Global Information Assurance Certification Paper

Securing and Administering Restricted FTP on a Unix Based Server

Richard L. Garfield
June 2003

Summary

The File Transfer Protocol (FTP) is an indispensable network application for exchanging information of all kinds. FTP implementations can be configured to provide access to various populations ranging from an individual user accessing information on his personal computer, to defined groups sharing information, to the Internet population at large. However, FTP is subject to various attacks that threaten the confidentiality and integrity of that information.

This paper will discuss methods for configuring and securing a restricted access FTP server running on an IRIX based SGI host. Standard FTP and Security Extended FTP are discussed as are options for increasing security at a site where a security enhanced FTP package is not being used. The paper concludes with a review of recommended practices for managing the site once it has been configured and placed online.

Overview of FTP

The File Transfer Protocol is an interactive application level protocol that uses TCP to deliver data [1]. The protocol is defined in RFC 959 [2] which specifies a client-server connection scheme and a set of commands for authentication, setting the parameters of the file transfer and executing the transfer. The standard also provides for an optional second client to be the recipient of a file transfer under the direction of the initiating client.

Two connections (socket pairs) are opened during an FTP session. The control connection is established first using the telnet protocol and is used to exchange FTP commands. Based upon the FTP commands issued the data connection is set up and the file transfer takes place.

The command set consists of three groups: access control commands, transfer parameter commands, and ftp service commands. Access control commands are used to authenticate the user, access the desired file system, and logout. Transfer parameter commands are used to change connection characteristics, data types, file structure, and transfer mode. These commands all have default values so they do not always have to be invoked. The service commands define the actual file transfer or a file system function. Some miscellaneous commands,

such as HELP, are also included in this group. These commands are discussed in detail in the RFC and in the UNIX online "man" pages.

An FTP server can be set up in various configurations. Authorized users, that is, those users that can log onto a host with a username and password can access the files they own on that host using FTP. They may move about the file system with the same privileges that they would have in a telnet session. Alternatively, an anonymous FTP service can be provided which requires no authentication but limits the users' access to files and prohibits movement outside a confined area. A third implementation is restricted FTP.

Restricted FTP

Restricted FTP is a special implementation of FTP that is similar to anonymous FTP in that it confines users to a rooted directory and prevents them from moving about the file system outside of that directory as a regular user might. Unlike anonymous FTP, however, restricted FTP requires user authentication.

Restricted FTP is designed to give only authorized users access to information not intended for open dissemination. For example, an organization with geographically dispersed operations may want to make internal documents available to these sites, or researchers at different laboratories may want to share data in a collaborative effort.

A single username and password can be established to be shared among the authorized users or each authorized user can be assigned a user ID and password. The latter is preferred for accountability but can vastly increase the user management task.

Basic Configuration

Proper configuration of a restricted FTP site is of fundamental importance. The basic setup for restricted FTP is described in the "man" pages but some important details are elaborated in [3] and [4]. Four areas need to be considered in configuring the site (the following discussion is based upon the SGI IRIX 6.5.19 implementation of FTP).

Server Environment

The location of a server host in the organization's network topology must be determined first. The goal is to minimize the risk to other hosts in the LAN. Ideally the host should perform only the FTP function, but if that is not possible it should at least not serve internal functions of the organization. The server should be behind a firewall but also isolated from the rest of the organization's network.

User management

The user entry for a restricted FTP account is made in the /etc/passwd file and is similar to a regular user entry with the exception of the shell specification which should be "*" or the null device, e.g.

ftpusername:x:uid:gid:Restricted FTP account:/usr/restricted_ftp:*

An entry in the file /etc/ftpusers must also be made allowing restricted access:

ftpusername restrict

This entry is important. Omitting the username will allow the user to move out of the rooted directory. Inserting the name without the word "restrict" will block FTP access.

Although one username and password can be shared among all of the users, security is better served by issuing a separate user ID and password to each authorized user. This approach gives the administrator more information for accounting and forensics if a security incident should occur.

Users will not be able to change passwords so the administrator must assign passwords and change them as required or requested. This requirement makes it much easier for the administrator to enforce strong password practices, but burdens her with the task of managing each user's password.

Directory tree

The administrator must exercise great care in setting up the FTP directory structure. The home directory contains not only the user directories, but also system directories in which binary, library, and text files reside. The access permission for these directories and files must be set to prevent unauthorized modification.

These system directories and their files are:

- bin – mode 555, owned by root, contains the ls command binaries with mode 111
- etc – mode 555, owned by root, contains modified passwd and group files with mode 444.
- lib32 – mode 555, owned by root, contains libraries, mode 555, for the ls command.

- dev – mode 555, owned by root and contains the device file zero, mode 444, used by one of the libraries

Ownership of the home directory and its permissions will be dictated by the requirements of information management, but the best approach is to make the directory unwritable by the users except possibly a designated file manager for the site.

Finally, a README file displayed automatically at login time should be created containing legal warnings to the user. This file is a requirement for building a case against an intruder should a penetration occur.

FTP daemon

Vendor supplied FTP implementations do not always comply with standard (RFC 959) FTP. The vendor documentation should be consulted, including the online "man" pages. The following discussion is based upon the ftp and ftpd man pages for SGI IRIX 6.5.19.

The FTP daemon is specified in /etc/inetd.conf. Certain options should be invoked to change the default values that may not be optimal for restricted FTP.

- l - Log each successful and failed login attempt. If specified twice the invocation of certain commands, such as "get" and "put", are logged.

- h – suppress printing of server hostname and version information in response to the "stat" command.

- t – set a timeout interval in seconds for an inactive session. The default is 15 minutes (900 seconds). This default is probably too long and increases the risk that the files will be accessible on an unattended terminal.

- u – Default file protection scheme (umask). Default value is 022. This mask can be adjusted as needed for more complicated access control situations. The user can also set a umask in an interactive session.

- p – Allow data transfers only if the client control channel IP address is the same as the client data channel IP address thus preventing the "port" command from redirecting the data to another client [5]

Security Issues

Restricted FTP shares most of the security vulnerabilities of the other types of FTP. These vulnerabilities lead to risks to the availability of the system and risks to the integrity and confidentiality of the data. The threat to the availability of the server is reduced for restricted FTP compared to anonymous FTP because the

set of authorized users is trusted. There is no open door for an abuser to easily enter the site so that abuses such as disk filling or drop-off sites described in [6] are not as likely to occur. On the other hand, the requirement for preserving privacy, that is confidentiality and integrity, is of great importance. Therefore, prevention of unauthorized access is a high priority activity. Equally critical to the preservation of privacy through secure authentication is the protection of the information as it transients the network.

Secure authentication and privacy on a public network implies encryption. Some form of authentication, command, and file encryption is required to have any assurance of maintaining privacy over the Internet. Unfortunately, standard FTP (RFC 959) does not provide for encryption of any kind. It performs cleartext transmission of authentication information, control commands, and data. As a result, in 1997 RFC 2228 was introduced [7] to provide security extensions.

RFC 2228

RFC 2228 incorporates a mechanism for negotiating a security association between the server and client. This association may include, separately or in combination, user authentication, control command integrity, and data privacy. Additional access control commands are provided to carry out the negotiations. These include:

- AUTH - Authentication/Security Mechanism – send an argument identifying a supported security mechanism. The server and client exchange information until a mechanism is agreed upon.

- ADAT – Authentication/Security Data – conveys additional information for negotiation of the parameters of the security mechanism agreed on by AUTH commands.

- PROT – Data Channel Protection Level – sends a code indicating the level of protection to be used on the data channel. These levels are:

  Clear – cleartext, no security protection.
  Safe – integrity protection, content has not been corrupted.
  Confidential – content not readable by unauthenticated user.
  Private – content is integrity and confidentiality protected.

- Control channel protection commands to further specify protected exchanges.

- Security enhanced RFC959 commands.

The RFC mandates compatibility with non-security-aware FTP by recognizing and accommodating for reply code 500 meaning an unrecognized command, namely AUTH. In such a case the session will appear to proceed as a standard FTP.

Options

Although the security extensions of this RFC solve the problem of securing authentication and data transfers, not all system vendors have implemented it. Gromek [8] provides a good discussion of alternatives for securing FTP. The first option is to replace standard FTP with a third party RFC2228 FTP on server and clients. The author gives some examples of available RFC 2228 compliant applications. Alternatively, proxy software may be installed on the server and client. Standard FTP then communicates through the secure proxy software. An example of a proxy application is Secure Shell (SSH) using its tunneling feature. SSH also features an FTP-like replacement but it is not compatible with FTP (see [9] for a thorough reference on SSH).

These options all have some disadvantages, one of which is that additional software must be installed on all of the clients as well as the server. Training of the users may also be necessary. There are real world situations where installing special software on all of the clients to be included in a restricted FTP environment and training their users is not feasible. Management must then decide to accept the increased risk of theft of sensitive information and other damage. There are some things that can be done to improve security in this situation.

Authentication

If no means of encryption for authentication is to be used then greater reliance must be placed on basic password management. Users must be issued sufficiently long strings of random or pseudo-random characters using upper and lower case letters, numbers, and special characters. Passwords must be changed more frequently, but this becomes a burden for the system administrator since the user cannot change the password himself. None of these practices avoids the problem of cleartext transmission of the password. Since the password can be easily captured an extra layer of difficulty for an intruder in making a connection to the server can be introduced through the use of a network application called tcp_wrapper [10] which is distributed with many implementations of UNIX. This program “wraps” around inetd, the network daemon to provide both access control and logging for the various services administered by inetd. Access to the FTP service can be specified by individual client host name or IP address. Thus, a restricted FTP user must be authenticated based on username, password, and client IP. In addition, detailed logging of the FTP access, both successful and failed, is available. A further advantage is that tcp_wrapper only needs to be installed on the server. An

intruder can defeat this mechanism by spoofing the client IP address, but it is more difficult than just network sniffing for a cleartext password. Another disadvantage is that the user cannot log in from an arbitrary client, nor can she log in from a host using dynamic IP addressing.

An alternative for reducing the risk of cleartext passwords is use of one-time passwords. In this method part or all of a user's password is time variant with a span measured in minutes or seconds. If an intruder captures a password it will not remain valid long enough for the intruder to take advantage of it. The disadvantage of this method is that each user must have a passcode generating device synchronized to the server. Timely distribution of the passcode device can be a problem.

Confidentiality

Since standard FTP does not provide for encryption of data during transmission, files will have to be encrypted before placing them in the FTP directory. Most UNIX implementations have an encryption/decryption package. SGI IRIX uses an algorithm similar to the German Enigma machine, using a single 256-element rotor. This is a private key system so the key must be sent in a secure fashion to the client. The client must have the same algorithm for decryption. Thus, only another IRIX host can easily handle these encrypted files without additional software. It is not likely that all of the clients accessing a restricted FTP server over the internet are going to be SGI's. Therefore, if encryption is to be used it should be an application ported to a large number of platforms, such as PGP, or other utilities using major cryptosystems like DES3, AES, RSA, or ECC [11]. Unfortunately, this approach resurrects the problem of insuring all of the clients have the appropriate software installed.

Integrity

A widely used method for assuring integrity is a hashing code such as MD5 which creates a unique "message digest", essentially a fingerprint of the file. MD5 is built into many systems. Other, older algorithms include checksum and Cyclic Redundancy Codes (CRC).

Management

Once a restricted FTP package has been selected, installed, and configured security assurance does not endure unless active management is carried out. All of the recommendations for good security management apply to maintaining an FTP site. One of the most useful practices is for the administrator to try to penetrate the site herself. She should use the same techniques that an intruder would to detect vulnerabilities. These would include scanning and exploitation of known vulnerabilities. Care should be exercised, however, in performing these

activities to avoid compromising the server or creating a denial-of-service incident.

One of the most crucial activities is to keep patches up to date. In the case of SGI a web site is maintained by the vendor to distribute patches and, for security related patches, does not require a support contract. Other vendors supply similar services. A number of organizations distribute security alerts by web sites or email subscription, many free of charge. These organizations include SANS, the Computer Emergency Response Team, Symantec, Internet Software Consortium, various government agencies, etc.

Although the server administrator has assigned passwords for restricted FTP rather than the user, it is worthwhile to periodically run a password cracking utility, such as Crack, to insure that in fact the passwords are strong enough to not succumb to this widely available utility.

Detailed logging should be carried out along with a regular review of the logs. A cron job can be submitted to frequently scan the logs for key words that would alert the administrator to a possible incident. There are third party tools such as "Swatch" that perform this function.

Finally, the administrator must maintain technical currency. He must stay up to date on security issues and particularly the activities of the "enemy". Knowledge of the latest developments in computing will help him make effective decisions towards building and operating secure systems.

Conclusion

"Out of the box" restricted FTP implementations generally are not secure. The best solution is to install an FTP incorporating the security extensions of RFC 2228. However, the security of standard FTP can be improved by careful configuration and the use of various open source and third party packages that provide network filtering and encryption.

References

1. Hunt, Craig. TCP/IP Network Administration, 2nd edition. Sebastopol, CA: O'Reilly, 1998

2. Postel, Jon and Joyce Reynolds. "File Transfer Protocol (FTP)" RFC959. Internet Engineering Task Force. October 1985. http://www.ietf.org/rfc/rfc959.txt

3. "Anonymous FTP Configuration Guidelines" CERT Coordination Center. May 4, 2001. http://www.cert.org/tech_tips/anonymous_ftp_config.html

4. “Restricting a Restricted FTP”. Packet Storm Project. 2001
http://packetstormsecurity.nl/0001-exploits/mi009en.htm

5. "Problems With The FTP PORT Command or Why You Don't Want Just Any PORT in a Storm" CERT Coordination Center. February 12, 1999.
http://www.cert.org/tech_tips/ftp_port_attacks.html

6. "Anonymous FTP Abuses" CERT Coordination Center. May 4, 2001.
http://www.cert.org/tech_tips/anonymous_ftp_abuses.html

7. Horowitz, M., and S. Lunt. “FTP Security Extensions” RFC 2228. Internet Engineering Task Force. October 1997
http://www.ietf.org/rfc/rfc2228.txt

8. Gromek, Mike. “Securing FTP Authentication” SANS Institute. February 12, 2002
http://www.sans.org/rr/protocols/sec_ftp.php

9. Barrett, Daniel and Richard Silverman. SSH, the Secure Shell. Sebastopol, CA. O’Reilly, 2001

10. Bush, Christopher. “Enhancing Network Security with tcp_wrapper” Sys Admin. 2001
http://www.samag.com/documents/s=1178/sam9905b/9905b.htm

11. Wynn, Bruce Alan and Michael Carpenter. “Cryptography Tools for the Systems Administrator” Sys Admin. 2001
http://www.samag.com/documents/s=1203/sam9706d/9706d.htm