Network Working Group
Internet Draft
draft-metz-spasv-00.txt

Craig Metz
The Inner Net
January 15, 1998

Short Passive (SPASV) Command for FTP

Status of this Memo

Abstract

RFC 1639[Pis94] documents experimental long port (LPRT) and long passive (LPSV) commands that many IP Version 6 implementations are using as the replacement for the PORT and PASV commands in FTP [PR85]. The author believes that this is the incorrect direction to be heading and that the replacement for PORT and PASV should carry less information instead of more.

The passive command (SPASV) is a replacement for the PASV command. It only carries port numbers and does not carry addresses. This makes it usable with IPv4 and IPv6. A benefit of not carrying addresses is that pure network address translators (NAT) do not have to do a search-and-replace on the TCP stream, which is an expensive operation. This also eliminates three-way FTP, which is a rarely used mode of operation that leaves most existing FTP servers wide open to the FTP Bounce Attack [Hob95]. Because the FTP PORT command is unfriendly to some kinds of firewall configurations [Bel94] and that unfriendliness is there to support three-way FTP, there is no replacement for the PORT command -- all transfers should use passive mode instead.

The author's inet6-apps kit (available on ftp.ipv6.inner.net and ftp.inner.net) includes a client and server that supports the current

version of these commands. Those FTP servers implement this command.

1. Introduction

The FTP protocol defines two commands that are used to control the addresses and ports used for data connections. The first is the PORT command, with syntax of:

PORT a1,a2,a3,a4,p1,p2<CRLF>

And a success response of 200.

The second is the PASV command, with a syntax of:

PASV<CRLF>

And a success response of:

227 Entering Passive Mode. a1,a2,a3,a4,p1,p2<CRLF>

These commands carry four bytes of address information to make a 32 bit IPv4 address. To carry IPv6 addresses, these commands must be changed. The FOOBAR approach is to use a long port command, with a syntax of:

LPRT af,hal,h1,h2,h3,h4...,pal,p1,p2...<CRLF>

And a success response of 200.

FOOBAR also defines a long passive command, with a syntax of:

LPSV<CRLF>

And a success response of:

228 Entering Long Passive Mode
(af, hal, h1, h2, h3,..., pal, p1, p2...)

The long port and long passive commands are a more flexible version of the original PORT and PASV commands that can handle longer addresses and port numbers. It has the same advantages and disadvantages as the original design. The author's opinion is that the replacements for PORT and PASV should modify this design to reflect current experiences and requirements.

The File Transfer Protocol was designed to support a "three-way" mode of operation in which a server can be directed to send a file to a system other than the client. This feature is easily abused to

create the FTP Bounce Attack. In his paper on this attack, [Hob95] rather bluntly characterizes the attack: "The mechanism used is probably well-known, but to date interest in detailing or fixing it seems low to nonexistent. ... It is chosen in an effort to make the reader sit up and notice that there are some really ill-conceived aspects of the standard FTP protocol." The fix for this attack is for servers to verify that the address sent in the PORT command matches the far address on the TCP control connection. Because the server's TCP stack must maintain the control connection address anyway and must check that the PORT argument matches, the address argument to the PORT command is redundant and only serves to create extra processing.

The address argument to the PORT command creates headaches for implementors of Network Address Translators (NATs) because that address information must be translated along with the addresses on packets. However, unlike other addresses on packets, the PORT command arguments are at the application layer, above the telnet protocol, which is in turn above TCP. NATs have to re-assemble the TCP stream and then decode the telnet protocol looking for the PORT command's arguments. Once found, these boxen have to synthesize a new stream. Changing the TCP stream, and possibly the length of the string, in transit is a very expensive operation. Two alternatives to doing this are:

1. To implement the search-and-replace in a faster but less correct manner making certain assumptions (the path many existing NAT implementors chose to take, and a dangerous one), which means that some commands can be missed.
2. To not put the address information in the PORT commands.

The short passive command makes life much easier for those implementing NATs because there is no address to be translated in those commands. Also, by removing the address information from the PORT commands, IPv4 and IPv6 addresses and other stream transports are naturally supported. Note that a form of NAT being considered for use in the transition to IPv6, the IPv6 to IPv4 translator, also benefits from the short port commands.

There is only a short passive command and not a short port command because the short port command's operation is somewhat odd in order to support three-way FTP, which is eliminated in the short commands. Most Internet protocols work by having a client open one or more connections to a server, which then returns data. FTP using the PORT command stands out as one of the few protocols that doesn't work this way; the client opens one connection to the server, and the server opens one or more connections back to the client. [Bel94] says about this, "this connection is set up by an active open from the FTP

server to the FTP client. However, this scheme does not work well with packet filter-based firewalls, which in general cannot permit incoming calls to random port numbers."

2. Syntax

The short passive command's syntax is:

SPASV<CRLF>

The success response to this command is:

229 Entering Passive Mode. <service><CRLF>

Where <service> is a printable service name in numeric form, as would be returned by the IPv6 BSD API's getnameinfo() function with the NI_NUMERICSERV flag set and suitable for passing to the POSIX p1003.1g getaddrinfo() function. Note that for IP versions 4 and 6, this will be the string representation of a decimal number between one and 65535.

The failure responses to this command are the same as for the PASV command in FTP.

3. Security Considerations

This document describes a protocol extension that, among other things, designs away a security hole in the current FTP protocol known as the FTP Bounce Attack and is more friendly to some kinds of firewall configurations. This extension is believed by the author to otherwise have the same security properties as the protocol commands (PORT and PASV) it replaces.

Acknowledgments

This work was done at the Center for High Assurance Computer Systems at the U.S. Naval Research Laboratory. This work was sponsored by the Information Security Program Office (PMW-161), U.S. Space and Naval Warfare Systems Command (SPAWAR) and the Computing Systems Technology Office, Defense Advanced Research Projects Agency (DARPA/CSTO). The author and his co-workers really appreciates their sponsorship of NRL's network security efforts and their continued support of IPsec development. Without that support, this document, among many others, would not exist.

Mike Allman suggested replacing the FTP-style decimal ports that were used in an earlier version of this draft with a more protocol-independent value.

D. J. Bernstein pointed out that only passive mode is needed if three-way FTP is eliminated.

References

[Bel94] S. Bellovin, "Firewall-Friendly FTP", RFC 1579, February 1994.

[Hob95] "*Hobbit*", "The FTP Bounce Attack", July 1995. (ftp.avian.org:/random/ftp-attack)

[Pis94] D. Piscitello, "FTP Operation Over Big Address Records (FOOBAR)", RFC 1639, June 1994.

[PR85] J. Postel and J. Reynolds, "File Transfer Protocol (FTP)", RFC 959, October 1985.

Disclaimer

The views and specification here are those of the author and are not necessarily those of his employer. His employer has not passed judgment on the merits, if any, of this work. The author and his employer specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Author's Address

Craig Metz
The Inner Net
Box 10314-1933
Blacksburg, VA 24062-0314
Phone: (DSN) 354-8590
E-mail: cmetz@inner.net