26.,myhost.mydomain.edu,.trusted-domain.org
```

```
Deny from all
</Limit>
```

AllowAll

Name

AllowAll –– Allow all clients

Synopsis

AllowAll [`AllowAll`]

`Default`
Default is to implicitly AllowAll, but not explicitly
`Context`
<Directory>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The AllowAll directive explicitly allows access to a <Directory>, <Anonymous> or <Limit> block. Although proftpd's default behavior is to allow access to a particular object, the default is an implicit allow. AllowAll creates an explicit allow, overriding any higher level denial directives.

See also

DenyAll

Examples

AllowChmod

Name

AllowChmod –– Enable the CHMOD command (deprecated)

Synopsis

AllowChmod [`on` | `off`]

`Default`
true
`Context`
server config, <Directory>, <Global>, <VirtualHost>, <Anonymous>, .ftpaccess
`Module`
mod_site
`Compatibility`
1.2.0rc1 and later –– Deprecated

Description

This directive is deprecated, please use >Limit SITE_CHMOD< instead.

AllowChmod allows control over whether the "SITE CHMOD" command is allowed to clients.

See also

Examples

AllowChmod false

AllowFilter

Name

AllowFilter –– Regular expression of command arguments to be accepted

Synopsis

AllowFilter [`regular-expression`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
1.2.0pre7 and later

Description

AllowFilter allows the configuration of a regular expression that must be matched for all command arguments sent to ProFTPD. It is extremely useful in controlling what characters may be sent in a command to ProFTPD, preventing some possible types of attacks against ProFTPD. The regular expression is applied against the arguments to the command sent by the client, so care must be taken when creating a proper regex. Commands that fail the regex match result in a "Forbidden command" error being returned to the client. If the regular–expression argument contains whitespace, it must be enclosed in quotes.

See also

DenyFilter

Examples

```
# Only allow commands containing alphanumeric characters and whitespace
AllowFilter "^[a-zA-Z0-9 ,]*$"
```

AllowForeignAddress

Name

AllowForeignAddress –– Control the use of the PORT command

Synopsis

AllowForeignAddress [`on`|`off`]

Default
: AllowForeignAddress off

Context
: server config, <VirtualHost>, <Anonymous>, <Global>

Module
: mod_core

Compatibility
: 1.1.7 and later

Description

Normally, proftpd disallows clients from using the ftp PORT command with anything other than their own address (the source address of the ftp control connection), as well as preventing the use of PORT to specify a low−numbered (< 1024) port. In either case, the client is sent an "Invalid port" error and a message is syslog'd indicating either "address mismatch" or "bounce attack". By enabling this directive, proftpd will allow clients to transmit foreign data connection addresses that do not match the client's address. This allows such tricks as permitting a client to transfer a file between two FTP servers without involving itself in the actual data connection. Generally it's considered a bad idea, security−wise, to permit this sort of thing. AllowForeignAddress only affects data connection addresses; not tcp ports. There is no way (and no valid reason) to allow a client to use a low−numbered port in its PORT command.

See also

Examples

AllowGroup

Name

AllowGroup –– Group based allow rules

Synopsis

AllowGroup [`group-expression`]

`Default`
None
`Context`
<Limit>
`Module`
mod_core
`Compatibility`
1.1.1 and later

Description

AllowGroup specifies a group–expression that is specifically permitted within the context of the <Limit> block it is applied to. group–expression has the same format as that used in DefaultRoot, in that it should contain a comma separated list of groups or "not" groups (by prefixing a group name with the `!' character) that are to be allowed access to the block. The expression is parsed as a boolean "and" list, meaning that ALL elements of the expression must evaluate to logically true in order for the explicit allow to apply.

See also

DenyGroup, DenyUser, AllowUser

Examples

Allow

Name

AllowLogSymlinks –– Permit logging to symlinked files

Synopsis

AllowLogSymlinks [`"on"` | `"off"`]

`Default`
AllowLogSymlinks off

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_log

`Compatibility`
1.2.2rc2 and later

Description

By default, the server will the path of any configured SystemLog, any configured TransferLogs, and any configured ExtendedLogs to see if they are symbolic links. If the paths are symbolic links, the server will refuse to log to that link unless explicitly configured to do so via this directive.

Security note:

Security note: this behaviour should not be allowed unless for a very good reason. By allowing the server to open symbolic links with its root privileges, you are allowing a potential symlink attack where the server could be tricked into overwriting arbitrary system files. You have been warned.

See also

Examples

```
AllowLogSymlinks on
```

AllowOverwrite

Name

AllowOverwrite –– Enable files to be overwritten

Synopsis

AllowOverwrite [`on`|`off`]

`Default`
AllowOverwrite off

`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>, .ftpaccess

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The AllowOverwrite directive permits newly transfered files to overwrite existing files. By default, ftp clients cannot overwrite existing files.

See also

Examples

AllowRetrieveRestart

Name

AllowRetrieveRestart –– Allow clients to resume downloads

Synopsis

AllowRetrieveRestart [`on` | `off`]

Default
: AllowRetrieveRestart on

Context
: server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>, .ftpaccess

Module
: mod_core

Compatibility
: 0.99.0 and later

Description

The AllowRetrieveRestart directive permits or denies clients from performing "restart" retrieve file transfers via the FTP REST command. By default this is enabled, so that clients may resume interrupted file transfers at a later time without losing previously collected data.

See also

AllowStoreRestart

Examples

AllowStoreRestart

Name

AllowStoreRestart –– Allow clients to resume uploads

Synopsis

AllowStoreRestart [`on`|`off`]

`Default`
AllowStoreRestart off

`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>, .ftpaccess

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The AllowStoreRestart directive permits or denies clients from "restarting" interrupted store file transfers (those sent from client to server). By default restarting (via the REST command) is not permitted when sending files to the server. Care should be taken to disallow anonymous ftp "incoming" transfers to be restarted, as this will allow clients to corrupt or increase the size of previously stored files (even if not their own).

The REST (Restart STOR) command is automatically blocked when HiddenStor is enabled, with the server returning a 501 error code to the client.

See also

AllowRetrieveRestart DeleteAbortedStores HiddenStor

Examples

AllowUser

Name

AllowUser –– User based allow rules

Synopsis

AllowUser [`user–expression`]

Default
: None

Context
: <Limit>

Module
: mod_core

Compatibility
: 1.1.7 and later

Description

AllowUser specifies a user–expression that is specifically permitted access within the context of the <Limit> block it is applied to. user–expression has a similar syntax as that used in AllowGroup, in that it should contain a comma delimited list of users or "not" users (by prefixing a user name with the `!' character) that are to be allowed access to the block. The expression is parsed as a boolean "and" list, meaning that ALL elements of the expression must evaluate to logically true in order to the explicit allow to apply.

See also

DenyUser DenyGroup AllowGroup

Examples

AnonRatio

Name

AnonRatio –– Ratio directive

Synopsis

AnonRatio [`foo1 foo2 foo3`]

`Default`
: None known

`Context`
: <Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
: mod_ratio

`Compatibility`
: at least 1.2.0 and later

Description

The AnonRatio directive ....

See also

AnonRatio

Examples

AnonRequirePassword

Name

AnonRequirePassword –– Make anonymous users supply a valid password

Synopsis

AnonRequirePassword [`on` | `off`]

`Default`
AnonRequirePassword off
`Context`
<Anonymous>
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

Normally, anonymous FTP logins do not require the client to authenticate themselves via the normal method of a transmitted cleartext password which is hashed and matched against an existing system user's password. Instead, anonymous logins are expected to enter their e–mail address when prompted for a password. Enabling the AnonRequirePassword directive requires anonymous logins to enter a valid password which must match the password of the user that the anonymous daemon runs as. However using AuthUsingAlias authentication can be matched against the password of the login username. This can be used to create "guest" accounts, which function exactly as normal anonymous logins do (and thus present a "chrooted" protected file system to the client), but require a valid password on the server's host system.

See also

AnonymousGroup AuthAliasOnly AuthUsingAlias

Examples

```
Example of a "guest" account configuration:
<Anonymous ~roger>
User roger
Group other
UserAlias proftpd roger
AnonRequirePassword on
# Deny write operations to all directories, underneath root-dir
# Default is to allow, so we don't need a <Limit> for read operations.
<Directory *>
<Limit WRITE>
```

```
DenyAll
</Limit>
</Directory>
# Deny all read/write operations in incoming. Because these are command-group
# limits, we can explicitly permit certain operations which will take precedence
# over our group limit.
<Directory incoming>
<Limit READ WRITE>
DenyAll
</Limit>
# The only command allowed in incoming is STOR (transfer file from client
to server)
<Limit STOR>
AllowAll
</Limit>
</Directory>
</Anonymous>
```

Anonymous

Name

Anonymous –– Define an anonymous server

Synopsis

Anonymous [`root-directory`]

`Default`
: None

`Context`
: server config,<VirtualHost>, <Global>

`Module`
: mod_core

`Compatibility`
: 0.99.0 and later

Description

The Anonymous configuration block is used to create an anonymous FTP login, and is terminated by a matching </Anonymous> directive. The root–directory parameters specifies which directory the daemon will first chdir to, and then chroot, immediately after login. Once the chroot operation successfully completes, higher level directories are no longer accessible to the running child daemon (and thus the logged in user). By default, proftpd assumes an anonymous login if the remote client attempts to login as the currently running user; unless the current user is root, in which case anonymous logins are not allowed regardless of the presence of an <Anonymous> block. To force anonymous logins to be bound to a user other than the current user, see the User and Group directives. In addition, if a User or Group directive is present in an <Anonymous> block, the daemon permanently switches to the specified uid/gid before chroot()ing. Normally, anonymous logins are not required to authenticate with a password, but are expected to enter a valid e–mail address in place of a normal password (which is logged). If this behavior is undesirable for a given <Anonymous> configuration block, it can be overridden via the AnonRequirePassword directive.

Note: Chroot()ed anonymous directories do not need to have supplemental system files in them, nor do they need to have any sort of specific directory structure. This is because proftpd is designed to acquire as much system information as possible before the chroot, and to leave open those files which are needed for normal operation and reside outside the new root directory.

See also

Examples

```
Example of a typical anonymous FTP configuration:
<Anonymous /home/ftp>
```

```
User ftp # After anonymous login, daemon runs as user ftp.
Group ftp # After anonymous login, daemon runs as group ftp.
UserAlias anonymous ftp # Client login as 'anonymous' is aliased to 'ftp'.
# Deny write operations to all directories, underneath root-dir
# Default is to allow, so we don't need a <Limit> for read operations.
<Directory *>
<Limit WRITE>
DenyAll
</Limit>
</Directory>
<Directory incoming>
<Limit READ WRITE>
DenyAll
</Limit>
<Limit STOR>
AllowAll
</Limit>
</Directory>
</Anonymous>
```

AnonymousGroup

Name

AnonymousGroup –– Treat group members as anonymous users

Synopsis

AnonymousGroup [`group−expression`]

`Default`
None
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_core
`Compatibility`
1.1.3 and later

Description

The AnonymousGroup directive specifies a group−expression to which all matching users will be considered anonymous logins. The group−expression argument is a boolean logically ANDed list of groups to which the user must be a member of (or non−member if the group name is prefixed with a `!' character). For more information on group−expressions see the DefaultRoot directive. If the authenticating user is matched by an AnonymousGroup directive, no valid password is required, and a special dynamic anonymous configuration is created, with the user's home directory as the default root directory. If a DefaultRoot directive also applies to the user, this directory is used instead of the user's home dir. Great care should be taken when using AnonymousGroup, as improper configuration can open up user home directories to full read/write access to the entire world.

See also

AuthAliasOnly AuthUsingAlias AnonRequirePassword DefaultRoot

Examples

AuthAliasOnly

Name

AuthAliasOnly −− Allow only aliased login names

Synopsis

AuthAliasOnly [`on` | `off`]

`Default`
AuthAliasOnly off
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
1.1.3 and later

Description

AuthAliasOnly restricts authentication to "aliased" logins only; i.e. those usernames provided by clients which are "mapped" to a real userid by the UserAlias directive. Turning AuthAliasOnly `on' in a particular context will cause proftpd to completely ignore all non−aliased logins for the entire context. If no contexts are available without AuthAliasOnly set to `on', proftpd rejects the client login and sends an appropriate message to syslog.

See also

AnonymousGroup AuthUsingAlias AnonRequirePassword UserAlias

Examples

AuthGroupFile

Name

AuthGroupFile −− Specify alternate group file

Synopsis

AuthGroupFile [`path`]

`Default`
None
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_unixpw
`Compatibility`
1.0.3/1.1.1 and later

Description

AuthGroupFile specifies an alternate groups file, having the same format as the system /etc/group file, and if specified is used during authentication and group lookups for directory/access control operations. The path argument should be the full path to the specified file. AuthGroupFile can be configured on a per−VirtualHost basis, so that virtual FTP servers can each have their own authentication database (most often used in conjunction with AuthUserFile).

Note that this file need not reside inside a chroot()ed directory structure for Anonymous or DefaultRoot logins, as it is held open for the duration of client connections.

See also

AuthUserFile

Examples

AuthPAM

Name

AuthPAM –– Enable/Disable PAM authentication

Synopsis

AuthPAM [`on`|`off`]

`Default`
on
`Context`
server config,<VirtualHost>, <Global>
`Module`
mod_pam
`Compatibility`
1.2.0rc1 and later

Description

This directive determines whether PAM is used as an authentication method by ProFTPD. Enabled by default to fit in with the design policy of using PAM as the primary authentication mechanism.

See also

Examples

AuthPAMAuthoritative

Name

AuthPAMAuthoritative –– Set whether PAM is the authoritive authentication scheme

Synopsis

AuthPAMAuthoritative [`on` | `off`]

`Default`
off
`Context`
server config,<VirtualHost>, <Global>
`Module`
mod_pam
`Compatibility`
1.2.0pre3 and later

Description

This directive allows you to control whether or not PAM is the ultimate authority on authentication. Setting this directive to on will cause authentication to fail if PAM authentication fails. The default setting, off, allows other modules and directives such as AuthUserFile and friends to authenticate users, should PAM authentication fail. If you are having problems with PAM and using other directives like AuthUserFile, set this directive to off.

See also

Examples

AuthPAMConfig

Name

AuthPAMConfig –– Select PAM service name

Synopsis

AuthPAMConfig [`service`]

Default
: ftp

Context
: server config,<VirtualHost>, <Global>

Module
: mod_pam

Compatibility
: 1.2.0rc1 and later

Description

This directive allows you to specify the PAM service name used in authentication. PAM allows you to specify a service name to use when authenticating. This allows you to configure different PAM service names to be used for different virtual hosts. The directive was renamed from PAMConfig post 1.2.0 pre10.

See also

Examples

```
# Virtual host foobar authenticates differently than the rest

AuthPAMConfig foobar

# This assumes, that you have a PAM service named foobar
# configured in your /etc/pam.conf file or /etc/pam.d directory.
```

AuthUserFile

Name

AuthUserFile −− Specify alternate passwd file

Synopsis

AuthUserFile [`path`]

`Default`
None
`Context`
server config,<VirtualHost>, <Global>
`Module`
mod_unixpw
`Compatibility`
1.0.3/1.1.1 and later

Description

AuthUserFile specifies an alternate passwd file, having the same format as the system /etc/passwd file, and if specified is used during authentication and user lookups for directory/access control operations. The path argument should be the full path to the specified file. AuthUserFile can be configured on a per−VirtualHost basis, so that virtual FTP servers can each have their own authentication database (most often used in conjunction with AuthGroupFile).

Note that this file need not reside inside a chroot()ed directory structure for Anonymous or DefaultRoot logins, as it is held open for the duration of client connections.

See also

AuthGroupFile

Examples

AuthUsingAlias

Name

AuthUsingAlias –– Authenticate via Alias–name instead of mapped username

Synopsis

AuthUsingAlias [`on` | `off`]

Default
: AuthUsingAlias off

Context
: <Anonymous>

Module
: mod_core

Compatibility
: 1.2.0pre9 and later

Description

AuthUsingAlias disables the resolving of mapped usernames for authentication purposes. For example, if you have mapped the username anonymous to the "real" user ftp, the password gets checked against the user "anonymous". When AuthUsingAlias is disabled, the checked username would be "ftp".

See also

AnonymousGroup AuthAliasOnly AnonRequirePassword

Examples

```
An example of an Anonymous configuration using
AuthUsingAlias
# Basic Read-Only Anonymous Configuration.
<Anonymous /home/ftp>
UserAlias             anonymous  nobody
UserAlias             ftp        nobody
AuthAliasOnly         on
<Limit WRITE>
DenyAll
</Limit>
</Anonymous>
# Give Full Read-Write Anonymous Access to certain users
<Anonymous /home/ftp>
AnonRequirePassword   on
AuthAliasOnly         on
AuthUsingAlias        on
# The list of authorized users.
```

```
# user/pass lookup is for each user, not password entry
# of server uid ('nobody' in this example).
UserAlias             fred       nobody
UserAlias             joe        nobody
<Limit ALL>
AllowAll
</Limit>
</Anonymous>
```

Bind

Name

Bind –– Bind the server or Virtualhost to a specific IP address

Synopsis

Bind [`IP address`]

`Default`
None
`Context`
server config, <VirtualHost>
`Module`
mod_core
`Compatibility`
1.1.6 and later

Description

The Bind directive allows additional IP addresses to be bound to a main or VirtualHost configuration. Multiple Bind directives can be used to bind multiple addresses. The address argument should be either a fully qualified domain name or a numeric dotted–quad IP address. Incoming connections destined to an additional address added by Bind are serviced by the context containing the directive. Additionally, if SocketBindTight is set to on, a specific listen connection is created for each additional address.

See also

Examples

ByteRatioErrMsg

Name

ByteRatioErrMsg –– Ratio directive

Synopsis

ByteRatioErrMsg [`foo1 foo2 foo3`]

`Default`
None known

`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
mod_ratio

`Compatibility`
at least 1.2.0 and later

Description

The ByteRatioErrMsg directive .... Example: ByteRatioErrMsg

See also

Examples

CDPath

Name

CDPath –– Sets "search paths" for the cd command

Synopsis

CDPath [`directory`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
1.2.0pre2 and later

Description

Adds an entry to a search path that is used when changing directories. For example: CDPath /home/public CDPath /var/devel This allows a user to cd into any directory directly under /home/public or /var/devel, provided they have the appropriate rights. So, if /home/public/proftpd exists, cd proftpd will bring the user to that directory, regardless of where they currently are in the directory tree.

See also

Examples

Class

Name

Class –– Definition statements for class based tracking

Synopsis

Class [`"name" limit|regex|ip value`]

`Default`
None

`Context`
server config, <VirtualHost>

`Module`
mod_core

`Compatibility`
1.2.0pre9 and later

Description

Controls class based access. Class base access allows each connecting IP to be classified into a separate class. Each class has its own maximum number of connections. limit sets the maximum number of connections (default is 100) for that class name, regex sets a hostname regex (POSIX) for inclusion in the class and ip sets an IP/netmask based inclusion.

See also

Examples

Classes on
Class local limit 100
Class default limit 10
Class local regex .*foo.com
Class local ip 172.16.1.0/24

This creates two classes, local and default, with local being everything in *.foo.com and 172.16.1.* combined.

Classes

Name

Classes –– Enable Class based connection tracking

Synopsis

Classes [`on` | `off`]

`Default`
Off

`Context`
server config, <VirtualHost>

`Module`
mod_core

`Compatibility`
1.2.0pre9 and later

Description

Controls class based access. Enables class based access control. see: Class

See also

Examples

For examples, see Class

CommandBufferSize

Name

CommandBufferSize –– Limit the maximum command length

Synopsis

CommandBufferSize [`size`]

`Default`
None
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_core
`Compatibility`
1.2.0pre7 and later

Description

The CommandBufferSize directive controls the maximum command length permitted to be sent to the server. This allows you to effectively control what the longest command the server may accept it, and can help protect the server from various Denial of Service or resource–consumption attacks.

See also

Examples

CwdRatioMsg

Name

CwdRatioMsg –– Ratio directive

Synopsis

CwdRatioMsg [`foo1 foo2 foo3`]

`Default`
None known

`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
mod_ratio

`Compatibility`
at least 1.2.0 and later

Description

The CwdRatioMsg directive .... Example: CwdRatioMsg

See also

Examples

DefaultChdir

Name

DefaultChdir –– Set starting directory for FTP sessions

Synopsis

DefaultChdir [`directory [group-expression]]`

Default
: ~

Context
: server config, <VirtualHost>, <Anonymous>, <Global>

Module
: mod_auth

Compatibility
: 1.2.0pre2 and later

Description

Determines the directory a user is placed in after logging in. By default, the user is put in their home directory. The specified directory can be relative to the user's home directory. NOTE: if the specified directory is not available the user will not be able to log in.

See also

DefaultRoot

Examples

DefaultQuota

Name

DefaultQuota –– Sets the default quota

Synopsis

DefaultQuota [`value in bytes`]

`Default`
0
`Context`
server, <VirtualHost>, <Anonymous>
`Module`
mod_quota
`Compatibility`
at least 1.2.0 and later

Description

The DefaultQuota directive sets the default quota in bytes, this value is used if the .quota file does not exist.

See also

Quotas

Examples

```
#
# Set default to 1kb
#
DefaultQuota 1024
```

DefaultRoot

Name

DefaultRoot –– Sets default chroot directory

Synopsis

DefaultRoot [`directory [group-expression]]`

`Default`
DefaultRoot /
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_auth
`Compatibility`
0.99.0pl7 and later

Description

The DefaultRoot directive controls the default root directory assigned to a user upon login. If DefaultRoot is set to a directory other than "/", a chroot operation is performed immediately after a client authenticates. This can be used to effectively isolate the client from a portion of the host system filespace. The specified root directory must begin with a / or can be the magic character '~'; meaning that the client is chroot jailed into their home directory.

If the DefaultRoot directive specifies a directory which disallows access to the logged–in user's home directory, the user's current working directory after login is set to the DefaultRoot instead of their normal home directory. DefaultRoot cannot be used in <Anonymous> configuration blocks, as the <Anonymous> directive explicitly contains a root directory used for Anonymous logins. The special character '~' is replaced with the authenticating user's home directory immediately after login. Note that the default root may be a subdirectory of the home directory, such as "~/anon–ftp".

The optional group–expression argument can be used to restrict the DefaultRoot directive to a unix group, groups or subset of groups. The expression takes the format: [!]group–name1[,[!]group–name2[,...]]. The expression is parsed in a logical boolean AND fashion, such that each member of the expression must evaluate to logically TRUE in order for the DefaultRoot directive to apply. The special character '!' is used to negate group membership.

Care should be taken when using DefaultRoot. Chroot "jails" should not be used as methods for implementing general system security as there are potentially ways that a user can "escape" the jail.

See also

Examples

```
Example of a DefaultRoot configuration:
ServerName "A test ProFTPD Server"
ServerType inetd
User ftp
Group ftp
#
# This causes proftpd to perform a chroot into the authenticating user's directory
immediately after login.
# Once this happens, the user is unable to "see" higher level directories.
# Because a group-expression is included, only users who are a member of
# the group 'users' and NOT a member of 'staff' will have their default
# root directory set to '~'.
DefaultRoot ~ users,!staff
...
```

DefaultServer

Name

DefaultServer –– Set the default server

Synopsis

DefaultServer [`on`|`off`]

`Default`
DefaultServer off

`Context`
server config,<VirtualHost>

`Module`
mod_core

`Compatibility`
0.99.0pl6 and later

Description

The DefaultServer directive controls which server configuration is used as the default when an incoming connection is destined for an IP address which is neither the host's primary IP address or one of the addresses specified in a <VirtualHost> configuration block. Normally such "unknown" connections are issued a "no server available to service your request" message and disconnected. When DefaultServer is turned on for either the primary server configuration or a virtual server, all unknown destination connections are serviced by the default server. Only a single server configuration can be set to default.

See also

Examples

DefaultTransferMode

Name

DefaultTransferMode −− Set the default method of data transfer

Synopsis

DefaultTransferMode [`ascii` | `binary`]

`Default`
DefaultTransferMode ascii

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_core

`Compatibility`
1.2.0pre9 and later

Description

DefaultTransferMode sets the default transfer mode of the server. By default, carriage−return/linefeed translation will be performed (ASCII mode).

See also

Examples

DeferWelcome

Name

DeferWelcome −− Don't show welcome message until user has authenticated

Synopsis

DeferWelcome [`DeferWelcome on|off`]

`Default`
DeferWelcome off
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The DeferWelcome directive configures a master or virtual server to delay transmitting the ServerName and address to new connections, until a client has successfully authenticated. If enabled, the initial welcome message will be exceedingly generic and will not give away any type of information about the host that the daemon is actively running on. This can be used by security−conscious administrators to limit the amount of "probing" possible from non−trusted networks/hosts.

See also

ServerIdent ServerName

Examples

DeleteAbortedStores

Name

DeleteAbortedStores –– Enable automatic deletion of partially uploaded files

Synopsis

DeleteAbortedStores [`DeleteAbortedStores on|off`]

Default
: off

Context
: server, <VirtualHost>, <Directory>, <Anonymous>, <Global>, .ftpaccess

Module
: mod_core

Compatibility
: 1.2.0rc2 and later

Description

The DeleteAbortedStores directive controls whether ProFTPD deletes partially uploaded files if the transfer is stopped via the ABOR command rather than a connection failure.

See also

HiddenStor

Examples

Deny

Name

Deny -- Access control directive

Synopsis

Deny [`Deny ["from"] "all"|"none"|host|network[,host|network[,...]]]`

`Default`
None
`Context`
<Limit>
`Module`
mod_core
`Compatibility`
0.99.0pl6 and later

Description

The Deny directive is used to create a list of hosts and/or networks which will explicitly be denied access to a given <Limit> context block. The magic keywords all and none can be used to indicate that all hosts are denied access, or that no hosts are explicitly denied (respectively). For more information on the syntax and usage of Deny see: Allow and Order.

See also

Allow Order Limit

Examples

DenyAll

Name

DenyAll –– Deny all clients

Synopsis

DenyAll [`DenyAll`]

`Default`
: None

`Context`
: <Directory>, <Anonymous>, <Limit>, .ftpaccess

`Module`
: mod_core

`Compatibility`
: 0.99.0 and later

Description

The DenyAll directive is analogous to a combination of "order deny,allow <cr> deny from all", with the exception that it has a higher precendance when parsed. It is provided as a convenient method of completely denying access to a directory, anonymous ftp or limit block. Because of its precedence, it should not be intermixed with normal Order/Deny directives. The DenyAll directive can be overridden at a lower level directory by using AllowAll. DenyAll and AllowAll are mutually exclusive.

See also

AllowAll

Examples

DenyFilter

Name

DenyFilter –– Regular expression of command arguments to be blocked

Synopsis

DenyFilter [`DenyFilter regular-expression`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
1.2.0pre7 and later

Description

Similar to AllowFilter, DenyFilter specifies a regular expression which must not match any of the command arguments. If the regex does match, a "Forbidden command" error is returned to the client. This can be especially useful for forbidding certain command argument combinations from ever reaching ProFTPD.

Notes: The 'PASV' command cannot be blocked using this directive.

See also

AllowFilter

Examples

```
# We don't want to allow any commands with % being sent to the server
DenyFilter "%"
```

DenyGroup

Name

DenyGroup –– Group based deny rules

Synopsis

DenyGroup [`DenyGroup group-expression`]

`Default`
: None

`Context`
: <Limit>

`Module`
: mod_core

`Compatibility`
: 1.1.1 and later

Description

DenyGroup specifies a group–expression that is specifically denied within the context of the <Limit> block it is applied to. group–expression has the same format as that used in DefaultRoot, in that it should contain a comma separated list of groups or "not" groups (by prefixing a group name with the `!' character) that are to be denied access to the block. The expression is parsed as a boolean "and" list, meaning that ALL elements of the expression must evaluate to logically true in order for the explicit deny to apply.

See also

DenyUser, AllowUser AllowGroup

Examples

DenyUser

Name

DenyUser –– User based deny rules

Synopsis

DenyUser [`DenyUser user-expression`]

`Default`
None

`Context`
<Limit>

`Module`
mod_core

`Compatibility`
1.1.7 and later

Description

DenyUser specifies a user–expression that is specifically denied within the context of the <Limit> block it is applied to. user–expression is a comma delimited list of users or "not" users (by prefixing a user name with the `!' character). The expression is parsed as a boolean "and" list, meaning that all elements of the expression must evaluate to logically true in order for the explicit deny to apply.

See also

DenyGroup, AllowUser AllowGroup

Examples

Directory

Name

Directory –– FIXME FIXME

Synopsis

Directory [`<Directory pathname>`]

Default
: None

Context
: server config, <VirtualHost>, <Anonymous>, <Global>

Module
: mod_core

Compatibility
: 0.99.0 and later

Description

This directive creates a block of configuration directives which applies only to the specified directory and its sub–directories. The block is ended with </Directory>. Per–directory configuration is enabled during run–time with a "closest" match algorithm, meaning that the <Directory> directive with the closest matching path to the actual pathname of the file or directory in question is used. Per–directory configuration is inherited by all sub–directories until a closer matching <Directory> is encountered, at which time the original per–directory configuration is replaced with the closer match. Note that this does not apply to <Limit> </Limit> blocks, which are inherited by all sub–directories until a <Limit> block is reached in a closer match.

A trailing slash and wildcard ("/*") can be appended to the directory, specifying that the configuration block applies only to the contents (and sub–contents), not to the actual directory itself. Such wildcard matches always take precedence over non–wildcard <Directory> configuration blocks. <Directory> blocks cannot be nested (they are automatically nested at run–time based on their pathnames). Pathnames must always be absolute (except inside <Anonymous>), and should not reference symbolic links. Pathnames inside an <Anonymous> block can be relative, indicating that they are based on the anonymous root directory.

[Notes for ProFTPD 1.1.3 and later only] Pathnames that begin with the special character '~' and do not specify a username immediately after ~ are put into a special deferred mode. When in deferred mode, the directory context is not hashed and sorted into the configuration tree at boot time, but rather this hashing is deferred until a user authenticates, at which time the '~' character is replaced with the user's home directory. This allows a global <Directory> block which applies to all user's home directories, or sub–directories thereof.

See also

Limit

Examples

```
#Default usage of the directory directive
<Directory /users/robroy/private>
  HideNoAccess
</Directory>

#Example with username-expanding
<Directory ~/anon-ftp>
  <Limit WRITE>
    DenyAll
  </Limit>
</Directory>
```

DirFakeGroup

Name

DirFakeGroup –– Hide real file/directory group

Synopsis

DirFakeGroup [`DirFakeGroup On|Off [groupname]`]

`Default`
DirFakeGroup Off

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_ls

`Compatibility`
1.1.5

Description

DirFakeGroup can be used to hide the true group of files (including directories, fifos, etc.) in a directory listing. If simply turned On, DirFakeGroup will display all files as being owned by group 'ftp'. Optionally, the groupname argument can be used to specify a specific group other than 'ftp'. "~" can be used as the argument in order to display the primary group name of the current user.

Both DirFakeGroup and DirFakeUser are completely cosmetic; the groupname or username specified don't need to exist on the system, and neither directive affects permissions, real ownership or access control in any way.

See also

DirFakeUser DirFakeMode

Examples

DirFakeMode

Name

DirFakeMode –– Hide real file/directory permissions

Synopsis

DirFakeMode [`DirFakeMode octal-mode`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>

`Module`
mod_ls

`Compatibility`
1.1.6

Description

The DirFakeMode directive configures a mode (or permissions) which will be displayed for ALL files and directories in directory listings. For each subset of permissions (user, group, other), the "execute" permission for directories is added in listings if the "read" permission is specified by this directive. As with DirFakeUser, and DirFakeGroup, the "fake" permissions shown in directory listings are cosmetic only, they do not affect real permissions or access control in any way.

See also

DirFakeUser DirFakeGroup

Examples

```
  DirFakeMode 0640

Will result in:

  -rw-r----- ... arbitrary.file
  drwxr-x--- ... arbitrary.directory
```

DirFakeUser

Name

DirFakeUser –– Hide real file/directory owner

Synopsis

DirFakeUser [`DirFakeUser On|Off [username]`]

`Default`
DirFakeUser Off

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_ls

`Compatibility`
1.1.5

Description

DirFakeUser can be used to hide the true user owners of files (including directories, fifos, etc.) in a directory listing. If simply turned On, DirFakeUser will display all files as being owned by user 'ftp'. Optionally, the username argument can be used to specify a specific user other than 'ftp'. "~" can be used as the argument in order to display the current user's username.

Both DirFakeGroup and DirFakeUser are completely cosmetic; the groupname or username specified don't need to exist on the system, and neither directive affects permissions, real ownership or access control in any way.

See also

DirFakeGroup DirFakeMode

Examples

DisplayConnect

Name

DisplayConnect –– Sets connect banner file

Synopsis

DisplayConnect [`DisplayConnect filename`]

`Default`
None

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_code

`Compatibility`
1.2.0pre2 and later

Description

The DisplayConnect directive configures an ASCII text filename which will be displayed to the user when they initially connect but before they login. The filename can be either relative or absolute. In the case of a relative filename, the file is searched for starting in the home directory of the user the server is running as. As this can lead confusion, absolute pathnames are suggested. If the file cannot be found or accessed, no error occurs and nothing is logged or displayed to the client.

See also

Examples

DisplayFirstChdir

Name

DisplayFirstChdir –– FIXME FIXME

Synopsis

DisplayFirstChdir [`DisplayFirstChdir filename`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>

`Module`
mod_code

`Compatibility`
0.99.0 and later, magic cookies only in 0.99.0pl10 and later

Description

The DisplayFirstChdir directive configures an ASCII text filename which will be displayed to the user the first time they change into a directory (via CWD) per a given session. The file will also be displayed if proftpd detects that its last modification time has changed since the previous CWD into a given directory. If the filename is relative, it is looked for in the new directory that the user has changed into. Note that for anonymous ftp logins (see <Anonymous>), the file must reside inside the chroot()ed file system space. If the file cannot be found or accessed, no error occurs and nothing is logged or displayed to the client.

DisplayFirstChdir, DisplayConnect, DisplayLogin and DisplayQuit support the following "magic cookies" (only in 0.99.0pl10 and later), which are replaced with their respective strings before being displayed to the user.

```
%T      Current Time

%F      Available space on file system

%C      Current working directory

%R      Remote host name

%L      Local host name

%u      Username reported by ident protocol

%U      Username originally used in login

%M      Max number of connections

%N      Current number of connections

%E      Server admin's e-mail address
```

```
%x      The name of the user's class

%y      Current number of connections from the user's class

%z      Max number of connections from the user's class
```

NOTE: not all of these may have a rational value, depending on the context in which they're used (e.g., %u if ident lookups are off).

See also

DisplayConnect DisplayLogin DisplayQuit

Examples

DisplayGoAway

Name

DisplayGoAway –– FIXME FIXME

Synopsis

DisplayGoAway [`DisplayGoAway filename`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_core

`Compatibility`
1.2.0pre8 and later

Description

The DisplayGoAway directive specifies an ASCII text filename which will be displayed to the user if the class they're a member of has too many users logged in and their login request has been denied. DisplayGoAway supports the same "magic cookies" as DisplayFirstChdir.

See also

DisplayFirstChdir

Examples

DisplayLogin

Name

DisplayLogin –– FIXME FIXME

Synopsis

DisplayLogin [`DisplayLogin filename`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The DisplayLogin directive configures an ASCII text filename which will be displayed to the user when they initially login. The filename can be either relative or absolute. In the case of a relative filename, the file is searched for in the initial directory a user is placed in immediately after login (home directory for unix user logins, anonymous–root directory for anonymous logins). Note: that for jailed logins, the file must reside inside the chroot()ed file system space. If the file cannot be found or accessed, no error occurs and nothing is logged or displayed to the client. DisplayLogin supports the same "magic cookies" as DisplayFirstChdir.

See also

Examples

DisplayQuit

Name

DisplayQuit –– FIXME FIXME

Synopsis

DisplayQuit [`DisplayQuit filename`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_core

`Compatibility`
1.2.0pre8 and later

Description

DisplayQuit configures an ASCII text filename which will be displayed to the user when they quit. The filename can be either relative or absolute. In the case of a relative filename, the file is searched for in current directory a user is in when they logout –– for this reason, a absolute filename is usually preferable. NOTE: for jailed logins, the file must reside inside the chroot()ed file system space. If the file cannot be found or accessed, no error occurs and nothing is logged or displayed to the client. DisplayQuit supports the "magic cookies" listed under DisplayFirstChdir.

See also

Examples

DisplayReadme

Name

DisplayReadme –– FIXME FIXME

Synopsis

DisplayReadme [`DisplayReadme filename or pattern`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_readme

`Compatibility`
1.2.0pre8 and later

Description

Module: mod_readme The DisplayReadme directive notifies the user of the last change date of the specified file or pattern. Only a single DisplayReadme directive is allowed per configuration scope. DisplayReadme README Will result in: Please read the file README it was last modified on Sun Oct 17 10:36:14 1999 – 0 days ago Being displayed to the user on a cwd. DisplayReadmePattern README* Will result in: Please read the file README it was last modified on Tue Jan 25 04:47:48 2000 – 0 days ago Please read the file README.first it was last modified on Tue Jan 25 04:48:04 2000 – 0 days ago Being displayed to the user on a cwd.

See also

Examples

ExtendedLog

Name

ExtendedLog –– FIXME FIXME

Synopsis

ExtendedLog [`filename [[command-classes] format-nickname]]`

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous> <Global>

`Module`
mod_log

`Compatibility`
1.1.6pl1 and later

Description

The ExtendedLog directive allows customizable logfiles to be generated, either globally or per VirtualHost. The filename argument must contain an absolute pathname to a logfile which will be appended to when proftpd starts; the pathname should not be to a file in a nonexistent directory, to a world–writeable directory, or be a symbolic link (unless AllowLogSymlinks is set to on). Multiple logfiles (potentially with different command classes and formats) can be created. Optionally, the command–classes argument can be used to control which types of commands are logged. If not command classes are specified, proftpd logs all commands by default (passwords are hidden). command–classes is a comma delimited (no whitespace!) list of which commands to log.

The following are valid classes: NONE No commands AUTH Authentication commands (USER, PASS) INFO Informational commands (PWD, SYST, etc) DIRS Directory commands (LIST, CWD, MKD, etc) READ File reading (RETR) WRITE File/directory writing or creation MISC Miscellaneous commands (SITE, etc) ALL All commands (default)

If a format–nickname argument is supplied, ExtendedLog will use the predefined logformat (created by LogFormat). Otherwise, the default format of "%h %l %u %t \"%r\" %s %b" is used.

See also

AllowLogSymlinks, LogFormat, TransferLog

Examples

For example, to log all read and write operations to /var/log/ftp.log (using the default format), you could:

```
ExtendedLog /var/log/ftp.log read,write
```

FileRatioErrMsg

Name

FileRatioErrMsg –– FIXME FIXME

Synopsis

FileRatioErrMsg [`FileRatioErrMsg foo1 foo2 foo3`]

`Default`
None known

`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
mod_ratio

`Compatibility`
at least 1.2.0 and later

Description

The FileRatioErrMsg directive .... Example: FileRatioErrMsg

See also

Examples

FooBarDirective

Name

FooBarDirective –– FIXME FIXME

Synopsis

FooBarDirective [`FooBarDirective thingy`]

`Default`
none
`Context`
server config, <Anonymous>, <Limit>
`Module`
mod_sample
`Compatibility`
at least 1.2.0 and later

Description

FooBarDirective is a dummy directive to be used as a coding example only.

See also

Examples

Global

Name

Global –– FIXME FIXME

Synopsis

Global [`<Global>`]

`Default`
None
`Context`
server config, <VirtualHost>
`Module`
mod_core
`Compatibility`
1.1.6 and later

Description

The Global configuration block is used to create a set of configuration directives which is applied universally to both the main server configuration and all VirtualHost configurations. Most, but not all other directives can be used inside a Global block. In addition, multiple <Global> blocks can be created. At runtime, all Global blocks are merged together and finally into each server's configuration. Global blocks are terminated by a matching </Global> directive.

See also

Examples

Group

Name

Group –– FIXME FIXME

Synopsis

Group [`Group groupid`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The Group directive configures which group the server daemon will normally run at. See User for more details.

See also

Examples

GroupOwner

Name

GroupOwner –– FIXME FIXME

Synopsis

GroupOwner [`GroupOwner groupname`]

`Default`
: None

`Context`
: <Anonymous>, <Directory>, .ftpaccess

`Module`
: mod_core

`Compatibility`
: 0.99.0 and later

Description

The GroupOwner directive configures which group all newly created directories and files will be owned by, within the context that GroupOwner is applied to. The group ID of groupname cannot be 0. Note that GroupOwner cannot be used to override the host OS/file system user/group paradigm. If the current user is not a member of the specified group, new files and directories will not be able to be chown()ed to the GroupOwner group. If this happens, file STOR (send file from client to server) and MKD/XMKD (mkdir) operations will succeed normally, however the new directory entries will be owned by the current user's default group (a warning message is also logged) instead of by the desired group. If you also use UserOwner in the same context, this restriction is lifted.

See also

Examples

GroupPassword

Name

GroupPassword –– FIXME FIXME

Synopsis

GroupPassword [`GroupPassword groupid hashed-password`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
0.99.0pl5 and later

Description

The GroupPassword directive creates a special "group" password which allows all users in the specified group to authenticate using a single password. The group/password supplied is only effective inside the context to which GroupPassword is applied. The hashed–password argument is a standard cleartext password which has been passed through the standard unix crypt() library function. Extreme care should be taken when using GroupPassword, as serious security problems may arise if group membership is not carefully controlled.

See also

UserPassword

Examples

GroupRatio

Name

GroupRatio –– Ratio directive

Synopsis

GroupRatio [`GroupRatio foo1 foo2 foo3`]

`Default`
None known

`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
mod_ratio

`Compatibility`
at least 1.2.0 and later

Description

The GroupRatio directive .... Example: GroupRatio

See also

Examples

HiddenStor

Name

HiddenStor -- Enables more safe file uploads

Synopsis

HiddenStor [`HiddenStor on|off`]

Default
: HiddenStor off

Context
: <Directory>, <Anonymous>, <VirtualHost>, <Global>

Module
: mod_core

Compatibility
: 1.2.0pre5 and later

Description

The HiddenStor directive enables two-step file uploads: files are uploaded as ".in.filename." and once the upload is complete, renamed to just "filename". This provides a degree of atomicity and helps prevent 1) incomplete uploads and 2) files being used while they're still in the progress of being uploaded. Note: if the temporary file name is already in use (e.g., a server crash during upload), it will prevent the file from being uploaded.

The REST (Restart STOR) command is automatically blocked when HiddenStor is enabled, with the server returning a 501 error code to the client.

See also

AllowStoreRestart DeleteAbortedStores

Examples

HideGroup

Name

HideGroup –– FIXME FIXME

Synopsis

HideGroup [`HideGroup groupid`]

`Default`
None
`Context`
<Directory>, <Anonymous>
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The HideGroup directive configures a <Directory> or < Anonymous> block to hide all directory entries owned by the specified group, unless the group is the primary group of the currently logged–in, authenticated user . Normally, hidden directories and files cannot be seen via LIST or NLST commands but can be operated on via other FTP commands (CWD, DELE, RETR, etc). This behavior can be modified via the IgnoreHidden directive.

See also

See Also: HideUser, HideNoAccess, IgnoreHidden

Examples

HideNoAccess

Name

HideNoAccess –– Block the listing of directory entries to which the user has no access permissions

Synopsis

HideNoAccess [`HideNoAccess on|off`]

`Default`
None

`Context`
<Directory>,<Anonymous>

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The HideNoAccess directive configures a <Directory> or <Anonymous> block to hide all directory entries in a directory listing (via the LIST or NLST FTP commands) to which the current logged–in, authenticated user has no access to. Normal Unix–style permissions always apply, so that although a user may not be able to see a directory entry that has HideNoAccess applied, they will receive a normal "Permission denied" error message when attempting to blindly manipulate the file system object. The directory or file can be made completely invisible to all FTP commands by applying IgnoreHidden in conjunction with HideNoAccess.

See also

See Also: HideUser, HideGroup, IgnoreHidden

Examples

HideUser

Name

HideUser –– FIXME FIXME

Synopsis

HideUser [`HideUser userid`]

`Default`
None
`Context`
<Directory>, <Anonymous>
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The HideUser directive configures a <Directory> or <Anonymous> block to hide all directory entries owned by the specified user, unless the owning user is the currently logged–in, authenticated user. Normally, hidden directories and files cannot be seen via LIST or NLST commands but can be operated on via other FTP commands (CWD, DELE, RETR, etc). This behavior can be modified via the IgnoreHidden directive.

See also

HideGroup, HideNoAccess, IgnoreHidden

Examples

HostRatio

Name

HostRatio –– Ratio directive

Synopsis

HostRatio [`HostRatio foo1 foo2 foo3`]

`Default`
None known

`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
mod_ratio

`Compatibility`
at least 1.2.0 and later

Description

The HostRatio directive .... Example: HostRatio

See also

Examples

IdentLookups

Name

IdentLookups –– Toggle ident lookups

Synopsis

IdentLookups [`IdentLookups on|off`]

Default
: IdentLookups on

Context
: server config, <VirtualHost>, <Global>

Module
: mod_core

Compatibility
: 1.1.5 and later

Description

Normally, when a client initially connects to proftpd, the ident protocol (RFC1413) is used to attempt to identify the remote username. This can be controlled via the IdentLookups directive.

See also

Examples

IgnoreHidden

Name

IgnoreHidden –– Treat 'hidden' files as if they don't exist

Synopsis

IgnoreHidden [`IgnoreHidden on|off`]

Default
: IgnoreHidden off

Context
: <Limit>

Module
: mod_core

Compatibility
: 0.99.0 and later

Description

Normally, files hidden via HideNoAccess, HideUser or HideGroup can be operated on by all FTP commands (assuming Unix file permissions allow access), even though they do not appear in directory listings. Additionally, even when normal file system permissions disallow access, proftpd returns a "Permission denied" error to the client, indicating that the requested object does exist, even if it cannot be acted upon. IgnoreHidden configures a <Limit> block to completely ignore any hidden directory entries for the set of limited FTP commands. This has the effect of returning an error similar to "No such file or directory" when the client attempts to use the limited command upon a hidden directory or file.

See also

Examples

Include

Name

Include –– Load additional configuration directives from a file

Synopsis

Include [`Include file`]

`Default`
None
`Context`
server config, <Directory>, <Anonymous>, <VirtualHost>, <Global>
`Module`
mod_core
`Compatibility`
1.2.0 and later

Description

This directive allows you to include another configuration file within your current configuration file. The given file argument must be the full path to the file to be included.

See also

Examples

LDAPAuthBinds

Name

LDAPAuthBinds –– FIXME FIXME

Synopsis

Syntax: LDAPAuthBinds [`on` `off`]

FIX FIX FIX

`Default`
: LDAPAuthBinds off in mod_ldap <= 2.7.6, LDAPAuthBinds on in mod_ldap >= 2.8

`Context`
: server config, <VirtualHost>, <Global>

`Module`
: mod_ldap

`Compatibility`
: mod_ldap v2.5 and later

Description

By default, the DN specified by LDAPDNInfo will be used to bind to the LDAP server to obtain user information, including the userPassword attribute. If LDAPAuthBinds is set to on, the DN specified by LDAPDNInfo will be used to fetch all user information except the userPassword attribute. Then, mod_ldap will bind to the LDAP server as the user who is logging in via FTP with the user−supplied password. If this bind succeeds, the user is considered authenticated and is allowed to log in. This method of LDAP authentication has the added benefit of supporting any password encryption scheme that your LDAP server supports.

See also

Examples

LDAPDefaultAuthScheme

Name

LDAPDefaultAuthScheme —— Set the authentication scheme/hash that is used when no leading {hashname} is present.

Synopsis

LDAPDefaultAuthScheme [`crypt` `clear`]

`Default`
LDAPDefaultAuthScheme "crypt"

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_ldap

`Compatibility`
mod_ldap v2.0 and later

Description

Specifies the authentication scheme used for passwords with no {prefix} in the LDAP database. For example, if you are using something like userPassword: mypass in your LDAP database, you would want to set LDAPDefaultAuthScheme to clear.

See also

Examples

LDAPDefaultGID

Name

LDAPDefaultGID –– Set the default GID to be assigned to users when no uidNumber attribute is found.

Synopsis

LDAPDefaultGID [`default-gid`]

`Default`
None
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_ldap
`Compatibility`
mod_ldap v2.0 and later

Description

This directive is useful primarily in virtual–user environments common in large–scale ISPs and hosting organizations. If a user does not have a LDAP gidNumber attribute, the LDAPDefaultGID is used. This allows one to have a large number of users in an LDAP database without gidNumber attributes; setting this configuration directive will automatically assign those users a single GID.

See also

Examples

LDAPDefaultUID

Name

LDAPDefaultUID −− Set the default GID to be assigned to users when no uidNumber attribute is found.

Synopsis

LDAPDefaultUID [`default-uid`]

`Default`
None
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_ldap
`Compatibility`
mod_ldap v2.0 and later

Description

This directive is useful primarily in virtual−user environments common in large−scale ISPs and hosting organizations. If a user does not have a LDAP uidNumber attribute, the LDAPDefaultUID is used. This allows one to have a large number of users in an LDAP database without uidNumber attributes; setting this configuration directive will automatically assign those users a single UID.

See also

Examples

LDAPDNInfo

Name

LDAPDNInfo –– Set DN information to be used for initial bind

Synopsis

LDAPDNInfo [`LDAPDNInfo "ldap-dn" "dn-password"`]

`Default`
LDAPDNInfo "" "" (anonymous bind)

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_ldap

`Compatibility`
mod_ldap v2.0 and later

Description

This directive specifies the LDAP DN and password to use when binding to the LDAP server. If this configuration directive is not specified, anonymous binds are used.

See also

Examples

LDAPDoAuth

Name

LDAPDoAuth –– Enable LDAP authentication

Synopsis

LDAPDoAuth [`on off`] [`"auth-base-prefix"`] [`"search-filter-template"`]

Default
: LDAPDoAuth off

Context
: server config, <VirtualHost>, <Global>

Module
: mod_ldap

Compatibility
: mod_ldap v2.0 and later

Description

This configuration directive activates LDAP authentication. The second argument to this directive is the LDAP prefix to use for authentication. The third argument is a template to be used for the search filter; %u will be replaced with the username that is being authenticated. By default, the search filter template "(&(uid=%u)(objectclass=posixAccount))" is used. Search filter templates are only supported in mod_ldap v2.7 and later.

See also

Examples

LDAPDoGIDLookups

Name

LDAPDoGIDLookups −− Enable LDAP lookups for user group membership and GIDs in directory listings

Synopsis

LDAPDoGIDLookups [`on off`] [`"uid-base-prefix"`] [`"search-filter-template"`]

`Default`
LDAPDoGIDLookups off

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_ldap

`Compatibility`
mod_ldap v2.0 and later

Description

This configuration directive activates LDAP GID−to−name lookups in directory listings. The second argument to this directive is the LDAP prefix to use for GID−to−name lookups. The third argument is a template to be used for the search filter; %u will be replaced with the GID that is being looked up. By default, the search filter template "(&(gidNumber=%u)(objectclass=posixGroup))" is used. Search filter templates are only supported in mod_ldap v2.7 and later.

See also

Examples

LDAPDoUIDLookups

Name

LDAPDoUIDLookups −− Enable LDAP lookups for UIDs in directory listings

Synopsis

LDAPDoUIDLookups [`on` `off`] [`"search-filter-template"`] [`"uid-base-prefix"`]

`Default`
LDAPDoUIDLookups off

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_ldap

`Compatibility`
mod_ldap v2.0 and later

Description

This configuration directive activates LDAP UID−to−name lookups in directory listings. The second argument to this directive is the LDAP prefix to use for UID−to−name lookups. The third argument is a template to be used for the search filter; %u will be replaced with the UID that is being looked up. By default, the search filter template "(&(uidNumber=%u)(objectclass=posixAccount))" is used. Search filter templates are only supported in mod_ldap v2.7 and later.

See also

Examples

LDAPForceDefaultGID

Name

LDAPForceDefaultGID –– Force all LDAP–authenticated users to use the same GID.

Synopsis

Syntax: LDAPForceDefaultGID [`on` `off`]

Default
: LDAPForceDefaultGID off

Context
: server config, <VirtualHost>, <Global>

Module
: mod_ldap

Compatibility
: mod_ldap v2.8 and later

Description

Even when a LDAPDefaultGID is configured, mod_ldap will allow individual users to have gidNumber attributes that will override this default GID. With LDAPForceDefaultGID enabled, all LDAP–authenticated users are given the default GID; GIDs may not be overridden by gidNumber attributes.

See also

Examples

LDAPForceDefaultUID

Name

LDAPForceDefaultUID –– Force all LDAP–authenticated users to use the same UID.

Synopsis

Syntax: LDAPForceDefaultUID [`on` `off`]

`Default`
LDAPForceDefaultUID off
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_ldap
`Compatibility`
mod_ldap v2.8 and later

Description

Even when a LDAPDefaultUID is configured, mod_ldap will allow individual users to have uidNumber attributes that will override this default UID. With LDAPForceDefaultUID enabled, all LDAP–authenticated users are given the default UID; UIDs may not be overridden by uidNumber attributes.

See also

Examples

LDAPHomedirOnDemand

Name

LDAPHomedirOnDemand –– Enable the creation of user home directories on demand

Synopsis

LDAPHomedirOnDemand [`on` `off`] [`directory-mode`]

`Default`
LDAPHomedirOnDemand off
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_ldap
`Compatibility`
mod_ldap v2.0 and later

Description

LDAPHomedirOnDemand activates on–demand home directory creation. If a user logs in and does not yet have a home directory, a home directory is created automatically.

In mod_ldap <= 2.7.6, the home directory will be owned by the same user and group that ProFTPD runs as (see the User and Group configuration directives). mod_ldap >= 2.8 can create home directories for users with any UID/GID, not just those with the same UID/GID as the main ProFTPD server.

The second argument allows you to specify the mode (default permissions) to use when creating home directories on demand, subject to ProFTPD's umask (see the Umask directive). If no directory mode is specified, the default of 0755 is used. Directory mode setting is only supported in mod_ldap v2.7 or later.

See also

Examples

LDAPHomedirOnDemandPrefix

Name

LDAPHomedirOnDemandPrefix –– Enable the creation of user home directories on demand

Synopsis

LDAPHomedirOnDemandPrefix [`leading-path`]

`Default`
LDAPHomedirOnDemandPrefix off

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_ldap

`Compatibility`
mod_ldap v2.8 and later

Description

LDAPHomedirOnDemandPrefix enables a prefix to be specified for on–demand home directory creation. This is most useful if mod_ldap is being used to authenticate against an LDAP directory that does not return a homeDirectory attribute, either because it cannot (Microsoft Active Directory, for example) or because you do not wish to extend your existing directory schema.

For example, setting this directive to "/home" and logging in as the user "joe" would result in his home directory being created as "/home/joe". The directory will be created with the mode specified in LDAPHomedirOnDemand. To use this directive, LDAPHomedirOnDemand must be enabled.

See also

Examples

LDAPHomedirOnDemandPrefixNoUsername

Name

LDAPHomedirOnDemandPrefixNoUsername –– FIXFIXFIX

Synopsis

LDAPHomedirOnDemandPrefixNoUsername [`"name" limit|regex|ip value`]

Default
: FIXFIXFIX

Context
: server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

Module
: mod_ldap

Compatibility
: 1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

LDAPHomedirOnDemandSuffix

Name

LDAPHomedirOnDemandSuffix — Specify an additional directory to be created inside a user's home directory on demand.

Synopsis

LDAPHomedirOnDemandSuffix [`additional-directory1` `additional-directory2` `additional-directory3`]

Default
: LDAPHomedirOnDemandSuffix ""

Context
: server config, <VirtualHost>, <Global>

Module
: mod_ldap

Compatibility
: mod_ldap v2.6 and later.

Description

to be created within a user's home directory when it is created on demand. For example, if a user's home directory is "/home/user", setting this configuration directive to "public_html" will also create "/home/user/public_html" on demand. In mod_ldap v2.7.6 and earlier, you must also activate LDAPHomedirOnDemand in your configuration.

mod_ldap >= 2.8 supports multiple suffix arguments and does not require LDAPHomedirOnDemand to be enabled.

See also

Examples

LDAPNegativeCache

Name

LDAPNegativeCache –– Enable negative caching for LDAP lookups

Synopsis

LDAPNegativeCache [`on off`]

`Default`
LDAPNegativeCache off
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_ldap
`Compatibility`
mod_ldap v1.1 and later

Description

LDAPNegativeCache specifies whether or not to cache negative responses from the LDAP server when using LDAP for UID/GID lookups. This option is useful if you also use/are in transition from another authentication system; if there are many users in your old authentication system that aren't in the LDAP database, there can be a significant delay when a directory listing is performed as the UIDs not in the LDAP database are repeatedly looked up in an attempt to present usernames instead of UIDs in directory listings. With LDAPNegativeCache set to on, negative ("not found") responses from the LDAP server will be cached and speed will improve on directory listings that contain many users not present in the LDAP database.

See also

Examples

LDAPQueryTimeout

Name

LDAPQueryTimeout –– Set a timeout for LDAP queries

Synopsis

LDAPQueryTimeout [`timeout-seconds`]

`Default`
LDAPQueryTimeout default–api–timeout

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_ldap

`Compatibility`
mod_ldap v2.0 and later

Description

Sets the timeout used for LDAP directory queries. The default is the default timeout used by your LDAP API.

See also

Examples

LDAPSearchScope

Name

LDAPSearchScope –– Specify the search scope used in LDAP queries

Synopsis

LDAPSearchScope [`onelevel` `subtree`]

`Default`
LDAPSearchScope subtree
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_ldap
`Compatibility`
mod_ldap v2.6 and later

Description

Set the scope used for LDAP searches. The default setting, subtree, searches for all entries in the tree from the current level down. Setting this directive to onelevel searches only one level deep in the LDAP tree.

See also

Examples

LDAPServer

Name

LDAPServer –– Specify the LDAP server to use for lookups

Synopsis

LDAPServer [`"hostname1:port1 hostname2:port2"`]

`Default`
LDAPServer "localhost"

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_ldap

`Compatibility`
mod_ldap v1.0 and later

Description

LDAPServer allows you to to specify the hostname(s) and port(s) of the LDAP server(s) to use for LDAP authentication. If no LDAPServer configuration directive is present, the default LDAP servers specified by your LDAP API will be used.

See also

Examples

LDAPUseTLS

Name

LDAPUseTLS –– Enable TLS/SSL connections to the LDAP server.

Synopsis

Syntax: LDAPUseTLS [`on` `off`]

`Default`
LDAPUseTLS off
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_ldap
`Compatibility`
mod_ldap v2.8 and later

Description

By default, mod_ldap connects to the LDAP server via a non–encrypted connection. Enabling this option causes mod_ldap to use an encrypted (TLS/SSL) connection to the LDAP server. If a secure connection to the LDAP server fails, mod_ldap will not authenticate users (mod_ldap will *not* fall back to an unsecure connection).

See also

Examples

LeechRatioMsg

Name

LeechRatioMsg –– Sets the 'over ratio' error message

Synopsis

LeechRatioMsg [`LeechRatioMsg foo1 foo2 foo3`]

`Default`
None known

`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
mod_ratio

`Compatibility`
at least 1.2.0 and later

Description

The LeechRatioMsg directive defines the response message sent back to the client upon breaking their quota limits.

See also

Examples

```
LeechRatioMsg "please upload as well as download"
```

Limit

Name

Limit -- FIXME FIXME

Synopsis

Limit [`<Limit command|command-group [command2 ..]>`]

`Default`
: None

`Context`
: server config, <VirtualHost>, <Directory>, <Anonymous>, <Global>, .ftpaccess

`Module`
: mod_core

`Compatibility`
: 0.99.0 and later

Description

The Limit configuration block is used to place access restrictions on one or more FTP commands, within a given context. Limits flow downward, so that a Limit configuration in the server config context applies to all <Directory> and <Anonymous> blocks that also reside in the configuration; until it is overridden by a "lower" <Limit> block. Any number of command parameters can be specified, against which the contents of the <Limit> block will be applied. command can be any valid FTP command, but is generally one of the following: CWD (Change Working Directory) Sent by client when changing directories. MKD / XMKD (MaKe Directory) Sent by client to create a new directory. RNFR (ReName FRom), RNTO (ReName TO) Sent as a pair by client to rename a directory entry. DELE (DELEte) Sent by client to delete a file. RMD / XRMD (ReMove Directory) Sent by client to remove a directory. RETR (RETRieve) Transfer a file from the server to the client. STOR (STORe) Transfer a file from the client to the server. In addition, the following command-groups are accepted. They have a lower precedence than real commands, meaning that a real command limit will always be applied instead of the command-group. READ All FTP commands which deal with file reading (directory listing not included): RETR, SITE, SIZE, STAT WRITE All FTP commands which deal with file or directory write/creation/deletion: APPE, DELE, MKD, RMD, RNTO, STOR, XMKD, XRMD DIRS All FTP commands which deal with directory listing: CDUP, CWD, LIST, MDTM, NLST, PWD, RNFR, XCUP, XCWD, XPWD ALL ALL FTP commands (identical to READ WRITE DIRS). Note this group has the lowest precedence of all; it will not override a limit imposed by another command-group (e.g. DIRS). Finally, a special command is allowed which can be used to control login access: LOGIN Connection or login to the server. Applying a <Limit> to this pseudo-command can be used to allow or deny initial connection or login to the context. It has no effect, and is ignored, when used in a context other than server config, <VirtualHost> or <Anonymous> (i.e. using it in a <Directory> context is meaningless). <Limit> command restrictions should not be confused with file/directory access permission. While limits can be used to restrict a command on a certain directory, they cannot be used to override the file permissions inherent to the base operating/file system. The following FTP commands cannot be restricted via <Limit>: ABOR HELP MODE (not implemented, always S) NOOP PASS (use <Limit LOGIN>) PASV PORT QUIT REST (use AllowRetrieveRestart, AllowStoreRestart) STRU (not implemented, always F) SYST TYPE USER (use <Limit LOGIN>)

See also

See Also: IgnoreHidden

Examples

LogFormat

Name

LogFormat -- Specify a logging format

Synopsis

LogFormat [`LogFormat nickname "format-string"`]

`Default`
LogFormat default "%h %l %u %t \"%r\" %s %b"

`Context`
server config

`Module`
mod_log

`Compatibility`
1.1.6pl1 and later

Description

The LogFormat directive can be used to create a custom logging format for use with the ExtendedLog directive. Once created, the format can be referenced by the specified nickname. The format-string argument can consist of any combination of letters, numbers and symbols. The special character % is used to start a meta-sequence (see below). To insert a literal % character, use %%.

The following meta sequences are available and are replaced as indicated when logging. %A Anonymous username (password given), or UNKNOWN if non-anonymous %b Bytes sent for request %f Filename stored or retrieved, absolute path (not chrooted) %F Filename stored or retrieved, as the client sees it %{FOOBAR}e Contents of environment variable FOOBAR. Note that the server does not set any environment variables itself. %h Remote host name %a Remote IP address %l Remote username (from ident), or UNKNOWN if ident lookup failed %m Command (method) name received from client, e.g., RETR %p Local server port number %v Local server name %P Local server process id (pid) %r Full command line received from client %t Current local time %{format}t Current local time formatted (strftime(3) format) %T Time taken to transmit/receive file, in seconds %s Numeric FTP response code (status) %u Local authenticated userid

See also

ExtendedLog, TransferLog

Examples

LoginPasswordPrompt

Name

LoginPasswordPrompt –– FIXME FIXME

Synopsis

LoginPasswordPrompt [`LoginPasswordPrompt on|off`]

`Default`
LoginPasswordPrompt on
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_auth
`Compatibility`
1.2.0pre1 and later

Description

If set to off, ProFTPd will skip the password request if the login will be denied regardless of password, e.g., if a <Limit LOGIN> directive forbids the connection.

See also

Examples

LsDefaultOptions

Name

LsDefaultOptions –– FIXME FIXME

Synopsis

LsDefaultOptions [`LsDefaultOptions "options string"`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_ls

`Compatibility`
1.1.6 and later

Description

Normally, FTP commands involving directory listings (NLST, LIST and STAT) use the arguments (options) passed by the client to determine what files are displayed and the format they are displayed in. Using the LsDefaultOptions directive can alter the default behavior of such listings, but implying that a certain option (or options) is always present. For example, to force all directory listings to always display ".dotfiles", one might: LsDefaultOptions "–a"

See also

Examples

MasqueradeAddress

Name

MasqueradeAddress –– Configure the server address presented to clients

Synopsis

MasqueradeAddress [`MasqueradeAddress ip-address|dns-hostname`]

`Default`
: none

`Context`
: server config, <VirtualHost>

`Module`
: mod_core

`Compatibility`
: 1.2.2 and later

Description

MasqueradeAddress causes the server to display the network information for the specified IP address or DNS hostname to the client, on the assumption that that IP address or DNS host is acting as a NAT gateway or port forwarder for the server.

See also

Examples

```
MasqueradeAddress nat-gw.mydomain.com
```

MaxClients

Name

MaxClients –– Limits the number of users that can connect

Synopsis

MaxClients [`MaxClients number|none [message]`]

`Default`
: MaxClients none

`Context`
: server config, <Anonymous>, <VirtualHost>, <Global>

`Module`
: mod_core

`Compatibility`
: 0.99.0 and later

Description

The MaxClients directive configures the maximum number of authenticated clients which may be logged into a server or anonymous account. Once this limit is reached, additional clients attempting to authenticate will be disconnected. The special value none may be supplied which removes all maximum connection limits from the applicable configuration context. Additionally, an optional message argument may be used which will be displayed to a client attempting to exceed the maximum value; immediately before disconnection. The message argument is parsed for the magic string "%m", which is replaced with the configured maximum value. If message is not supplied, a system−wide default message is used. Example: MaxClients 5 "Sorry, the maximum number of allowed users are already connected (%m)" Results in: 530 Sorry, the maximum number of allowed users are already connected (5)

See also

Examples

MaxClientsPerHost

Name

MaxClientsPerHost –– Limits the connections per client machine

Synopsis

MaxClientsPerHost [`MaxClientsPerHost number|none [message]]`

Default
: MaxClientsPerHost none

Context
: server config, <Anonymous>, <VirtualHost>, <Global>

Module
: mod_core

Compatibility
: 1.1.7 and later

Description

The MaxClientsPerHost directive configures the maximum number of clients allowed to connect per host. The optional argument message may be used which will be displayed to a client attempting to exceed the maximum value. If message is not supplied, a default message of "Sorry, the maximum number clients (%m) from your host are already connected." is used.

See also

MaxClients, MaxHostsPerUser

Examples

```
MaxClientsPerHost 1 "Sorry, you may not connect more than one time."
Results in: 530 Sorry, you may not connect more than one time.
```

MaxHostsPerUser

Name

MaxHostsPerUser –– Limit the number of connections per userid

Synopsis

MaxHostsPerUser [`MaxHostsPerUser number|none [message]`]

Default
: MaxHostsPerUser none

Context
: server config, <Anonymous>, <VirtualHost>, <Global>

Module
: mod_core

Compatibility
: 1.2.4 and later

Description

The MaxHostsPerUser directive configures the maximum number of times any given login can connection at any given time. The optional argument message may be used which will be displayed to a client attempting to exceed the maximum value. If message is not supplied, a default message of "Sorry, the maximum number of hosts (%m) for this user already connected."

See also

MaxClients, MaxClientsPerHost

Examples

```
MaxHostsPerUser 1 "Sorry, you may not connect more than one time."
Results in: 530 Sorry, you may not connect more than one time.
```

MaxInstances

Name

MaxInstances –– Sets the maximum number of child processes to be spawned

Synopsis

MaxInstances [`MaxInstances number`]

`Default`
MaxInstances none
`Context`
server config
`Module`
mod_core
`Compatibility`
1.1.6pl1

Description

The MaxInstances directive configures the maximum number of child processes that may be spawned by a parent proftpd process in standalone mode. The directive has no effect when used on a server running in inetd mode. Because each child proftpd process represents a single client connection, this directive also controls the maximum number of simultaneous connections allowed. Additional connections beyond the configured limit are syslog'd and silently disconnected. The MaxInstances directive can be used to prevent undesirable denial−of−service attacks (repeatedly connecting to the ftp port, causing proftpd to fork−bomb). By default, no limit is placed on the number of child processes that may run at one time.

See also

Examples

MaxLoginAttempts

Name

MaxLoginAttempts –– Sets how many password attempts are allowed before disconnection

Synopsis

MaxLoginAttempts [`MaxLoginAttempts number`]

`Default`
MaxLoginAttempts 3

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The MaxLoginAttempts directive configures the maximum number of times a client may attempt to authenticate to the server during a given connection. After the number of attempts exceeds this value, the user is disconnected and an appropriate message is logged via the syslog mechanism.

See also

Examples

MultilineRFC2228

Name

MultilineRFC2228 –– FIXME FIXME

Synopsis

MultilineRFC2228 [`MultilineRFC2228 on|off`]

Default
: MultilineRFC2228 off

Context
: server config

Module
: mod_core

Compatibility
: 1.2.0pre3 and later

Description

By default, proftpd sends multiline responses as per RFC 959, i.e.: 200−First line More lines... 200 Last line RFC 2228 specifies that "6xy" response codes will be sent as follows: 600−First line 600−More lines... 600 Last line Note that 2228 ONLY specifies this for response codes starting with '6'. Enabling this directive causes ALL responses to be sent in this format, which may be more compatible with certain web browsers and clients. Also note that this is NOT the same as wu−ftpd's multiline responses, which do not comply with any RFC. Using this method of multilines is more likely to be compatible with all clients, although it isn't strictly RFC, and is thus not enabled by default.

See also

Examples

MySQLInfo

Name

MySQLInfo –– Configures the MySQL driver

Synopsis

MySQLInfo [`hostname`] [`sqluser`] [`sqlpass`] [`dbname`]

`Default`
none

`Context`
server config, <Global>, <VirtualHost>

`Module`
mod_sql

`Compatibility`
1.2.0rc2 and later

Description

This directive is deprecated as of 1.2.0. Please use SQLConnectInfo instead.

Configures the MySQL database driver (the database may be remote). A connection isn't made until use of a SQL feature requires it, after which it may be held open for the lifetime of the FTP session depending on the directives in use. Use `"" to specify a null password.

See also

Examples

Order

Name

Order –– Configures the precedence of the Limit directives

Synopsis

Order [`Order allow,deny|deny,allow`]

`Default`
Order allow,deny

`Context`
<Limit>

`Module`
mod_core

`Compatibility`
0.99.0pl6 and later

Description

The Order directive configures the order in which Allow and Deny directives are checked inside of a <Limit> block. Because Allow directives are permissive, and Deny directives restrictive, the order in which they are examined can significantly alter the way security functions. If the default setting of allow,deny is used, "allowed" access permissions are checked first. If an Allow directive explicitly allows access to the <Limit> context, access is granted and any Deny directives are never checked. If Allow did not explicitly permit access, Deny directives are checked. If any Deny directive applies, access is explicitly denied. Otherwise, access is granted. When deny,allow is used, "deny" access restrictions are checked first. If any restriction applies, access is denied immediately. If nothing is denied, Allow permissions are checked. If an Allow explicitly permits access, access to the entire context is permitted; otherwise access is implicitly denied. For clarification, the following illustrates the steps used when checking Allow/Deny access: Order allow,deny Check Allow directives. If one or more apply, exit with result: ALLOW Check Deny directives. If one or more apply, exit with result: DENY Exit with default implicit ALLOW Order deny,allow Check Deny directives. If one or more apply, exit with result: DENY Check Allow directives. If one or more apply, exit with result: ALLOW Exit with default implicit: DENY

See also

Examples

PassivePorts

Name

PassivePorts −− Specify the ftp−data port range to be used

Synopsis

PassivePorts [`PassivePorts min−pasv−port max−pasv−port`]

`Default`
None
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_core
`Compatibility`
1.2.0rc2 and later

Description

PassivePorts restricts the range of ports from which the server will select when sent the PASV command from a client. The server will randomly choose a number from within the specified range until an open port is found. Should no open ports be found within the given range, the server will default to a normal kernel−assigned port, and a message logged.

The port range selected must be in the non−privileged range (eg. greater than or equal to 1024); it is STRONGLY RECOMMENDED that the chosen range be large enough to handle many simultaneous passive connections (for example, 49152−65534, the IANA−registered ephemeral port range).

See also

Examples

```
# Use the IANA registered ephemeral port range
PassivePorts 49152 65534
```

PathAllowFilter

Name

PathAllowFilter –– FIXME FIXME

Synopsis

PathAllowFilter [`PathAllowFilter regular–expression`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
1.1.7 and later

Description

PathAllowFilter allows the configuration of a regular expression that must be matched for all newly uploaded (stored) files. The regular expression is applied against the entire pathname specified by the client, so care must be taken when creating a proper regex. Paths that fail the regex match result in a "Forbidden filename" error being returned to the client. If the regular–expression argument contains whitespace, it must be enclosed in quotes.

See also

Examples

```
# Only allow a-z 0-9 . - _ in file names,
PathAllowFilter ^[a-z0-9._-]+$

# as above but with upper case characters as well
PathAllowFilter ^[A-Za-z0-9._-]+$
```

PathDenyFilter

Name

PathDenyFilter –– FIXME FIXME

Synopsis

PathDenyFilter [`PathDenyFilter regular-expression`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
1.1.7 and later

Description

Similar to PathAllowFilter, PathDenyFilter specifies a regular expression which must not match any uploaded pathnames. If the regex does match, a "Forbidden filename" error is returned to the client. This can be especially useful for forbidding .ftpaccess or .htaccess files.

See also

Examples

```
# We don't want .ftpaccess or .htaccess files to be uploaded
PathDenyFilter "(\\.ftpaccess)|(\\.htaccess)$"
```

PersistentPasswd

Name

PersistentPasswd –– FIXME FIXME

Synopsis

PersistentPasswd [`PersistentPasswd on|off`]

`Default`
Platform dependent
`Context`
server config
`Module`
mod_unixpw
`Compatibility`
1.1.5 and later

Description

The PersistentPasswd directive controls how proftpd handles authentication, user/group lookups, and user/group to name mapping. If set to On, proftpd will attempt to open the system–wide /etc/passwd, /etc/group (and /etc/shadow, potentially) files itself, holding them open even during a chroot()ed login (note that /etc/shadow is never held open, for security reasons). On some platforms, you must turn this option on, as the libc functions are incapable of accessing these databases from inside of a chroot(). At configure–time, the configuration script will attempt to detect whether or not you need this support, and make it the default. However, such "guessing" may fail, and you will have to manually enable or disable the feature. If you cannot see user or group names when performing a directory listing inside an anonymous chrooted login, this indicates you must enable the directive. Use of the AuthUserFile or AuthGroupFile directives will force partial support for persistent user or group database files; regardless of PersistentPasswd's setting.

Note: NIS or NIS+ users will most likely want to disable this feature, regardless of proftpd's detected configuration defaults. Failure to disable this will make your NIS/NIS+ maps not work! On certain systems, you may also need to compile ProFTPD with the ––enable–autoshadow option in order to authenticate both users from NIS maps and local users.

See also

Examples

PidFile

Name

PidFile –– FIXME FIXME

Synopsis

PidFile [`PidFile filename`]

`Default`
none
`Context`
server config, <Global>
`Module`
mod_core
`Compatibility`
1.2.0rc2 and later

Description

The PidFile directive sets the file to which the server records the process id of the daemon. The filename should be relative to the system root, ie /var/run/proftpd/pidfile. The PidFile is only used in standalone mode. It is often useful to be able to send the server a signal, so that it closes and then reopens its ErrorLog and TransferLog, and re−reads its configuration files. This is done by sending a SIGHUP (kill −1) signal to the process id of the master daemon listed in the PidFile.

See also

Examples

Port

Name

Port –– FIXME FIXME

Synopsis

Port [`Port port-number`]

`Default`
: Port 21

`Context`
: server config, <VirtualHost>

`Module`
: mod_core

`Compatibility`
: 0.99.0 and later

Description

The Port directive configures the TCP port which proftpd will listen on while running in standalone mode. It has no effect when used upon a server running in inetd mode (see ServerType). The directive can be used in conjunction with <VirtualHost> in order to run a virtual server on the same IP address as the master server, but listening on a different port.

For any server, either <VirtualHost> or server config, setting Port 0 effectively turns off that server.

See also

Examples

PostgresInfo

Name

PostgresInfo –– Postgres backend configuration (Deprecated)

Synopsis

PostgresInfo [`hostname`] [[`sqluser`] [`sqlpass`]] [`dbname`]

`Default`
none
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0rc2 and later

Description

This directive is deprecated, please use SQLConnectInfo instead.

Configures the Posgresql database driver (the database may be remote). A connection isn't made until use of a SQL feature requires it, after which it may be held open for the lifetime of the FTP session depending on the directives in use.

See also

Examples

PostgresInfo myserver.example.com proftpd wibble ftpusers

PostgresPort

Name

PostgresPort –– Sets the port postgres is listening on

Synopsis

PostgresPort [`portnumber`]

`Default`
5432
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0rc2 and later

Description

This directive is deprecated, please use SQLConnectInfo instead

Specifies which TCP/IP port to use for connecting. Default is 5432, or UNIX socket for localhost.

See also

Examples

PostgresPort 3306

QuotaBlockName

Name

QuotaBlockName –– FIXME FIXME

Synopsis

QuotaBlockName [`QuotaBlockName name`]

`Default`
byte

`Context`
server, <VirtualHost>, <Anonymous>

`Module`
mod_quota

`Compatibility`
at least 1.2.0 and later

Description

The QuotaBlockName directive is used in conjunction with the QuotaBlockSize directive to control user output from the module. This directive specifies the name given to the values displayed (ie byte, kilobyte, kb etc etc). Example: QuotaBlockName kb

See also

Examples

QuotaBlockSize

Name

QuotaBlockSize –– FIXME FIXME

Synopsis

QuotaBlockSize [`QuotaBlockSize number of bytes`]

`Default`
None
`Context`
server, <VirtualHost>, <Anonymous>
`Module`
mod_quota
`Compatibility`
at least 1.2.0 and later

Description

The QuotaBlockSize directive is used in conjuntion with the QuotaBlockName directive to control how the user output is handled. QuotaBlockSize specifies the factor by which the values in the user reports are divided before display. Example: QuotaBlockSize 1024

See also

Examples

QuotaCalc

Name

QuotaCalc –– FIXME FIXME

Synopsis

QuotaCalc [`QuotaCalc foo1 foo2 foo3`]

`Default`
None
`Context`
server, <VirtualHost>, <Anonymous>
`Module`
mod_quota
`Compatibility`
at least 1.2.0 and later

Description

The QuotaCalc directive controls whether calculation is done on the fly. If the directive is set to 'on' and either there is no .quota file or the quota would go negative then calculation is done on the fly rather than at the end of the session.

See also

Examples

QuotaExempt

Name

QuotaExempt –– FIXME FIXME

Synopsis

QuotaExempt [`QuotaExempt uid, uid, uid`]

`Default`
None
`Context`
server, <VirtualHost>, <Anonymous>
`Module`
mod_quota
`Compatibility`
at least 1.2.0 and later

Description

The QuotaExempt directive lists the UIDs which are not subject to quota controls, using UIDs rather than symbolic user names speeds up the loading and resolution process. Example: QuotaExempt 3000,3401,500

See also

Examples

Quotas

Name

Quotas –– FIXME FIXME

Synopsis

Quotas [`Quotas on|off`]

`Default`
none

`Context`
server, <VirtualHost>, <Anonymous>

`Module`
mod_quota

`Compatibility`
at least 1.2.0 and later

Description

The Quotas directive enables or disables Quota support. Example: Quotas on

See also

Examples

QuotaType

Name

QuotaType –– FIXME FIXME

Synopsis

QuotaType [`QuotaType soft|hard`]

Default
: soft

Context
: server, <VirtualHost>, <Anonymous>

Module
: mod_quota

Compatibility
: at least 1.2.0 and later

Description

The QuotaType directive defines what happens to files which break the quota limits as they are uploaded. Setting the type to hard ensures that the file which violates the quota is deleted. uploaded.

See also

Examples

RateReadBPS

Name

RateReadBPS –– FIXME FIXME

Synopsis

RateReadBPS [`RateReadBPS byte_per_sec-number`]

`Default`
0
`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>
`Module`
mod_xfer
`Compatibility`
1.2.0 and later

Description

RateReadBPS sets the allowed byte per second download bandwidth in the given config context. Zero means no bandwidth limit. (See RateReadFreeBytes about limiting bandwidth only after some amount of downloaded bytes.) The usual place for this directive is in <VirtualHost> or <Directory> sections.

See also

Examples

RateReadFreeBytes

Name

RateReadFreeBytes –– FIXME FIXME

Synopsis

RateReadFreeBytes [`RateReadFreeBytes number of bytes`]

`Default`
0
`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>
`Module`
mod_xfer
`Compatibility`
1.2.0 and later

Description

RateReadFreeBytes is the amount of bytes to be transferred without any bandwidth limits, so with that option you can give full bandwidth for small files while limiting big ones. (See RateReadHardBPS on further info about what happens after the free amount was transferred.)

See also

Examples

RateReadHardBPS

Name

RateReadHardBPS –– FIXME FIXME

Synopsis

RateReadHardBPS [`RateReadHardBPS on/off`]

`Default`
off
`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>
`Module`
mod_xfer
`Compatibility`
1.2.0 and later

Description

RateReadHardBPS forces the bandwidth to the given RateReadBPS value after the RateReadFreeBytes amount of file was transfered. This means that if the user have huge bandwidth and downloaded the "free" amount fast, HardBPS will stop the transfer until the average goes down to the given limit. If the amount of FreeBytes is high and the ReadBPS is low then the user may wait for extended periods of time until the transfer continues. :–)

See also

Examples

RateWriteBPS

Name

RateWriteBPS –– FIXME FIXME

Synopsis

RateWriteBPS [`RateWriteBPS byte_per_sec-number`]

`Default`
0
`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>
`Module`
mod_xfer
`Compatibility`
1.2.0 and later

Description

RateWriteBPS sets the allowed byte per second upload bandwidth in the given config context. Zero means no bandwidth limit. (See RateWriteFreeBytes about limiting bandwidth only after some amount of uploaded bytes.) The usual place for this directive is in <VirtualHost> or <Directory> sections.

See also

Examples

RateWriteFreeBytes

Name

RateWriteFreeBytes –– FIXME FIXME

Synopsis

RateWriteFreeBytes [`RateWriteFreeBytes number of bytes`]

`Default`
0
`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>
`Module`
mod_xfer
`Compatibility`
1.2.0 and later

Description

RateWriteFreeBytes is the amount of bytes to be transferred without any bandwidth limits, so with that option you can give full bandwidth for small files while limiting big ones. (See RateWriteHardBPS on further info about what happens after the free amount was transferred.)

See also

Examples

RateWriteHardBPS

Name

RateWriteHardBPS –– FIXME FIXME

Synopsis

RateWriteHardBPS [`RateWriteHardBPS on/off`]

`Default`
off
`Context`
server config, <VirtualHost>, <Anonymous>, <Directory>, <Global>
`Module`
mod_xfer
`Compatibility`
1.2.0 and later

Description

RateWriteHardBPS forces the bandwidth to the given RateWriteBPS value after the RateWriteFreeBytes amount of file was transfered. This means that if the user have huge bandwidth and uploaded the "free" amount fast, HardBPS will stop the transfer until the average goes down to the given limit. If the amount of FreeBytes is high and the WriteBPS is low then the user may wait for extended periods of time until the transfer continues. :–) RateWriteHardBPS RatioFile (mod_ratio) Incomplete Ratios (mod_ratio) Incomplete RatioTempFile (mod_ratio) Incomplete

See also

Examples

RatioFile

Name

RatioFile –– Ratio directive

Synopsis

RatioFile [`RatioFile foo1 foo2 foo3`]

`Default`
: None known

`Context`
: <Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
: mod_ratio

`Compatibility`
: at least 1.2.0 and later

Description

The RatioFile directive .... Example: RatioFile

See also

Examples

Ratios

Name

Ratios –– FIXME FIXME

Synopsis

Ratios [`Ratios foo1 foo2 foo3`]

`Default`
: None known

`Context`
: <Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
: mod_ratio

`Compatibility`
: at least 1.2.0 and later

Description

The Ratios directive .... Example: Ratios

See also

Examples

RatioTempFile

Name

RatioTempFile –– Ratio directive

Synopsis

RatioTempFile [`RatioTempFile foo1 foo2 foo3`]

`Default`
None known

`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
mod_ratio

`Compatibility`
at least 1.2.0 and later

Description

The RatioTempFile directive .... Example: RatioTempFile

See also

Examples

RequireValidShell

Name

RequireValidShell –– Allow connections based on /etc/shells

Synopsis

RequireValidShell [`RequireValidShell on|off`]

`Default`
RequireValidShell on
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The RequireValidShell directive configures the server, virtual host or anonymous login to allow or deny logins which do not have a shell binary listed in /etc/shells. By default, proftpd disallows logins if the user's default shell is not listed in /etc/shells. If /etc/shells cannot be found, all default shells are assumed to be valid.

See also

Examples

RLimitCPU

Name

RLimitCPU –– Configure the maximum CPU time in seconds used by a process

Synopsis

RLimitCPU [`RLimitCPU soft-limit|"max" [hard-limit|"max"]`]

`Default`
System defaults

`Context`
server config

`Module`
mod_core

`Compatibility`
1.2.1rc1 and later

Description

RLimitCPU takes 1 or 2 parameters. The first parameter sets the soft resource limit for all proftpd processes. The optional second parameter sets the maximum resource limit. Either parameter can be a number, or max to indicate to the server that the limit should be set to the maximum allowed by the operating system configuration.

CPU resource limits are expressed in seconds per process.

See also

RLimitMemory, RLimitOpenFiles

Examples

RLimitMemory

Name

RLimitMemory –– Configure the maximum memory in bytes used by a process

Synopsis

RLimitMemory [`RLimitMemory soft-limit[units]`|`"max"`
`[hard-limit[units]`|`"max"]]`

`Default`
None
`Context`
server config
`Module`
mod_core
`Compatibility`
1.2.1rc1 and later

Description

RLimitMemory takes 1 or 2 parameters. The first parameter sets the soft resource limit for all proftpd processes. The optional second parameter sets the maximum resource limit. Either parameter can be a number, or max to indicate to the server that the limit should be set to the maximum allowed by the operating system configuration.

Memory resource limits are expressed in bytes per process. An optional case–insensitive units specifier may follow the number of bytes given: G (Gigabytes), M (Megabytes), K (Kilobytes), or B (bytes). If the units specifier is used, the given number of bytes is multiplied by the appropriate factor.

See also

RLimitCPU, RLimitMaxProcesses, RLimitOpenFiles

RLimitOpenFiles

Name

RLimitOpenFiles –– Configure the maximum number of open files used by a process

Synopsis

RLimitOpenFiles [`RLimitOpenFiles soft-limit|"max" [hard-limit|"max"]]`

`Default`
None
`Context`
server config
`Module`
mod_core
`Compatibility`
1.2.1rc1 and later

Description

RLimitOpenFiles takes 1 or 2 parameters. The first parameter sets the soft resource limit for all proftpd processes. The optional second parameter sets the maximum resource limit. Either parameter can be a number, or max to indicate to the server that the limit should be set to the maximum allowed by the operating system configuration.

File resource limits are expressed in number of files per process.

See also

RLimitCPU, RLimitMaxProcesses, RLimitMemory

RootLogin

Name

RootLogin –– Permit root user logins

Synopsis

RootLogin [`RootLogin on|off`]

`Default`
RootLogin off

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_auth

`Compatibility`
1.1.5 and later

Description

Normally, proftpd disallows root logins under any circumstance. If a client attempts to login as root, using the correct password, a special security message is sent to syslog. When the RootLogin directive is turned On, the root user may authenticate just as any other user could (assuming no other access control measures deny access); however the root login security message is still sysloged. Obviously, extreme care should be taken when using this directive.

The use of RootLogin in the Anonymous context is only valid when the User / Group defined in the Anonymous block is set to 'root'

See also

Examples

SaveRatios

Name

SaveRatios –– FIXME FIXME

Synopsis

SaveRatios [`SaveRatios foo1 foo2 foo3`]

`Default`
None known

`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess

`Module`
mod_ratio

`Compatibility`
at least 1.2.0 and later

Description

The SaveRatios directive .... Example: SaveRatios

See also

Examples

ScoreboardPath

Name

ScoreboardPath –– Sets the path to the scoreboard file

Synopsis

ScoreboardPath [`path`]

`Default`
ScoreboardPath /var/run

`Context`
server config

`Module`
mod_core

`Compatibility`
1.1.6 and later

Description

The ScoreboardPath directive sets the directory where proftpd run–time scoreboard files (proftpd–*) are kept. These file(s) are necessary for MaxClients to work properly, as well as other utilities (such as ftpwho and ftpcount).

See also

Examples

ServerAdmin

Name

ServerAdmin –– Set the address for the server admin

Synopsis

ServerAdmin [`ServerAdmin "admin-email-address"`]

`Default`
ServerAdmin root@[ServerName]

`Context`
server config, <VirtualHost>

`Module`
mod_core

`Compatibility`
0.99.0pl10 and later

Description

The ServerAdmin directive sets the email address of the administrator for the server or virtualhost. This address is displayed in magic cookie replacements (see DisplayLogin and DisplayFirstChdir).

See also

Examples

ServerIdent

Name

ServerIdent –– Set the message displayed on connect

Synopsis

ServerIdent [`ServerIdent off|on [identification string]`]

`Default`
: ServerIdent ProFTPD [version] Server (server name) [hostname]

`Context`
: server config, <VirtualHost>, <Global>

`Module`
: mod_core

`Compatibility`
: 1.2.0pre2 and later

Description

The ServerIdent directive sets the default message displayed when a new client connects. Setting this to off displays "[hostname] FTP server ready." If set to on, the directive can take an optional string argument, which will be displayed instead of the default text. Sites desiring to give out minimal information will probably want a setting like ServerIdent on "FTP Server ready.", which won't even reveal the hostname.

See also

Examples

ServerIdent on "Welcome to ftp.linux.co.uk"

ServerName

Name

ServerName –– Configure the name displayed to connecting users

Synopsis

ServerName [`ServerName "name"`]

`Default`
ServerName "ProFTPD Server [version]"

`Context`
server config, <VirtualHost>

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The ServerName directive configures the string that will be displayed to a user connecting to the server (or virtual server if the directive is located in a <VirtualHost> block). See Also: <VirtualHost>

See also

Examples

ServerType

Name

ServerType –– Set the mode proftpd runs in

Synopsis

ServerType [`ServerType type−identifier`]

`Default`
ServerType standalone

`Context`
server config

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The ServerType directive configures the server daemon's operating mode. The type−identifier can be one of two values: inetd The daemon will expect to be run from the inetd "super server." New connections are passed from inetd to proftpd and serviced immediately. standalone The daemon starts and begins listening to the configured port for incoming connections. New connections result in spawned child processes dedicated to servicing all requests from the newly connected client.

See also

Examples

ShowDotFiles

Name

ShowDotFiles –– Toggle display of 'dotfiles'

Synopsis

ShowDotFiles [`ShowDotFiles on|off`]

`Default`
ShowDotFiles Off

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_ls

`Compatibility`
0.99.0pl6 and later –– Deprecated

Description

If set to on, files starting with a '.', except for the directories '.' and '..', will be displayed in directory listings. This directive has been deprecated in favor of LsDefaultOptions –– e.g., LsDefaultOptions "–A" –– and may be removed in future versions. See Also: LsDefaultOptions

See also

Examples

ShowSymlinks

Name

ShowSymlinks — Toggle the display of symlinks

Synopsis

ShowSymlinks [`ShowSymlinks on|off`]

`Default`

(versions 1.1.5 and beyond) ShowSymlinks On

`Context`

server config, <VirtualHost>, <Anonymous>, <Global>

`Module`

mod_core

`Compatibility`

Description

Compatibility: 0.99.0pl6 and later Symbolic links (if supported on the host OS and filesystem) can be either shown in directory listings (including the target of the link) or can be "hidden" (proftpd dereferences symlinks and reports the target's permissions and ownership). The default behavior is to show all symbolic links when normal users are logged in, and hide them for anonymous sessions. If a symbolic link cannot be dereferenced for any reason (permissions, target does not exist, etc) and ShowSymlinks is off, proftpd displays the link as a directory entry of type 'l' (link) with the ownership and permissions of the actual link. Under ProFTPD versions 1.1.5 and higher, the default behavior in regard to ShowSymlinks has been changed so that symbolic links are always displayed as such (in all cases), unless ShowSymlinks off is explicitly set.

See also

Examples

SocketBindTight

Name

SocketBindTight –– Controls how TCP/IP sockets are created

Synopsis

SocketBindTight [`SocketBindTight on|off`]

`Default`
SocketBindTight off

`Context`
server config

`Module`
mod_core

`Compatibility`
0.99.0pl6 and later

Description

The SocketBindTight directive controls how proftpd creates and binds its initial tcp listen sockets in standalone mode (see ServerType). The directive has no effect upon servers running in inetd mode, because listen sockets are not needed or created. When SocketBindTight is set to off (the default), a single listening socket is created for each port that the server must listen on, regardless of the number of IP addresses being used by <VirtualHost> configurations. This has the benefit of typically requiring a relatively small number of file descriptors for the master daemon process, even if a large number of virtual servers are configured. If SocketBindTight is set to on, a listen socket is created and bound to a specific IP address for the master server and all configured virtual servers. This allows for situations where an administrator may wish to have a particular port be used by both proftpd (on one IP address) and another daemon (on a different IP address). The drawback is that considerably more file descriptors will be required if a large number of virtual servers must be supported. Example: Two servers have been configured (one master and one virtual), with the IP addresses 10.0.0.1 and 10.0.0.2, respectively. The 10.0.0.1 server runs on port 21, while 10.0.0.2 runs on port 2001. SocketBindTight off #default # proftpd creates two sockets, both bound to ALL available addresses. # one socket listens on port 21, the other on 2001. Because each socket is # bound to all available addresses, no other daemon or user process will be # allowed to bind to ports 21 or 2001. ... SocketBindTight on # proftpd creates two sockets again, however one is bound to 10.0.0.1, port 21 # and the other to 10.0.0.2, port 2001. Because these sockets are "tightly" # bound to IP addresses, port 21 can be reused on any address OTHER than # 10.0.0.1, and visa–versa with 10.0.0.2, port 2001. One side–effect of setting SocketBindTight to on is that connections to non–bound addresses will result in a "connection refused" message rather than the typical "500 Sorry, no server available to handle request on xxx.xxx.xxx.xxx.", due to the fact that no listen socket has been bound to the particular address/port pair. This may or may not be aesthetically desirable, depending on your circumstances.

See also

Examples

SQLAuthenticate

Name

SQLAuthenticate — Specify authentication methods and what to authenticate

Synopsis

SQLAuthenticate {on | off}

or

SQLAuthenticate [users [*]] [group [*]] [userset [fast]] [groupset [fast]]

`Default`
: SQLAuthenticate `on`

`Context`
: server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
: mod_sql

`Compatibility`
: 1.2.5rc1 and later

Description

The SQLAuthenticate directive controls the behavior of mod_sql regarding the authentication process. SQLAuthenticate can provide fine grained control over authentication of logins and file access for both users and groups. Using this directive, mod_sql can be configured to be the authoritative authentication mechanism – in that case, mod_sql provides authentication and all other authentication mechanisms will be bypassed.

The syntax for SQLAuthenticate can take one of two possible formats. The simplest syntax is a simple `on | off` format:

on

mod_sql will perform login authentication and will also control file access using both user ID and group ID. This is equivalent to the following alternative syntax:

```
SQLAuthenticate users groups userset groupset
```

off

mod_sql will not perform user or group lookups nor will it control file access or functionality.

A more complex syntax is provided to provide finer control of the behavior of mod_sql. Two features in particular may be controlled via this syntax:

- Authorititative lookups and authentication

-

File access or functionality control based on UID or GID

The following command options are used to control these features. Note that each of these options may be listed in any order.

users[*]

If this option is present, user lookups will take place. Appending an asterisk to `users` will cause mod_sql to become authoritiative for user lookups. All other user authentication methods will be ignored. If this option is not included, mod_sql will not perform any user lookups.

groups[*]

If this option is present, group lookups will take place. Appending an asterisk to `groups` will cause mod_sql to become authoritiative for group lookups. All other authentication methods will be ignored. If this option is not included, mod_sql will not perform any group lookups.

userset[fast]

If this option is present, mod_sql will control file access or functionality by processing the (get|set|end)pwent calls. These calls are used to determine file access rights based on username. This option has no effect if the `user[*]` option is not present.

If mod_sql is used to authenticate a significant number of users, the (set|get|end)pwent calls can become expensive. The number of queries will be n+1, where n is the number of users to be looked up. On a large system, this can significantly slow logins. Using the `usersetfast` option will cause a single query to be performed to lookup all users, speeding up the login process. The drawback to this option is that memory utilization will be increased.

groupset[fast]

If this option is present, mod_sql will control file access or functionality by processing the (get|set|end)grent calls. These calls are used to determine file access rights based on groupname. This option has no effect if the `group[*]` option is not present.

If mod_sql is used to authenticate a significant number of groups, the (set|get|end)grent calls can become expensive. The number of queries will be n+1, where n is the number of groups to be looked up. On a large system, this can significantly slow logins. Using the `groupsetfast` option will cause a single query to be performed to lookup all groups, speeding up the login process. The drawback to this option is that memory utilization will be increased.

Turning off (not including) userset or groupset affects the functionality of mod_sql. Not allowing these lookups may remove the ability to control access or control functionality by group membership, depending on your other auth handlers and the data available to them. At the same time, choosing not to do these lookups may dramatically speed login for many large sites.

The 'fast' suffix is not appropriate for every site. Normally, mod_sql will retrieve a list of users and groups, and get information from the database on a per−user or per−group basis. This is query intensive −− it requires (n+1) queries, where n is the number of users or groups to lookup. By choosing 'fast' lookups, mod_sql will make a single SELECT query to get information from the database.

In exchange for the radical reduction in the number of queries, the single query will increase the memory consumption of the process −− all group or user information will be read at once rather than in discrete chunks.

Note: If the `groupset` option is specified, mod_sql requires that the SQL group table contain only a single record for each group. All members of a group must be specified in the single record. Make sure that the group table is created with a sufficent column size for group members – for example, a MySQL group table should use type `TEXT` for the group members column, providing 65535 characters for listing all of the group members in a comma–separated list.

See also

SQLUserTable , SQLGroupTable , SQLUserInfo , SQLGroupInfo

Examples

If user and group lookups are desired, but other means will be used to perform file access control, and the user/group lookups are not to be authoritatuve, the following directive syntax is appropriate. This is not a particuarly interesting configuration.

```
SQLAuthenticate users groups
```

A more interesting configuration for mod_sql is shown below. In this configuration, mod_sql is authoritative for both users and groups, and also performs access control based on both user name and group membership. Utilizing a configuration such as this removes the need to provide a shell account for users on the server, while still providing "non–anonymous" ftp access with access control. The "fast" option is also used to speed up logins, at the expense of increased memory utilization.

```
SQLAuthenticate users* groups* usersetfast groupsetfast
```

SQLAuthoritative

Name

SQLAuthoritative –– FIXFIXFIX

Synopsis

SQLAuthoritative [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX
`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_sql
`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLAuthTypes

Name

SQLAuthTypes -- FIXME FIXME

Synopsis

SQLAuthTypes [[OpenSSL]] [[Crypt]] [[Backend]] [[Plaintext]] [[Empty]]

Default
: none

Context
: server config, <Global>, <VirtualHost>

Module
: mod_sql

Compatibility
: 1.2.0 and later

Description

This directive deprecates 'SQLEmptyPasswords', 'SQLScrambledPasswords', 'SQLSSLHashedPasswords', 'SQLPlaintextPasswords', and 'SQLEncryptedPasswords'. Specifies the allowed authentication types and their check order. YOU MUST SPECIFY AT LEAST ONE AUTHENTICATION METHOD. For example: SQLAuthTypes Crypt Empty means check whether the password in the database matches in UNIX crypt() format; if that fails, check to see if the password in the database is empty (matching ANY given password); if that fails, mod_sql refuses to authenticate the user. Current Types Plaintext: allows passwords in the database to be in plaintext OpenSSL: allows passwords in the database to be of the form '{digestname}hashedvalue'. This check is only available if you define 'HAVE_OPENSSL' when you compile proftd and you link with the OpenSSL 'crypto' library. Crypt: allows passwords in the database to be in UNIX crypt() form Backend: a database-specific backend check function. Not all backends support this. Specifically, the MySQL backend uses this type to authenticate MySQL 'PASSWORD()' encrypted passwords. The Postgres backend does nothing. Empty: allows empty passwords in the database, which match against ANYTHING the user types in. The database field must be a truly empty string -- that is, NULL values are never accepted. BE VERY CAREFUL WITH THIS AUTHTYPE.

SQLConnectInfo

Name

SQLConnectInfo –– FIXME FIXME

Synopsis

SQLConnectInfo [`connection-info`] [`[username]`] [`[password]`]

`Default`
: none

`Context`
: server config, <Global>, <VirtualHost>

`Module`
: mod_sql

`Compatibility`
: 1.2.0 and later

Description

This directive deprecates 'MySQLInfo', 'PostgresInfo', and 'PostgresPort'. Specifies connection information. Connection–info specifies the database, host, port, and other backend–specific information. username and password specify the username and password to connect as, respectively. Both default to NULL, which the backend will treat in some backend–specific manner. If you specify a password, you MUST specify a username. Any given backend has the opportunity (but not the responsibility) to check for syntax errors in the connection–info field at proftpd startup, but you shouldn't expect semantic errors (i.e., can't connect to the database) to be caught until mod_sql attempts to connect for a given host. The MySQL and Postgres backends connection–info is expected to be of the form: database[@hostname][:port] hostname will default to a backend–specific hostname (which happens to be 'localhost' for both the MySQL and Postgres backends), and port will default to a backend–specific default port (3306 for the MySQL backend, 5432 for the Postgres backend). Examples: SQLConnectInfo ftpusers@foo.com means "Try connecting to the database 'ftpuser' via the default port at 'foo.com'. Use a NULL username and a NULL password." SQLConnectInfo ftpusers:3000 admin means "Try connecting to the database 'ftpuser' via port 3000 at 'localhost'. Use the username 'admin' and a NULL password." SQLConnectInfo ftpusers@foo.com:3000 admin mypassword means "Try connecting to the database 'ftpuser' via port 3000 at 'foo.com'. Use the username 'admin' and the password 'mypassword'" Backends may require different information in the connection–info field; check your backend module for specifics.

SQLDefaultGID

Name

SQLDefaultGID — FIXME FIXME

Synopsis

SQLDefaultGID [`defaultgid`]

`Default`
65533
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0 and later

Description

Sets the default GID for users. Must be greater than SQLMinID.

SQLDefaultHomedir

Name

SQLDefaultHomedir –– FIXFIXFIX

Synopsis

SQLDefaultHomedir [`"name" limit|regex|ip value`]

Default
: FIXFIXFIX

Context
: server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

Module
: mod_sql

Compatibility
: 1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLDefaultUID

Name

SQLDefaultUID –– FIXME FIXME

Synopsis

SQLDefaultUID [`defaultuid`]

`Default`
65533
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0 and later

Description

Sets the default UID for users. Must be greater than SQLMinID.

SQLDoAuth

Name

SQLDoAuth –– FIXME FIXME

Synopsis

SQLDoAuth [`on` | `off`]

    `Default`
    on
    `Context`
    server config, <Global>, <VirtualHost>
    `Module`
    mod_sql
    `Compatibility`
    1.2.0 and later

Description

Activates SQL authentication. This overrides all other directives –– SQLDoGroupAuth and SQLAuthoritative are ineffectual if SQLDoAuth is off.

SQLDoGroupAuth

Name

SQLDoGroupAuth –– FIXME FIXME

Synopsis

SQLDoGroupAuth [`on` | `off`]

`Default`
on

`Context`
server config, <Global>, <VirtualHost>

`Module`
mod_sql

`Compatibility`
1.2.0 and later

Description

This directive causes mod_sql to pretend it has no group information. It necessarily breaks ALL CONFIG FILES up to 1.2.0rc2, since mod_sql now assumes that group information is available UNLESS this directive is set to OFF. This DOESN'T override SQLAuthoritative –– if SQLAuthoritative is set to 'On' but SQLDoGroupAuth is set to 'Off', all group–related queries will fail without giving other modules the opportunity to handle them. Prior to 1.2.0, there was no way to provide group information from the database. This caused a few bugs, and reduced the functionality of this module.

SQLEmptyPasswords

Name

SQLEmptyPasswords −− Allow zero length passwords (DEPRECATED)

Synopsis

SQLEmptyPasswords [`on` | `off`]

`Default`
off

`Context`
server config, <Global>, <VirtualHost>

`Module`
mod_sql

`Compatibility`
1.2.0rc2 and later

Description

This directive is deprecated, please use SQLAuthTypes instead

Specifies whether an empty (non−NULL but zero−length) password is acceped from the database. Default is no, and truly NULL passwords are never accepted. If the retrieved password is empty then whatever password the user typed is accepted as valid, but the module logs a warning at debug level 4.

See also

Examples

SQLEmptyPasswords on

SQLEncryptedPasswords

Name

SQLEncryptedPasswords −− Assume SQL passwords are encrypted (DEPRECATED)

Synopsis

SQLEncryptedPasswords [`on`|`off`]

`Default`
on
`Context`
server config
`Module`
mod_sql
`Compatibility`
1.2.0rc2 and later

Description

This directive is deprecated, please SQLAuthTypes instead

Specifies whether the password in the database may be in UNIX crypt() format. Default is true, with this being the only check done. A tool for generating crypted password text may be found at ftp://ftp.linpeople.org/pub/People/lilo/source/makepasswd−1.07.tar.gz

See also

Examples

SQLEncryptedPasswords on

SQLGidField

Name

SQLGidField –– FIXFIXFIX

Synopsis

SQLGidField [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX
`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_sql
`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLGroupGIDField

Name

SQLGroupGIDField –– FIXFIXFIX

Synopsis

SQLGroupGIDField [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLGroupInfo

Name

SQLGroupInfo –– FIXFIXFIX

Synopsis

SQLGroupInfo [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLGroupMembersField

Name

SQLGroupMembersField –– FIXME FIXME

Synopsis

SQLGroupMembersField [`fieldname`]

`Default`
members
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0 and later

Description

Specifies the field in the group table that holds the group's member list.

SQLGroupnameField

Name

SQLGroupnameField –– FIXME FIXME

Synopsis

SQLGroupnameField [`Syntax: fieldname`]

`Default`
: groupname

`Context`
: server config, <Global>, <VirtualHost>

`Module`
: mod_sql

`Compatibility`
: 1.2.0 and later

Description

Specifies the field in the group table that holds the group name.

SQLGroupTable

Name

SQLGroupTable -- FIXME FIXME

Synopsis

SQLGroupTable [`tablename`]

`Default`
groups
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0 and later

Description

Specifies the name of the table that holds group information.

SQLGroupWhereClause

Name

SQLGroupWhereClause -- FIXFIXFIX

Synopsis

SQLGroupWhereClause [`"name" limit|regex|ip value`]

Default
: FIXFIXFIX

Context
: server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

Module
: mod_sql

Compatibility
: 1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLHomedir

Name

SQLHomedir –– FIXFIXFIX

Synopsis

SQLHomedir [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLHomedirField

Name

SQLHomedirField –– FIXFIXFIX

Synopsis

SQLHomedirField [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX
`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_sql
`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLHomedirOnDemand

Name

SQLHomedirOnDemand –– FIXME FIXME

Synopsis

SQLHomedirOnDemand [`on`|`off`]

`Default`
off

`Context`
server config, <Global>, <VirtualHost>

`Module`
mod_sql

`Compatibility`
1.2.0 and later

Description

Specifies whether to automatically create a user's home directory if it doesn't exist at login.

SQLLog

Name

SQLLog –– FIXFIXFIX

Synopsis

SQLLog [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX
`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_sql
`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLLogDirs

Name

SQLLogDirs –– FIXFIXFIX

Synopsis

SQLLogDirs [`"name" limit|regex|ip value`]

Default
: FIXFIXFIX

Context
: server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

Module
: mod_sql

Compatibility
: 1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLLogHits

Name

SQLLogHits –– FIXFIXFIX

Synopsis

SQLLogHits [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX
`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_sql
`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLLogHosts

Name

SQLLogHosts –– FIXFIXFIX

Synopsis

SQLLogHosts [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLLoginCountField

Name

SQLLoginCountField –– FIXFIXFIX

Synopsis

SQLLoginCountField [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLLogStats

Name

SQLLogStats –– FIXFIXFIX

Synopsis

SQLLogStats [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLMinID

Name

SQLMinID –– FIXME FIXME

Synopsis

SQLMinID [`minimumid`]

Default
: 999

Context
: server config, <Global>, <VirtualHost>

Module
: mod_sql

Compatibility
: 1.2.0 and later

Description

SQLMinID is checked whenever retrieving a user's GID or UID. If the retrieved values for GID or UID are less than the value of SQLMinID, they are reported as the values of, respectively, 'SQLDefaultGID' and 'SQLDefaultUID'.

SQLMinUserGID

Name

SQLMinUserGID –– FIXFIXFIX

Synopsis

SQLMinUserGID [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX
`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_sql
`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLMinUserUID

Name

SQLMinUserUID –– FIXFIXFIX

Synopsis

SQLMinUserUID [`"name" limit|regex|ip value`]

`Default`
: FIXFIXFIX

`Context`
: server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
: mod_sql

`Compatibility`
: 1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLNamedQuery

Name

SQLNamedQuery –– FIXFIXFIX

Synopsis

SQLNamedQuery [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLPasswordField

Name

SQLPasswordField –– FIXFIXFIX

Synopsis

SQLPasswordField [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLProcessGrEnt

Name

SQLProcessGrEnt –– FIXFIXFIX

Synopsis

SQLProcessGrEnt [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX
`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_sql
`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLProcessPwEnt

Name

SQLProcessPwEnt –– FIXFIXFIX

Synopsis

SQLProcessPwEnt [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX
`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess
`Module`
mod_sql
`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLRatios

Name

SQLRatios -- FIXFIXFIX

Synopsis

SQLRatios [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLRatioStats

Name

SQLRatioStats –– FIXFIXFIX

Synopsis

SQLRatioStats [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLScrambledPasswords

Name

SQLScrambledPasswords –– FIXME FIXME

Synopsis

SQLScrambledPasswords [`on` | `off`]

`Default`
off
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0 and later

Description

This directive is DEPRECATED. Please use SQLAuthTypes instead. Specifies whether to accept passwords in a backend specific format. For the MySQL backend, this means 'PASSWORD()' scrambled passwords. For the Postgres backend, this check does nothing.

SQLShellField

Name

SQLShellField –– FIXME FIXME

Synopsis

SQLShellField [`fieldname`]

`Default`
shell
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0 and later

Description

Specifies the field in the user table that holds the user's shell. If this field doesn't exist or the result of the query is NULL, the shell is reported as "".

SQLShowInfo

Name

SQLShowInfo –– FIXFIXFIX

Synopsis

SQLShowInfo [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLSSLHashedPasswords

Name

SQLSSLHashedPasswords –– FIXME FIXME

Synopsis

SQLSSLHashedPasswords [`on` | `off`]

`Default`
off
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0 and later

Description

This directive is DEPRECATED. Please use SQLAuthTypes instead. Specifies whether to accept passwords of the form {digestname}hashedpassword from the database. This directive is only available if you define 'HAVE_OPENSSL' when you compile proftd and you link with the OpenSSL 'crypto' library. As an example, any of the following password entries in the database would match if the user typed the password 'testpassword': {SHA}IoFZRnP0iujh/70lps6DjKPgwkk= {SHA1}i7YRj4/Wk1rQh2o740pxfTJwj/0= {MD2}nS6iguewvAdrCnOMyQjB1w== {MD4}5wsGtJCkyXBzDJoVsQKjSg== {MD5}4WsquNEjFL9O+9YgOQbqbA==

SQLUidField

Name

SQLUidField –– FIXFIXFIX

Synopsis

SQLUidField [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLUserInfo

Name

SQLUserInfo –– FIXFIXFIX

Synopsis

SQLUserInfo [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLUsernameField

Name

SQLUsernameField –– FIXFIXFIX

Synopsis

SQLUsernameField [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLUserTable

Name

SQLUserTable –– FIXFIXFIX

Synopsis

SQLUserTable [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLUserWhereClause

Name

SQLUserWhereClause –– FIXFIXFIX

Synopsis

SQLUserWhereClause [`"name" limit|regex|ip value`]

`Default`
FIXFIXFIX

`Context`
server config, <Global>, <VirtualHost>, <Anonymous>, <Limit>, .ftpaccess

`Module`
mod_sql

`Compatibility`
1.2.5rc1 and later

Description

FIX FIX FIX

See also

Examples

FIXFIXFIX

FIXFIX

SQLWhereClause

Name

SQLWhereClause –– FIXME FIXME

Synopsis

SQLWhereClause [`whereclause`]

`Default`
none
`Context`
server config, <Global>, <VirtualHost>
`Module`
mod_sql
`Compatibility`
1.2.0 and later

Description

This directive deprecates 'SQLKey' and 'SQLKeyField'. Specifies a where clause that is added to every user query (this has no effect on group queries). The where clause *must* contain all relevant punctuation, and *must not* contain a leading 'and'. As an example of switching from the old–style 'SQLKey' and 'SQLKeyField' directives, if you had: SQLKey true SQLKeyfield LoginAllowed You would now use: SQLWhereClause "LoginAllowed = 'true'" This would be appended to every user–related query as the string " and (LoginAllowed = 'true')"

SyslogFacility

Name

SyslogFacility –– Set the facility level used for logging

Synopsis

SyslogFacility [`SyslogFacility facility-level`]

`Default`
None
`Context`
server config
`Module`
mod_core
`Compatibility`
1.1.6 and later

Description

Proftpd logs its activity via the Unix syslog mechanism, which allows for several different general classifications of logging messages, known as "facilities." Normally, all authentication related messages are logged with the AUTHPRIV (or AUTH) facility [intended to be secure, and never seen by unwanted eyes], while normal operational messages are logged with the DAEMON facility. The SyslogFacility directive allows ALL logging messages to be directed to a different facility than the default. When this directive is used, ALL logging is done with the specified facility, both authentication (secure) and otherwise. The facility–level argument must be one of the following: AUTH (or AUTHPRIV), CRON, DAEMON, KERN, LPR, MAIL, NEWS, USER, UUCP, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6 or LOCAL7. See Also: SystemLog

See also

Examples

SyslogLevel

Name

SyslogLevel –– Set the verbosity level of system logging

Synopsis

SyslogLevel [`SyslogLevel emerg|alert|crit|error|warn|notice|info|debug`]

`Default`
None

`Context`
server config, <VirtualHost>, <Anonymous>, <Global>

`Module`
mod_core

`Compatibility`
1.2.0rc2+cvs and later

Description

SyslogLevel adjusts the verbosity of the messages recorded in the error logs. The following levels are available, in order of decreasing significance: Level Description emerg Emergencies – system is unusable. alert Action must be taken immediately. crit Critical Conditions. error Error conditions. warn Warning conditions. notice Normal but significant condition. info Informational. debug Debug–level messages When a particular level is specified, messages from all other levels of higher significance will be reported as well. E.g., when SyslogLevel info is specified, then messages with log levels of notice and warn will also be posted. Using a level of at least crit is recommended.

See also

Examples

SystemLog

Name

SystemLog –– Redirect syslogging to a file

Synopsis

SystemLog [`SystemLog filename`|`NONE`]

`Default`
None
`Context`
server config
`Module`
mod_log
`Compatibility`
1.1.6pl1 and later

Description

The SystemLog directive disables proftpd's use of the syslog mechanism and instead redirects all logging output to the specified filename. The filename argument should contain an absolute path, and should not be to a file in a nonexistent directory, in a world–writeable directory, or be a symbolic link (unless AllowLogSymlinks is set to on). Use of this directive overrides any facility set by the SyslogFacility directive. Additionally, the special keyword NONE can be used which disables all syslog style logging for the entire configuration.

See also

AllowLogSymlinks

Examples

TCPAccessFiles

Name

TCPAccessFiles –– Sets the access files to use

Synopsis

TCPAccessFiles [`allow-filename deny-filename`]

`Default`
: none

`Context`
: server config, <VirtualHost>, <Global>, <Anonymous>

`Module`
: mod_wrap

`Compatibility`
: 1.2.1 and later

Description

TCPAccessFiles specifies two files, an allow and a deny file, each of which contain the IP addresses, networks or name–based masks to be allowed or denied connections to the server. The files have the same format as the standard tcpwrappers hosts.allow/deny files.

Both file names are required. Also, the paths to both files must be the full path, with two exceptions: if the path starts with ~/, the check of that path will be delayed until a user requests a connection, at which time the path will be resolved to that user's home directory; or if the path starts with ~user/, where user is some system user. In this latter case, mod_wrap will attempt to resolve and verify the given user's home directory on start–up.

The service name for which mod_wrap will look in the indicated access files is proftpd by default; this can be configured via the TCPServiceName directive. There is a built–in precedence to the TCPAccessFiles, TCPGroupAccessFiles, and TCPUserAccessFiles directives, if all are used. mod_wrap will look for applicable TCPUserAccessFiles for the connecting user first. If no applicable TCPUserAccessFiles is found, mod_wrap will search for TCPGroupAccessFiles which pertain to the connecting user. If not found, mod_wrap will then look for the server–wide TCPAccessFiles directive. This allows for access control to be set on a per–server basis, and allow for per–user or per–group access control to be handled without interfering with the server access rules.

See also

TCPGroupAccessFiles, TCPServiceName, TCPUserAccessFiles

Examples

server−wide access files TCPAccessFiles /etc/ftpd.allow /etc/ftpd.deny # per−user access files, which are to be found in the user's home directory TCPAccessFiles ~/my.allow ~/my.deny

TCPAccessSyslogLevels

Name

TCPAccessSyslogLevels –– Sets the logging levels for mod_wrap

Synopsis

TCPAccessSyslogLevels [`allow-level deny-level`]

`Default`
TCPAccessSyslogLevels info warn

`Context`
server config, <VirtualHost>, <Global>, <Anonymous>

`Module`
mod_wrap

`Compatibility`
1.2.1 and later

Description

ProFTPD can log when a connection is allowed, or denied, as the result of rules in the files specified in TCPAccessFiles, to the Unix syslog mechanism. A discussion on the syslog levels which can be used is given in the SyslogLevel directive.

The allow−level parameter sets the syslog level at which allowed connections are logged; the deny−level parameter sets the syslog level for denied connections.

See also

SyslogLevel

Examples

TCPAccessSyslogLevels debug warn

tcpBackLog

Name

tcpBackLog –– Control the tcp backlog in standalone mode

Synopsis

tcpBackLog [`tcpBackLog backlog−size`]

Default
: tcpBackLog 5

Context
: server config

Module
: mod_core

Compatibility
: 0.99.0 and later

Description

The tcpBackLog directive controls the tcp "backlog queue" when listening for connections in standalone mode (see ServerType). It has no affect upon servers in inetd mode. When a tcp connection is established by the tcp/ip stack inside the kernel, there is a short period of time between the actual establishment of the connection and the acceptance of the connection by a user−space program. The duration of this latency period is widely variable, and can depend upon several factors (hardware, system load, etc). During this period tcp connections cannot be accepted, as the port that was previously "listening" has become filled with the new connection. Under heavy connection load this can result in occasional (or even frequent!) "connection refused" messages returned to the incoming client, even when there is a service available to handle requests. To eliminate this problem, most modern tcp/ip stacks implement a "backlog queue" which is simply a pre−allocation of resources necessary to handle backlog−size connections during the latency period. The larger the backlog queue, the more connections can be established in a very short time period. The trade−off, of course, is kernel memory and/or other kernel resources. Generally it is not necessary to use a tcpBackLog directive, unless you intend to service a large number of virtual hosts (see <VirtualHost>), or have a consistently heavy system load. If you begin to notice or hear of "connection refused" messages from remote clients, try setting a slightly higher value to this directive.

See also

Examples

TCPGroupAccessFiles

Name

TCPGroupAccessFiles –– Sets the access files to use

Synopsis

TCPGroupAccessFiles [`group-expression allow-filename deny-filename`]

`Default`
none
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_wrap
`Compatibility`
1.2.1 and later

Description

TCPGroupAccessFiles allows for access control files, the same types of files required by TCPAccessFiles, to be applied to select groups. The given group–expression is a logical AND expression, which means that the connecting user must be a member of all the groups listed for this directive to apply. Group names may be negated with a ! prefix.

The rules for the filename paths are the same as for TCPAccessFiles settings.

See also

TCPAccessFiles, TCPUserAccessFiles

Examples

every member of group wheel must connect from restricted locations TCPGroupAccessFiles wheel /etc/ftpd–strict.allow /etc/ftpd–strict.deny # everyone else gets the standard access rules TCPGroupAccessFiles !wheel /etc/hosts.allow /etc/hosts.deny

tcpNoDelay

Name

tcpNoDelay –– Control the use of TCP_NODELAY

Synopsis

tcpNoDelay [`tcpNoDelay on|off`]

`Default`
tcpNoDelay on

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_core

`Compatibility`
1.2.0pre3a and later

Description

The tcpNoDelay directive controls the use of the TCP_NODELAY socket option (which disables the Nagle algorithm). ProFTPd uses TCP_NODELAY by default, which usually is a benefit but this can occasionally lead to problems with some clients, so tcpNoDelay is provided as a way to disable this option. You will not normally need to use this directive but if you have clients reporting unusually slow connections, try setting this to off.

See also

Examples

tcpReceiveWindow

Name

tcpReceiveWindow –– Set the size of the tcp receive window

Synopsis

tcpReceiveWindow [`tcpReceiveWindow window-size`]

`Default`
tcpReceiveWindow 8192

`Context`
server config, <VirtualHost>

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The tcpReceiveWindow directive configures the size (in octets) of all data connections' tcp receive windows. It is only used when receiving a file from a client over the data connection. Typically, a given tcp/ip implementation will use a relatively small receive window size (the number of octets that can be received at the tcp layer before a "turnaround" acknowledgement is required). When transferring a large amount of data over fast digital transmission lines which have a relatively high latency, a small receive window can dramatically affect perceived throughput because of the necessity to completely stop the transfer occasionally in order to wait for the remote endpoint to receive the acknowledgement and continue transmission. For example, on a T1 line (assuming full 1.544Mbps endpoint−to−endpoint throughput) with 100 ms latency, a 4k receive buffer will very dramatically reduce the perceived throughput. The default value of 8192 octets (8k) should be reasonable in common network configurations. Additionally, proftpd allocates its internal buffers to match the receive/send window sizes; in order to maximize the reception/transmission performance (reducing the number of times data must be transfered from proftpd to the kernel tcp/ip stack). The tradeoff, of course, is memory; both kernel− and user−space. If running proftpd on a memory tight host (and on a low−bandwidth connection), it might be advisable to decrease both the tcpReceiveWindow and tcpSendWindow sizes.

See also

Examples

tcpSendWindow

Name

tcpSendWindow –– Set the size of the tcp send window

Synopsis

tcpSendWindow [`tcpSendWindow window-size`]

`Default`
tcpSendWindow 8192
`Context`
server config, <VirtualHost>
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The tcpSendWindow directive configures the size (in octets) of all data connections' tcp send windows. It is only used when sending a file from the server to a client on the data connection. For a detailed description of receive/send window sizes see tcpReceiveWindow.

See also

Examples

TCPServiceName

Name

TCPServiceName –– Configures the name proftpd will use with mod_wrap

Synopsis

TCPServiceName [`name`]

`Default`
TCPServiceName proftpd

`Context`
server config, <VirtualHost>, <Global>

`Module`
mod_wrap

`Compatibility`
1.2.1 and later

Description

TCPServiceName is used to configure the name of the service under which mod_wrap will check the allow/deny files. By default, this is the name of the program started, i.e. "proftpd". However, some administrators may want to use a different, more generic service name, such as "ftpd"; use this directive for such needs.

See also

TCPUserAccessFiles

Name

TCPUserAccessFiles –– Sets the access files to use

Synopsis

TCPUserAccessFiles [`user−expression allow−filename deny−filename`]

`Default`
none
`Context`
server config, <VirtualHost>, <Global>
`Module`
mod_wrap
`Compatibility`
1.2.1 and later

Description

TCPUserAccessFiles allows for access control files, the same types of files required by TCPAccessFiles, to be applied to select users. The given user−expression is a logical AND expression. Listing multiple users in a user−expression does not make much sense; however, this type of AND evaluation allows for expressions such as "everyone except this user" with the use of the ! negation prefix.

The rules for the filename paths are the same as for TCPAccessFiles settings.

See also

TCPAccessFiles, TCPGroupAccessFiles

Examples

user admin might be allowed to connect from anywhere TCPUserAccessFiles admin /etc/ftpd−anywhere.allow /etc/ftpd−anywhere.deny # while every other user has to connect from LAN addresses TCPUserAccessFiles !admin /etc/ftpd−lan.allow /etc/ftpd−lan.deny

TimeoutIdle

Name

TimeoutIdle –– Sets the idle connection timeout

Synopsis

TimeoutIdle [`TimeoutIdle seconds`]

`Default`
TimeoutIdle 600
`Context`
server config
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The TimeoutIdle directive configures the maximum number of seconds that proftpd will allow clients to stay connected without receiving any data on either the control or data connection. If data is received on either connection, the idle timer is reset. Setting TimeoutIdle to 0 disables the idle timer completely (clients can stay connected for ever, without sending data). This is generally a bad idea as a "hung" tcp connection which is never properly disconnected (the remote network may have become disconnected from the Internet, etc) will cause a child server to never exit (at least not for a considerable period of time) until manually killed See Also: TimeoutLogin, TimeoutNoTransfer

See also

Examples

TimeoutLogin

Name

TimeoutLogin –– Sets the login timeout

Synopsis

TimeoutLogin [`TimeoutLogin seconds`]

`Default`
TimeoutLogin 300

`Context`
server config

`Module`
mod_core

`Compatibility`
0.99.0 and later

Description

The TimeoutLogin directive configures the maximum number of seconds a client is allowed to spend authenticating. The login timer is not reset when a client transmits data, and is only removed once a client has transmitted an acceptable USER/PASS command combination. See Also: TimeoutIdle, TimeoutNoTransfer

See also

Examples

TimeoutNoTransfer

Name

TimeoutNoTransfer –– Sets the connection without transfer timeout

Synopsis

TimeoutNoTransfer [`TimeoutNoTransfer seconds`]

`Default`
TimeoutNoTransfer 300
`Context`
server config
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The TimeoutNoTransfer directive configures the maximum number of seconds a client is allowed to spend connected, after authentication, without issuing a command which results in creating an active or passive data connection (i.e. sending/receiving a file, or receiving a directory listing). See Also: TimeoutIdle, TimeoutLogin

See also

Examples

TimeoutStalled

Name

TimeoutStalled –– Sets the timeout on stalled downloads

Synopsis

TimeoutStalled [`TimeoutStalled seconds`]

`Default`
TimeoutStalled 3600
`Context`
server config
`Module`
mod_core
`Compatibility`
1.1.6 and later

Description

The TimeoutStalled directive sets the maximum number of seconds a data connection between the proftpd server and an FTP client can exist but have no actual data transferred (i.e. "stalled"). If the seconds argument is set to 0, data transfers are allowed to stall indefinitely.

See also

Examples

TimesGMT

Name

TimesGMT –– Toggle time display between GMT and local

Synopsis

TimesGMT [`TimesGMT on|off`]

`Default`
(versions 1.2.0pre9 and beyond) on
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`

Description

Compatibility: 1.2.0pre9 and later The TimesGMT option causes the server to report all ls and MDTM times in GMT and not local time.

See also

Examples

TransferLog

Name

TransferLog –– Specify the path to the transfer log

Synopsis

TransferLog [`TransferLog filename` | `NONE`]

`Default`
TransferLog /var/log/xferlog

`Context`
server config, <Anonymous>, <VirtualHost>, <Global>

`Module`
mod_core

`Compatibility`
1.1.4 and later

Description

The TransferLog directive configures the full path to the "wu–ftpd style" file transfer log. Separate log files can be created for each Anonymous and/or VirtualHost. Additionally, the special keyword NONE can be used, which disables wu–ftpd style transfer logging for the context in which the directive is used (only applicable to version 1.1.7 and later). See Also: ExtendedLog, LogFormat

See also

Examples

Umask

Name

Umask –– Set the default Umask

Synopsis

Umask [`Umask file octal-mask [directory octal-mask]]`

`Default`
None
`Context`
server config, <Anonymous>, <VirtualHost>, <Directory>, <Global>, .ftpaccess
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

Umask sets the mask applied to newly created file and directory permissions within a given context. By default, the Umask in the server configuration, <VirtualHost> or <Anonymous> block is used, unless overridden by a "per–directory" Umask setting. Any arguments supplied must be an octal number, in the format 0xxx. An optional second argument can specify a Umask to be used when creating directories. If a second argument isn't specified, directories are created using the default Umask in the first argument. For more information on umasks, consult your operating system documentation/man pages.

Proftpd will not create files that have the execution bit turned on, this is a security driven design decision. The permissions of the uploaded file can be changed by issuing a SITE CHMOD command can be used to change the mode of the uploaded file. Syntax of the command is: SITE CHMOD <mode> <file>.

See also

Examples

UseFtpUsers

Name

UseFtpUsers –– Block based on /etc/ftpusers

Synopsis

UseFtpUsers [`UseFtpUsers on|off`]

Default
: UseFtpUsers on

Context
: server config, <Anonymous>, <VirtualHost>, <Global>

Module
: mod_core

Compatibility
: 0.99.0 and later

Description

Legacy FTP servers generally check a special authorization file (typically /etc/ftpusers) when a client attempts to authenticate. If the user's name is found in this file, FTP access is denied. For compatibility sake, proftpd defaults to checking this file during authentication. This behavior can be suppressed using the UseFtpUsers configuration directive.

See also

Examples

UseGlobbing

Name

UseGlobbing –– Toggles use of glob() functionality

Synopsis

UseGlobbing [`on` | `off`]

`Default`
UseGlobbing on

`Context`
server config, <VirtualHost>, <Global>, <Anonymous>

`Module`
mod_ls

`Compatibility`
1.2.5rc1 and later

Description

The UseGlobbing directive controls use of glob() functionality, which is needed for supporting wildcard characters such as *.

See also

User

Name

User –– Set the user the daemon will run as

Synopsis

User [`User userid`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
0.99.0 and later

Description

The User directive configures which user the proftpd daemon will normally run as. By default, proftpd runs as root which is considered undesirable in all but the most trustful network configurations. The User directive used in conjunction with the Group directive instructs the daemon to switch to the specified user and group as quickly as possible after startup. On some unix variants, the daemon will occasionally switch back to root in order to accomplish a task which requires super–user access. Once the task is completed, root privileges are relinquished and the server continues to run as the specified user and group. When applied to a <VirtualServer> block, proftpd will run as the specified user/group on connections destined for the virtual server's address or port. If either User or Group is applied to an <Anonymous> block, proftpd will establish an anonymous login when a user attempts to login with the specified userid, as well as permanently switching to the corresponding uid/gid (matching the User/Group parameters found in the anonymous block) after login. Note: When an authorized unix user is authenticated and logs in, all former privileges are released, the daemon switches permanently to the logged in user's uid/gid, and is never again capable of switching back to root or any other user/group.

See also

Examples

UserAlias

Name

UserAlias –– Alias a username to a system user

Synopsis

UserAlias [`UserAlias login-user userid`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_auth
`Compatibility`
0.99.0 and later

Description

ProFTPD requires a real username/uid when authenticating users as provided by PAM, AuthUserFile or another authentication mechanism. There are however times when additional aliases are required but it is undesirable to provide additional login accounts.

UserAlias provides a mechanism to do this, a typical and common example is within Anonymous configuration blocks. It is normal for the server to use 'ftp' as the primary authentication user, however it is common practice for users to login using "anonymous". This is achieved by adding the following to the config file.

See also

Examples

UserAlias anonymous ftp

UserDirRoot

Name

UserDirRoot –– Set the chroot directory to a subdirectory of the anonymous server

Synopsis

UserDirRoot [`UserDirRoot on|off`]

`Default`
off
`Context`
<Anonymous>
`Module`
mod_auth
`Compatibility`
1.2.0pre2 and later

Description

When set to true, the chroot base directory becomes a subdirectory of the anonymous ftp directory, based on the username of the current user. For example, assuming user "foo" is aliased to "ftp", logging in as "foo" causes proftpd to run as real user ftp, but to chroot into ~ftp/foo instead of just ~ftp.

See also

Examples

UseReverseDNS

Name

UseReverseDNS –– Toggle rDNS lookups

Synopsis

UseReverseDNS [`UseReverseDNS on|off`]

`Default`
: UseReverseDNS on

`Context`
: server config

`Module`
: mod_core

`Compatibility`
: 1.1.7 and later

Description

Normally, incoming active mode data connections and outgoing passive mode data connections have a reverse DNS lookup performed on the remote host's IP address. In a chroot environment (such as <Anonymous> or DefaultRoot), the /etc/hosts file cannot be checked and the only possible resolution is via DNS. If for some reason, DNS is not available or improperly configured this can result in proftpd blocking ("stalling") until the libc resolver code times out. Disabling this directive prevents proftpd from attempting to reverse–lookup data connection IP addresses.

See also

Examples

UserOwner

Name

UserOwner —– Set the user ownership of new files / directories

Synopsis

UserOwner [`UserOwner username`]

`Default`
None

`Context`
<Anonymous>, <Directory>

`Module`
mod_core

`Compatibility`
1.2pre11 and later

Description

The UserOwner directive configures which user all newly created directories and files will be owned by, within the context that UserOwner is applied to. The user ID of username cannot be 0 (root). Where it is used, the GroupOwner directive is not restricted to groups that the current user is a member of.

See also

Examples

UserPassword

Name

UserPassword –– Creates a hardcoded username/password pair

Synopsis

UserPassword [`UserPassword userid hashed–password`]

`Default`
None
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
0.99.0pl5 and later

Description

The UserPassword directive creates a password for a particular user which overrides the user's normal password in /etc/passwd (or /etc/shadow). The override is only effective inside the context to which UserPassword is applied. The hashed–password argument is a cleartext string which has been passed through the standard unix crypt() function. Do NOT use a cleartext password. This can be useful when combined with UserAlias to provide multiple logins to an Anonymous FTP site. See Also: GroupPassword

See also

Examples

UserRatio

Name

UserRatio –– Ratio directive

Synopsis

UserRatio [`UserRatio foo1 foo2 foo3`]

`Default`
None known
`Context`
<Directory>, <Anonymous>, <Limit>,.ftpaccess
`Module`
mod_ratio
`Compatibility`
at least 1.2.0 and later

Description

The UserRatio directive .... Example: UserRatio

See also

Examples

VirtualHost

Name

VirtualHost –– Define a virtual ftp server

Synopsis

VirtualHost [`<VirtualHost address>`]

Default
: None

Context
: server config

Module
: mod_core

Compatibility
: 0.99.0 and later

Description

The VirtualHost configuration block is used to create an independent set of configuration directives that apply to a particular hostname or IP address. It is often used in conjunction with system level IP aliasing or dummy network interfaces in order to establish one or more "virtual" servers which all run on the same physical machine. The block is terminated with a </VirtualHost> directive. By utilizing the Port directive inside a VirtualHost block, it is possible to create a virtual server which uses the same address as the master server, but listens on a separate tcp port (incompatible with ServerType inetd). When proftpd starts, virtual server connections are handled in one of two ways, depending on the ServerType setting: inetd The daemon examines the destination address and port of the incoming connection handed off from inetd. If the connection matches one of the configured virtual hosts, the connection is serviced based on the appropriate configuration. If no virtual host matches, and the main server does not match, the client is informed that no server is available to service their requests and disconnected. standalone After parsing the configuration file, the daemon begins listening for connections on all configured ports, spawning child processes as necessary to handle connections for either the main server or any virtual servers. Because of the method that the daemon uses to listen for connections when in standalone mode, it is possible to support an exceedingly large number of virtual servers, potentially exceeding the number of per–process file descriptors. This is due to the fact that a single file descriptor is used to listen to each configured port, regardless of the number of addresses being monitored. Note that it may be necessary to increase the tcpBackLog value on heavily loaded servers in order to avoid kernel rejected client connections ("Connection refused").

See also

Examples

WtmpLog

Name

WtmpLog –– Toggle logging to wtmp

Synopsis

WtmpLog [`WtmpLog on|off|NONE`]

`Default`
WtmpLog on
`Context`
server config, <VirtualHost>, <Anonymous>, <Global>
`Module`
mod_core
`Compatibility`
1.1.7 and later

Description

The WtmpLog directive controls proftpd's logging of ftp connections to the host system's wtmp file (used by such commands as `last'). By default, all connections are logged via wtmp. Please report any corrections or additions via http://bugs.proftpd.net/

See also

Examples

II. Configuration by Module

This is a list of all the configuration directives organised by the module in which they are defined with details of each module, it's purpose and the development team behind it.

Table of Contents

mod_auth

Name

mod_auth –– Authentication module

Synopsis

mod_auth

Description

FIXME FIXME FIXME

See also

DefaultChdir DefaultRoot LoginPasswordPrompt RootLogin UserAlias UserDirRoot

mod_code

Name

mod_code –– FIX ME FIX ME

Synopsis

mod_code

Description

FIXME FIXME FIXME

See also

DisplayConnect DisplayFirstChdir

mod_core

Name

mod_core –– Core module

Synopsis

mod_core

Description

This module provides all the core functionality ProFTPD needs to function, this module must be compiled in.

See also

AccessDenyMsg AccessGrantMsg Allow AllowAll AllowFilter AllowForeignAddress AllowGroup AllowOverwrite AllowRetrieveRestart AllowStoreRestart AllowUser AnonRequirePassword Anonymous AnonymousGroup AuthAliasOnly AuthUsingAlias Bind CDPath Class Classes CommandBufferSize DefaultServer DefaultTransferMode DeferWelcome DeleteAbortedStores Deny DenyAll DenyFilter DenyGroup DenyUser Directory DisplayGoAway DisplayLogin DisplayQuit Global Group GroupOwner GroupPassword HiddenStor HideGroup HideNoAccess HideUser IdentLookups IgnoreHidden Include Limit MasqueradeAddress MaxClients MaxClientsPerHost MaxHostsPerUser MaxInstances MaxLoginAttempts MultilineRFC2228 Order PassivePorts PathAllowFilter PathDenyFilter PidFile Port RequireValidShell RLimitCPU RLimitMemory RLimitOpenFiles ScoreboardPath ServerAdmin ServerIdent ServerName ServerType ShowSymlinks SocketBindTight SyslogFacility SyslogLevel tcpBackLog tcpNoDelay tcpReceiveWindow tcpSendWindow TimeoutIdle TimeoutLogin TimeoutNoTransfer TimeoutStalled TimesGMT TransferLog Umask UseFtpUsers User UseReverseDNS UserOwner UserPassword VirtualHost WtmpLog

mod_ldap

Name

mod_ldap –– LDAP authentication support

Synopsis

mod_ldap

Description

mod_ldap provides LDAP authentication support for ProFTPD. It supports many features useful in "toaster" environments such as default UID/GID and autocreation/autogeneration of home directories.

See also

LDAPAuthBinds LDAPDefaultAuthScheme LDAPDefaultGID LDAPDefaultUID LDAPDNInfo LDAPDoAuth LDAPDoGIDLookups LDAPDoUIDLookups LDAPForceDefaultGID LDAPForceDefaultUID LDAPHomedirOnDemand LDAPHomedirOnDemandPrefix LDAPHomedirOnDemandPrefixNoUsername LDAPHomedirOnDemandSuffix LDAPNegativeCache LDAPQueryTimeout LDAPSearchScope LDAPServer LDAPUseTLS

mod_log

Name

mod_log –– Logging support

Synopsis

mod_log

Description

Logging support, including enhanced formatting options.

See also

AllowLogSymlinks ExtendedLog LogFormat SystemLog

mod_ls

Name

mod_ls –– file listing functionality

Synopsis

mod_ls

Description

FIXME FIXME FIXME

See also

DirFakeGroup DirFakeMode DirFakeUser LsDefaultOptions ShowDotFiles UseGlobbing

mod_pam

Name

mod_pam –– Pluggable authentication modules support

Synopsis

mod_pam

Description

FIXME FIXME FIXME

See also

AuthPAM AuthPAMConfig

mod_quota

Name

mod_quota –– Module to implement per–user quotas

Synopsis

mod_quota

Description

FIXME FIXME FIXME

Notes

mod_quota forces a quota recalculation when the .quota file is missing. It is therefore possible to tie normal shell logins into deleting the file to force recalcualtions.

See also

DefaultQuota QuotaBlockName QuotaBlockSize QuotaCalc QuotaExempt Quotas QuotaType

mod_ratio

Name

mod_ratio –– FIX ME FIX ME

Synopsis

mod_ratio

Description

FIXME FIXME FIXME

See also

AnonRatio ByteRatioErrMsg CwdRatioMsg FileRatioErrMsg GroupRatio HostRatio LeechRatioMsg RatioFile Ratios RatioTempFile SaveRatios UserRatio

mod_readme

Name

mod_readme –– "README" file support

Synopsis

mod_readme

Description

FIXME FIXME FIXME

See also

DisplayReadme

mod_sample

Name

mod_sample –– Example module

Synopsis

mod_sample

Description

This module only provides an example set of code as a template for a budding module programmer.

See also

FooBarDirective

mod_site

Name

mod_site –– FIX ME FIX ME

Synopsis

mod_site

Description

FIXME FIXME FIXME

See also

AllowChmod

mod_sql

Name

mod_sql –– SQL support module

Synopsis

mod_sql

Description

This module provides the necessary support for SQL based authentication, logging and other features as required. It replaces the SQL modules which were shipped with 1.2.0rc2 and earlier.

See also

MySQLInfo PostgresInfo PostgresPort SQLAuthenticate SQLAuthoritative SQLAuthTypes SQLConnectInfo SQLDefaultGID SQLDefaultHomedir SQLDefaultUID SQLDoAuth SQLDoGroupAuth SQLEmptyPasswords SQLEncryptedPasswords SQLGidField SQLGroupGIDField SQLGroupInfo SQLGroupMembersField SQLGroupnameField SQLGroupTable SQLGroupWhereClause SQLHomedir SQLHomedirField SQLHomedirOnDemand SQLLog SQLLogDirs SQLLogHits SQLLogHosts SQLLoginCountField SQLLogStats SQLMinID SQLMinUserGID SQLMinUserUID SQLNamedQuery SQLPasswordField SQLProcessGrEnt SQLProcessPwEnt SQLRatios SQLRatioStats SQLScrambledPasswords SQLShellField SQLShowInfo SQLSSLHashedPasswords SQLUidField SQLUserInfo SQLUsernameField SQLUserTable SQLUserWhereClause SQLWhereClause

mod_unixpw

Name

mod_unixpw –– UNIX style authentication methods

Synopsis

mod_unixpw

Description

This module supports the password file (/etc/passwd) style of authentication methods.

See also

AuthGroupFile AuthPAMAuthoritative AuthUserFile PersistentPasswd

mod_wrap

Name

mod_wrap –– Interface to libwrap

Synopsis

mod_wrap

Description

It enables the daemon to use the common tcpwrappers access control library while in standalone mode, and in a very configurable manner. It is not compiled by default.

If not installed on your system, the TCP wrappers library, required by this module, can be found here, on Wietse Venema's site. Once installed, it highly recommended that the hosts_access(3) and hosts_access(5) man pages be read and understood.

Many programs will automatically add entries in the common allow/deny files, and use of this module will allow a ProFTPD daemon running in standalone mode to adapt as these entries are added. The portsentry program does this, for example: when illegal access is attempted, it will add hosts to the /etc/hosts.deny file.

See also

TCPAccessFiles TCPAccessSyslogLevels TCPGroupAccessFiles TCPServiceName TCPUserAccessFiles

mod_xfer

Name

mod_xfer –– FIX ME FIX ME

Synopsis

mod_xfer

Description

FIXME FIXME FIXME

See also

RateReadBPS RateReadFreeBytes RateReadHardBPS RateWriteBPS RateWriteFreeBytes
RateWriteHardBPS

III. Configuration by Context

This is a list of all the configuration directives organised by the module in which they are defined with details of each module, it's purpose and the development team behind it.

Table of Contents

server config

Name

server config –– server config

Synopsis

server config

Description

FIXME FIXME FIXME

See also

Global

Name

Global –– Global

Synopsis

Global

Description

FIXME FIXME FIXME

See also

VirtualHost

Name

VirtualHost –– VirtualHost

Synopsis

VirtualHost

Description

FIXME FIXME FIXME

See also

Anonymous

Name

Anonymous –– Anonymous

Synopsis

Anonymous

Description

FIXME FIXME FIXME

See also

AccessDenyMsg AccessGrantMsg AllowAll AllowChmod AllowFilter AllowForeignAddress AllowOverwrite AllowRetrieveRestart AllowStoreRestart AnonRatio AnonRequirePassword AuthAliasOnly AuthUsingAlias ByteRatioErrMsg CDPath CwdRatioMsg DefaultChdir DefaultQuota DeleteAbortedStores DenyAll DenyFilter Directory DirFakeGroup DirFakeMode DirFakeUser DisplayFirstChdir DisplayGoAway DisplayLogin DisplayQuit DisplayReadme ExtendedLog FileRatioErrMsg FooBarDirective Group GroupOwner GroupPassword GroupRatio HiddenStor HideGroup HideNoAccess HideUser HostRatio Include LDAPHomedirOnDemandPrefixNoUsername LeechRatioMsg Limit LoginPasswordPrompt LsDefaultOptions MaxClients MaxClientsPerHost MaxHostsPerUser PathAllowFilter PathDenyFilter QuotaBlockName QuotaBlockSize QuotaCalc QuotaExempt Quotas QuotaType RateReadBPS RateReadFreeBytes RateReadHardBPS RateWriteBPS RateWriteFreeBytes RateWriteHardBPS RatioFile Ratios RatioTempFile RequireValidShell RootLogin SaveRatios ShowDotFiles ShowSymlinks SQLAuthenticate SQLAuthoritative SQLDefaultHomedir SQLGidField SQLGroupGIDField SQLGroupInfo SQLGroupWhereClause SQLHomedir SQLHomedirField SQLLog SQLLogDirs SQLLogHits SQLLogHosts SQLLoginCountField SQLLogStats SQLMinUserGID SQLMinUserUID SQLNamedQuery SQLPasswordField SQLProcessGrEnt SQLProcessPwEnt SQLShowInfo SQLUidField SQLUserInfo SQLUsernameField SQLUserTable SQLUserWhereClause SyslogLevel TCPAccessFiles TCPAccessSyslogLevels TimesGMT TransferLog Umask UseFtpUsers UseGlobbing User UserAlias UserDirRoot UserOwner UserPassword UserRatio WtmpLog

Limit

Name

Limit –– Limit

Synopsis

Limit

Description

FIXME FIXME FIXME

See also

Allow AllowAll AllowGroup AllowUser AnonRatio ByteRatioErrMsg CwdRatioMsg Deny DenyAll DenyGroup DenyUser FileRatioErrMsg FooBarDirective GroupRatio HostRatio IgnoreHidden LDAPHomedirOnDemandPrefixNoUsername LeechRatioMsg Order RatioFile Ratios RatioTempFile SaveRatios SQLAuthenticate SQLAuthoritative SQLDefaultHomedir SQLGidField SQLGroupGIDField SQLGroupInfo SQLGroupWhereClause SQLHomedir SQLHomedirField SQLLog SQLLogDirs SQLLogHits SQLLogHosts SQLLoginCountField SQLLogStats SQLMinUserGID SQLMinUserUID SQLNamedQuery SQLPasswordField SQLProcessGrEnt SQLProcessPwEnt SQLShowInfo SQLUidField SQLUserInfo SQLUsernameField SQLUserTable SQLUserWhereClause UserRatio

.ftpaccess

Name

.ftpaccess -- .ftpaccess

Synopsis

.ftpaccess

Description

FIXME FIXME FIXME

See also

AllowAll AllowChmod AllowOverwrite AllowRetrieveRestart AllowStoreRestart AnonRatio ByteRatioErrMsg CwdRatioMsg DeleteAbortedStores DenyAll FileRatioErrMsg GroupOwner GroupRatio HostRatio LDAPHomedirOnDemandPrefixNoUsername LeechRatioMsg Limit RatioFile Ratios RatioTempFile SaveRatios SQLAuthenticate SQLAuthoritative SQLDefaultHomedir SQLGidField SQLGroupGIDField SQLGroupInfo SQLGroupWhereClause SQLHomedir SQLHomedirField SQLLog SQLLogDirs SQLLogHits SQLLogHosts SQLLoginCountField SQLLogStats SQLMinUserGID SQLMinUserUID SQLNamedQuery SQLPasswordField SQLProcessGrEnt SQLProcessPwEnt SQLShowInfo SQLUidField SQLUserInfo SQLUsernameField SQLUserTable SQLUserWhereClause Umask UserRatio

VI. Appendices

Table of Contents

Appendix A. Resources

This appendix is under development... I've borrowed the formatting from elsewhere and am busy hacking it around to what i want

placeholer for references and resources... ideas please.. I guess Mysql, postgres, rfc information etc should go here. Scripts for auto generating configs? Links to linux resources?

The quantity of information about SGML and XML is growing on a daily basis. This appendix strives to provide both a complete bibliography of the references mentioned explicitly in this book, and a sampling of resources for additional information about DocBook and about SGML and XML in general. Although not all of these resources are focused specifically on DocBook, they still provide helpful information for DocBook users.

Latest Versions of DocBook

As of July 1998, responsibility for the advancement and maintenance of the DocBook DTD has been transferred from the Davenport Group, which originated it, to the DocBook Technical Committee of OASIS (Organization for the Advancement of Structured Information Standards) at http://www.oasis−open.org/.

The latest releases of DocBook can be obtained from the official DocBook home page at http://www.oasis−open.org/docbook/.

Resources for Resources

Here's where to find pointers to the subjects you want to find.

The Most Recent Version of This Book
: The most recent online version of this book can be found at http://docbook.org/.

The Most Recent Version of Proftpd
: can be found... wibble wobble.

Another mirror
: desc...wibble.

comp.text.sgml and comp.text.xml
: USENET newsgroups devoted to SGML and XML issues.

FAQs
: For pointers to several SGML FAQs, see http://www.oasis−open.org/cover/general.html#faq. The XML FAQ is available at http://www.ucc.ie/xml.

XML.com
: XML.com, run jointly by Songline Studios and Seybold, is a site devoted to making XML accessible.

Introductory Material on the Web

These documents provide a good background for a better understanding of SGML and XML.

A Gentle Introduction to SGML
: A useful and simple document available in its original form at http://www−tei.uic.edu/orgs/tei/sgml/teip3sg/index.html.

A Technical Introduction to XML
: A close look at the ins−and−outs of XML is available at http://nwalsh.com/docs/articles/xml/.

References and Technical Noteson the Web

Entity Management
: OASIS Technical Resolution 9401:1997 (Amendment 2 to TR 9401).

  This document describes OASIS catalog files.

The SGML Declaration
: The SGML Declaration, by Wayne Wholer.

Table Interoperability: Issues for the CALS Table Model
: OASIS Technical Research Paper 9501:1995.

Exchange Table Model Document Type Definition
: OASIS Technical Resolution TR 9503:1995.

CALS Table Model Document Type Definition
: OASIS Technical Memorandum TM 9502:1995

XML Exchange Table Model Document Type Definition
: OASIS Technical Memorandum TM 9901:1999.

Internet RFCs

RFCs ("Request for Comments") are standards documents produced by the Internet Engineering Task Force (IETF).

RFC 959
: File Transfer Protocol (FTP).

RFC 2228
: FTP Security Extensions

Specifications

Here are pointers to the specifications.

The XML Specification
The W3C technical recommendation that defines XML 1.0.

Namespaces in XML
The W3C technical recommendation that defines XML namespaces.

Mathematical Markup Language (MathML) 1.0 Specification
The W3C technical recommendation that defines MathML, an XML representation of mathematical equations.

The Unicode Standard, Version 2.0
The Unicode standard.

Unicode Technical Report #8
Version 2.1 of the Unicode standard.

Books and Printed Resources

There are also a number of books worth checking out:

Bibliography

Developing SGML DTDs: From Text to Model to Markup, Eve Maler and Jeanne El Andaloussi, 0−13−309881−8, Prentice−Hall PTR, Upper Saddle River, 1996.

Practical SGML, Erik van Herwijnen, 2, 0−7923−9434−8, Kluwer Academic Press, 1994.

The SGML Handbook, Charles Goldfarb and Yuri Rubinksy, 0−7923−9434−8, 1991, Oxford University Press.

SGML: an author's guide to the Standard Generalized Markup Language, Martin Bryan, 0−201−17535−5, 1988, Addison−Wesley Publishing Company.

$GML: The Billion Dollar Secret, Chet Ensign, 0−13−226705−5, 1998, Prentice Hall.

Creating Documents with XML, Chris Maden, 1−56592−518−1, 1999, O'Reilly & Associates.

XML: A Primer, Simon St. Laurent, 1−5582−8592−X, 1998, MIS:Press/IDG Books Worldwide.

Understanding SGML and XML Tools, Peter Flynn, 0−7923−8169−6, 1998, Kluwer Academic Publishers.

The LaTeX Web Companion: Integrating TeX, HTML, and XML, Michel Goosens and Sebastian Rahtz, 0−201−43311−7, 1999, Addison−Wesley Publishing Company.

SGML/XML Tools

An attempt to provide a detailed description of all of the SGML/XML tools available is outside the scope of this book.

For a list of recent of SGML tools, check out Robin Cover's SGML/XML page at OASIS: http://www.oasis−open.org/cover.

For a list of XML tools, check out XML.com: http://www.xml.com/.

Appendix B. Cookbook examples

Example B−1. Basic Configuration

```
# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use.  It establishes a single server
# and a single anonymous login.  It assumes that you have a user/group
# "nobody" and "ftp" for normal operation and anon.

ServerName                      "ProFTPD Default Installation"
ServerType                      standalone
DefaultServer                   on

# Port 21 is the standard FTP port.
Port                            21
# Umask 022 is a good standard umask to prevent new dirs and files
# from being group and world writable.
Umask                           022

# To prevent DoS attacks, set the maximum number of child processes
# to 30.  If you need to allow more than 30 concurrent connections
# at once, simply increase this value.  Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd)
MaxInstances                    30

# Set the user and group that the server normally runs at.
User                            nobody
Group                           nogroup

# Normally, we want files to be overwriteable.
<Directory /*>
  AllowOverwrite                on
</Directory>

# A basic anonymous configuration, no upload directories.
<Anonymous ~ftp>
  User                          ftp
  Group                         ftp
  # We want clients to be able to login with "anonymous" as well as "ftp"
  UserAlias                     anonymous ftp

  # Limit the maximum number of anonymous logins
  MaxClients                    10

  # We want 'welcome.msg' displayed at login, and '.message' displayed
  # in each newly chdired directory.
  DisplayLogin                  welcome.msg
  DisplayFirstChdir             .message

  # Limit WRITE everywhere in the anonymous chroot
  <Limit WRITE>
    DenyAll
  </Limit>

</Anonymous>
```

Example B−2. VirtualHost Config

```
# This sample configuration file illustrates creating two
# virtual servers, and associated anonymous logins.

ServerName                      "ProFTPD"
ServerType                      inetd

# Port 21 is the standard FTP port.
Port                            21

# Global creates a "global" configuration that is shared by the
# main server and all virtualhosts.

<Global>
  # Umask 022 is a good standard umask to prevent new dirs and files
  # from being group and world writable.
  Umask                         022
</Global>

# Set the user and group that the server normally runs at.
User                            nobody
Group                           nogroup

# To prevent DoS attacks, set the maximum number of child processes
# to 30.  If you need to allow more than 30 concurrent connections
# at once, simply increase this value.  Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd)
MaxInstances                    30

# Maximum seconds a data connection may "stall"
TimeoutStalled                  300

# First virtual server
<VirtualHost ftp.virtual.com>

  ServerName                    "Virtual.com's FTP Server"

  MaxClients                    10
  MaxLoginAttempts              1

  # DeferWelcome prevents proftpd from displaying the servername
  # until a client has authenticated.
  DeferWelcome                  on

  # Limit normal user logins, because we only want to allow
  # guest logins.
  <Limit LOGIN>
    DenyAll
  </Limit>

  # Next, create a "guest" account (which could be used
  # by a customer to allow private access to their web site, etc)
  <Anonymous ~cust1>
    User                        cust1
    Group                       cust1
    AnonRequirePassword         on

    <Limit LOGIN>
      AllowAll
```

```
    </Limit>

    HideUser                    root
    HideGroup                   root

    # A private directory that we don't want the user getting in to.
    <Directory logs>
      <Limit READ WRITE DIRS>
        DenyAll
      </Limit>
    </Directory>

  </Anonymous>

</VirtualHost>

# Another virtual server, this one running on our primary address,
# but on port 4000.  The only access is to a single anonymous login.
<VirtualHost our.ip.address>

  ServerName                    "Our private FTP server"
  Port                          4000
  Umask                         027

  <Limit LOGIN>
    DenyAll
  </Limit>

  <Anonymous /usr/local/ftp/virtual/a_customer>

    User                        ftp
    Group                       ftp
    UserAlias                   anonymous ftp

    <Limit LOGIN>
      AllowAll
    </Limit>

    <Limit WRITE>
      DenyAll
    </Limit>

    <Directory incoming>
      <Limit WRITE>
        AllowAll
      </Limit>
    </Directory>

  </Anonymous>

</VirtualHost>
```

Example B−3. Complex Configuration

```
#
# Virtual Hosting Server Configuration
# by M.Lowes <markl@ftech.net>
# for Frontier Internet Services Limited
#      (http://www.ftech.net/)
#
```

Proftpd

```
ServerName                      "Master Webserver"
#
# Spawn from inetd?
#
#ServerType         inetd
#
# or maybe a standalone server...
#
ServerType          standalone
#
# don't give the server banner until _after_ authentication
#
DeferWelcome                    off
#
# Some basic defaults
#
Port                  21
Umask                002
TimeoutLogin         120
TimeoutIdle          600
TimeoutNoTransfer    900
TimeoutStalled      3600
#
# No, I don't think we'll run as root!
#
User                            ftp
Group                           ftp
#
# This is a non-customer usable name, (ie they should be connecting via www.{domain})
# not 'hostname'.  Therefore let's dump them in a dummy account and wait for them to
# scream.
#
DefaultRoot                     /web/Legacy/
#
# Performance, let's do DNS resolution when we process the logs...
#
UseReverseDNS        off
#
# Where do we put the pid files?
#
ScoreboardPath                  /var/run/proftpd
#
# Logging options
#
TransferLog                     /var/spool/syslog/proftpd/xferlog.legacy
#
# Some logging formats
#
LogFormat         default "%h %l %u %t \"%r\" %s %b"
LogFormat                       auth    "%v [%P] %h %t \"%r\" %s"
LogFormat                       write   "%h %l %u %t \"%r\" %s %b"
#
# Global settings
#
<Global>
        DisplayLogin                welcome.msg
        DisplayFirstChdir           readme
        #
        # having to delete before uploading is a pain ;)
        #
        AllowOverwrite              yes
        #
```

```
        # Turn off Ident lookups
        #
        IdentLookups         off
        #
        # Logging
        #
        # file/dir access
        #
        ExtendedLog             /var/spool/syslog/proftpd/access.log WRITE,READ write
        #
        #
        # Record all logins
        #
        ExtendedLog             /var/spool/syslog/proftpd/auth.log AUTH auth
        #
        # Paranoia logging level....
        #
   ##ExtendedLog    /var/spool/syslog/proftpd/paranoid.log ALL default
</Global>

#
# Deny writing to the base server...
#
<Limit WRITE>
        DenyAll
</Limit>


# --------------------------------------------
# Virtual Servers start here....
#
# (Note: this is normally auto generated by a
# script written in house).
# --------------------------------------------
#
# www.ftech.net.
# This is the default server
# Gets all the connections for www.{customer.domain},
# & www.ftech.net
#
<VirtualHost www.ftech.net>
        ServerAdmin             webmaster@Ftech.net
        ServerName              "Master Webserver"
        MaxLoginAttempts        2
        RequireValidShell       no
        TransferLog             /var/spool/syslog/proftpd/xferlog.www
        MaxClients              50
        DefaultServer           on
        DefaultRoot             ~ !staff
        AllowOverwrite          yes

        #
        # No quickly do we kick someone out
        #
        TimeoutLogin                    120
        TimeoutIdle                     600
        TimeoutNoTransfer               900

        # --------------------------------------------
        # Got a Frontpage customer who keeps breaking things????
        #  - stick 'em in group fpage
        # --------------------------------------------
```

```
        <Directory ~/public_html>
        #
        # Block them from doing anything other than reading...
        #
                <Limit STOR RNFR DELE>
                        DenyGroup fpage
                </Limit>
        </Directory>
        #
        # ditto for ftp_root if it's there...
        #
        <Directory ~/ftp_root>
                <Limit STOR RNFR DELE>
                        DenyALL
                </Limit>
        </Directory>
        #
        # Limit by IP...
        #
        <Directory /web/zsl>
                <Limit ALL>
                        Order Allow,Deny
                        Allow 195.200.31.220
                        Allow 212.32.17.0/26
                        Deny ALL
                </Limit>
        </Directory>

</VirtualHost>

# --------------------------------------------
#
# Legacy server, left in because some people
# haven't realised it's gone yet.  Shove 'em into
# a dummy $home
#
<VirtualHost web-1.ftech.net>
ServerAdmin             webmaster@Ftech.net
ServerName              "Legacy Web Upload Server"
MaxLoginAttempts        2
RequireValidShell       no
MaxClients              50
DefaultRoot             ~ !staff
MaxClients              2
AllowOverwrite          yes
TransferLog             /var/spool/syslog/proftpd/xferlog.web-1
</VirtualHost>

# --------------------------------------------
#
# ftp.ftech.net
#
<VirtualHost ftp.ftech.net>
ServerAdmin                     ftpmaster@ftech.net
ServerName                      "Frontier Internet Public FTP Server"
TransferLog                     /ftp/xferlog/ftp.ftech.net
MaxLoginAttempts                3
RequireValidShell               no
DefaultRoot                     /ftp/ftp.ftech.net
AllowOverwrite                  yes

#
```

```
# Auth files....
#
AuthUserFile                    /var/conf/ftp/authfiles/passwd.ftp.ftech.net
AuthGroupFile                   /var/conf/ftp/authfiles/group.ftp.ftech.net

# A basic anonymous configuration, no upload directories.
<Anonymous /ftp/ftp.ftech.net>
        User                    ftp
        Group                   ftp
        # We want clients to be able to login with "anonymous" as well as "ftp"
        UserAlias               anonymous ftp
        RequireValidShell               no

        # Limit the maximum number of anonymous logins
        MaxClients              50

        # We want 'welcome.msg' displayed at login, and '.message' displayed
        # in each newly chdired directory.

        <Directory pub/incoming>
                <Limit STOR>
                        AllowAll
                </Limit>
                <Limit WRITE DIRS READ>
                        DenyAll
                </Limit>
                <Limit CWD XCWD CDUP>
                        AllowAll
                </Limit>
        </Directory>

        <Directory home>
                <Limit ALL>
                        DenyAll
                </Limit>
        </Directory>

   #
   # Limit access to the mirrors to LINX
   # only
   #
   <Directory mirrors>
      <Limit RETR>
         Order Allow,Deny
         Allow .uk, .ftech.net
         Allow .vom.tm
         Deny ALL
      </Limit>
   </Directory>

        # Limit WRITE everywhere in the anonymous chroot
        <Limit WRITE>
                DenyAll
        </Limit>


</Anonymous>

</VirtualHost>

# -----------------------------------------------------
# Virtual ftp with anon access, but no incoming
```

```
#
<VirtualHost ftp.foo1.com>
ServerAdmin             ftpmaster@foo1.com
ServerName              "Foo1 FTP Server"
TransferLog             /var/spool/syslog/xfer/ftp.foo1.com
MaxLoginAttempts        3
RequireValidShell       no
DefaultRoot             /ftp/ftp.foo1.com
User                    foo1
Group                   foo1
AllowOverwrite          yes

#
# Auth files....
#
AuthUserFile    /var/conf/ftp//authfiles/passwd.ftp.foo1.com
AuthGroupFile   /var/conf/ftp//authfiles/group.ftp.foo1.com

<Anonymous /ftp/ftp.foo1.com>
        User                    ftp
        Group                   ftp
        UserAlias               anonymous ftp
        RequireValidShell       no
        MaxClients              20
        <Limit WRITE>
                DenyAll
        </Limit>
</Anonymous>
</VirtualHost>


# -----------------------------------------------------
# ftp.foo2.com
# Anon, no incoming, some private access areas
#
<VirtualHost ftp.foo2.com>
ServerAdmin             ftpmaster@mcresearch.co.uk
ServerName              "MC Research FTP Server"
TransferLog             /var/spool/syslog/xfer/ftp.foo2.com
MaxLoginAttempts        3
RequireValidShell       no
DefaultRoot             /ftp/ftp.foo2.com
User                    foo2
Group                   foo2
AllowOverwrite          yes

#
# Auth files....
#
AuthUserFile    /var/conf/ftp//authfiles/passwd.ftp.foo2.com
AuthGroupFile   /var/conf/ftp//authfiles/group.ftp.foo2.com

<Anonymous /ftp/ftp.foo2.com>
        User                    ftp
        Group                   ftp
        UserAlias               anonymous ftp
        RequireValidShell       no
        MaxClients              20

        <Directory download>
                <Limit ALL>
                        DenyAll
```

```
                </Limit>
        </Directory>
        <Limit WRITE>
                DenyAll
        </Limit>
</Anonymous>

        <Directory /ftp/ftp.foo2.com/pub>
                <Limit WRITE>
                        AllowUser mcres
                        DenyAll
                </Limit>
        </Directory>

        <Directory /ftp/ftp.foo2.com/download>
                <Limit ALL>
                        AllowUser mcres
                        AllowUser customer
                        DenyAll
                </Limit>
        </Directory>
</VirtualHost>


# ----------------------------------------------------
# ftp.foo3.com
#
#
<VirtualHost ftp.foo3.com>
ServerAdmin             ftpmaster@farrukh.co.uk
ServerName              "Farrukh FTP Archive"
TransferLog             /var/spool/syslog/xfer/ftp.foo3.com
MaxLoginAttempts        3
RequireValidShell       no
DefaultRoot             /web/farrukh2/ftp_root
User                    farrukh2
Group                   farrukh2
AllowOverwrite          yes

#
# Auth files....
#
AuthUserFile    /var/conf/ftp//authfiles/passwd.ftp.foo3.com
AuthGroupFile   /var/conf/ftp//authfiles/group.ftp.foo3.com

<Anonymous /web/farrukh2/ftp_root>
        User                    ftp
        Group                   ftp
        UserAlias               anonymous ftp
        RequireValidShell       no
        MaxClients              20

        <Directory pub/incoming/*>
                <Limit STOR>
                        AllowAll
                </Limit>
                <Limit WRITE DIRS READ>
                        DenyAll
                </Limit>
                <Limit CWD XCWD CDUP>
                        AllowAll
                </Limit>
```

```
        </Directory>


        <Directory pub/Incoming/*>
                <Limit STOR>
                        AllowAll
                </Limit>
                <Limit WRITE DIRS READ>
                        DenyAll
                </Limit>
                <Limit CWD XCWD CDUP>
                        AllowAll
                </Limit>
        </Directory>
        #
        # block access to the secure areas by anon...
        #
        <Directory fpub>
                <Limit ALL>
                        DenyAll
                </Limit>
        </Directory>

        <Directory fgroup>
                <Limit ALL>
                        DenyAll
                </Limit>
        </Directory>
        <Limit WRITE>
                DenyAll
        </Limit>
</Anonymous>

        #
        # define user based access
        #
        <Directory /web/farrukh2/ftp_root/fpub>
                <Limit ALL>
                        AllowUser farrukh
                        AllowUser fguest
                        DenyAll
                </Limit>
        </Directory>

        <Directory /web/farrukh2/ftp_root/fgroup>
                <Limit ALL>
                        AllowUser farrukh
                        AllowUser fgroup
                        DenyAll
                </Limit>
        </Directory>
</VirtualHost>


# ------------------------------------------------------
# ftp.foo4.com
# anon, with incoming upload
#
<VirtualHost ftp.foo4.com>
ServerAdmin             ftpmaster@teamwork.co.uk
ServerName              "Teamwork FTP Server"
TransferLog             /var/spool/syslog/xfer/ftp.foo4.com
```

```
MaxLoginAttempts        3
RequireValidShell       no
DefaultRoot             /ftp/ftp.foo4.com
User                    foo4
Group                   foo4
AllowOverwrite          yes

#
# Auth files....
#
AuthUserFile    /var/conf/ftp//authfiles/passwd.ftp.foo4.com
AuthGroupFile   /var/conf/ftp//authfiles/group.ftp.foo4.com

<Anonymous /ftp/ftp.foo4.com>
        User                    ftp
        Group                   ftp
        UserAlias               anonymous ftp
        RequireValidShell       no
        MaxClients              20

        <Directory pub/incoming/*>
                <Limit STOR>
                        AllowAll
                </Limit>
                <Limit WRITE DIRS READ>
                        DenyAll
                </Limit>
                <Limit CWD XCWD CDUP>
                        AllowAll
                </Limit>
        </Directory>


        <Directory pub/Incoming/*>
                <Limit STOR>
                        AllowAll
                </Limit>
                <Limit WRITE DIRS READ>
                        DenyAll
                </Limit>
                <Limit CWD XCWD CDUP>
                        AllowAll
                </Limit>
        </Directory>

        <Limit WRITE>
                DenyAll
        </Limit>
</Anonymous>
</VirtualHost>

# -----------------------------------------------------
# The end....
# -----------------------------------------------------
```

Example B-4.

Index

...

Colophon

Initial authoring of this of this book were produced with the DocBook DSSSL Stylesheets. In the best tradition of geek books I've decided to find an animal to shove on this document, given my handle I've picked the closest thing in nature to a flying hamster. The sugar glider.