Software Engineering Institute
Carnegie Mellon University

Problems With The FTP PORT Command or Why You Don't Want Just Any PORT in a Storm

CERT Division

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

http://www.sei.cmu.edu

REV-03.18.2016.0

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

This report was prepared for the

SEI Administrative Agent
AFLCMC/AZS
5 Eglin Street
Hanscom AFB, MA 01731-2100

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM17-0052

Table of Contents

1 Introduction

In the past few years, there have been ongoing discussions about problems related to the PORT command in the FTP protocol. These problems are based on the misuse of the PORT command in the FTP protocol.

2 The FTP Protocol

To understand these attacks, it is necessary to have a basic understanding of the FTP protocol [1].

A client opens a connection to the FTP control port (port 21) of an FTP server. So that the server will be later able to send data back to the client machine, a second (data) connection must be opened between the server and the client.

To make this second connection, the client sends a PORT command to the server machine. This command includes parameters that tell the server which IP address to connect to and which port to open at that address - in most cases this is intended to be a high numbered port on the client machine.

The server then opens that connection, with the source of the connection being port 20 on the server and the destination being the port identified in the PORT command parameters.

The PORT command is usually used only in the "active mode" of FTP, which is the default. It is not usually used in passive (also known as PASV [2]) mode. Note that FTP servers usually implement both modes, and the client specifies which method to use [3].

3 The FTP Bounce Attack

To conform with the FTP protocol, the PORT command has the originating machine specify an arbitrary destination machine and port for the data connection. However, this behavior also means that an attacker can open a connection to a port of the attacker's choosing on a machine that may not be the originating client.

Making this connection to an arbitrary machine for unauthorized purposes is the FTP bounce attack.

For illustrative purposes only, several examples of how attackers can use FTP bounce follow.

Port scanning

An attacker wishing to carry out a port scan against a site can do so from a third-party FTP server acting as a stage for the scan. The victim site sees the scan as coming from the FTP server rather than the true source (the FTP client).

Under some circumstances, this technique offers the attacker more benefits that just hiding the true source of the probe. When the intended victim site is on the same subnet as the FTP server, or when it does not filter traffic from the FTP server, the attacker can use the server machine as the source of the port scan rather than the client machine, thus managing to bypass access controls that might otherwise apply.

Bypassing basic packet filtering devices

An attacker may bypass a firewall (or other boundary protection measures) in certain network configurations.

For instance, assume that a site has its anonymous FTP server behind the firewall. Using the port scan technique above, an attacker determines that an internal web server at that site is available on port 8080, a port normally blocked by a firewall.

By connecting to the public FTP server at the site, the attacker initiates a further connection between the FTP server and an arbitrary port on a non-public machine at that site (for instance the internal web server at port 8080). As a result, the attacker establishes a connection to a machine that would otherwise be protected by the firewall.

Bypassing export restrictions

An example of how to bypass export restrictions was described by Hobbit in a posting to the bugtraq mailing list in 1995 [4]. This description is available from
ftp://avian.org/random/ftp-attack.

4 Bypassing Dynamic Packet Filtering Devices

Another problem involves client sites that have implemented firewalls that use dynamic packet filters to protect themselves. The sites are open to attack because the firewall trusts the information it receives.

In this example, the victim site houses all of its systems behind a firewall that uses dynamic packet filters. A person at the victim site browses web pages and downloads a Java applet constructed by the attacker. Without that person's knowledge, the Java applet then opens an outbound FTP connection to the attacker's machine. The applet then issues an FTP PORT command, instructing the server machine to open a connection to, say, the telnet port at some otherwise protected system behind the victim firewall.

Because the dynamic packet filtering firewall examines outbound packets to determine if any action is required on its part, it notes the PORT command and allows an incoming connection from the remote web server to the telnet port on the victim machine. This connection normally is not allowed by the firewall; it was allowed in this case because the PORT command was issued by the client.

Martin et al [5] discuss this particular attack, variations of it, and specific defense strategies.

5 Solutions

The example attacks in this tech tip demonstrate the core component of the vulnerability: the contents of the FTP PORT command are not trustworthy as they are under the control of a potential attacker. The FTP bounce example demonstrates what happens when a server trusts the information. The dynamic filter example demonstrates what happens when a firewall trusts the information.

Because the core element of the FTP bounce attack is required for RFC compliance, there is no clear-cut solution. An important point to remember, though, is that the RFC states that the feature must be present in the server software and usable to be RFC compliant. It does not state that the end user must actually be under constraint of using this feature.

FTP Server Software

The best solution to the FTP bounce problem from a security perspective is to ensure that your FTP server software cannot establish connections to arbitrary machines. However, sites that rely on the RFC-compliant behavior may find that implementing this solution will affect applications that they use. (We have not received any first-hand reports of such cases.) Consequently, many vendors offer solutions that allow the site offering the FTP service to make the choice that best suits them. Vendor implementations fall into three groups:

1. strict conformance with RFC functionality: The PORT command may be used to connect directly to a third-party machine, and this is the only functionality allowed. Some vendors who choose to maintain strict conformance have addressed this problem by modifying all other network services to reject connections originating from the FTP data port (port 20).
2. strict suppression of the PORT command: The PORT command may be used to connect to the originating client, and this is the only functionality allowed.
3. variable PORT command behavior: The PORT command may be used in either of the above two ways, with one way being the default. Switching between them is usually achieved with a command line parameter. You should be careful to verify which is the default.

You should be aware which category your server software falls into. Our recommendation is to use option 2, or option 3 with suppression enabled.

FTP Server Configuration

Some of the FTP bounce attacks described earlier rely on one or more server machines (depending on the attack) allowing uploaded files via FTP (usually anonymous FTP).

Your site should offer anonymous upload facilities only if it is absolutely necessary. Even then, you must carefully configure the incoming area. For more details, see "Anonymous FTP Configuration Guidelines" at http://www.cert.org/tech_tips/anonymous_ftp_config.html.

Note that these steps only repel attacks that rely on intermediate uploads. The steps are not effective against other attacks (such as a port scan).

Network Configuration

There are a few things to keep in mind when configuring your network boundaries (e.g., packet filtering routers and firewalls).

Sites should ensure that they carefully design their network topology so that effective traffic boundaries exist between systems that offer distinct levels of service. For instance, a site typically has an anonymous FTP service, web service, and an incoming electronic mail hub. The site uses good security practice by separating the machines that provide these external services from those that perform internal services. It is important to have strong network boundaries (preferably using firewalls) between these two sets of machines. In this way, even if an FTP server is vulnerable internal machines can be protected at the intervening network boundary.

For example, sites that have an FTP server that allows the PORT command to establish connections to third-party machines should block traffic between the FTP server and machines that offer services relying on hostname or IP address for authentication. Examples of such services are rlogin, rsh and NFS. While a firewall or filtering router should always prevent direct external access to such services, it should also filter traffic from an internal FTP server that behaves in this way. This prevents the FTP server being used as a relay machine to attack protocols with weak authentication mechanisms based on hostname or IP address.

There are several references which can assist you in configuring your network boundaries. For example, the CERT Coordination Center includes the following in their recommended reading list at

- http://www.cert.org/pub/other_sources/books.html
- Chapman, D. B., and Zwicky. E. D. *Building Internet Firewalls.* Sebastopol, CA: O'Reilly & Associates, Inc., 1995.
- Cheswick, William R., and Bellovin, Steven M. *Firewalls and Internet Security: Repelling the Wily Hacker.* New York: Addison-Wesley Publishing Company, 1994.

Sites using dynamic packet filtering firewalls may need to take additional steps to ensure that third-party PORT commands are blocked by the firewall. If you need to address this problem, we encourage you to check with your vendor to determine the steps you should take.

6 References

1. Postel, J., and J. Reynolds, "File Transfer Protocol," *STD 1, RFC 959*, USC/Information Sciences Institute, October 1985.
   Available electronically from ftp://ftp.isi.edu/in-notes/rfc959
2. Bellovin, S., "Firewall-Friendly FTP," *RFC 1579*, AT&T Bell Laboratories, February 1994.
   Available electronically from ftp://ftp.isi.edu/in-notes/rfc1579.txt
3. Cheswick, William R., and Bellovin, Steven M. *Firewalls and Internet Security: Repelling the Wily Hacker.* New York: Addison-Wesley Publishing Company, 1994.
4. Hobbit, "The FTP Bounce Attack," July 1995.
   Available electronically from ftp://avian.org/random/ftp-attack
5. Martin, David M., Rajagopalan, Sivaramakrishnan, and Rubin, Aviel D., "Blocking Java Applets at the Firewall," *The Proceedings of the 1997 Symposium on Network and Distributed Systems Security.* Available electronically from
   http://www.cs.bu.edu/techreports/96-026-java-firewalls.ps.Z

7 Acknowledgments

Thanks to Steve Bellovin for the technical sanity check. Thanks to Jeff Schiller for guidance on interpreting RFCs. Thanks also to Don Stokes for his technical input.

This document is available from: http://www.cert.org/tech_tips/ftp_port_attacks.html

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our website: http://www.cert.org/

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

NO WARRANTY
Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an "as is" basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

Conditions for use, disclaimers, and sponsorship information

Revision History

Apr 28, 1998	Corrected URLs for obtaining RFCs
Feb 13, 1998	Updates
Feb 12, 1999	Converted to new web format