Abstract

The only FTP mode that works without changes through an IPv6-to-IPv4 translate is extended passive, introduced in 1998. However, many existing FTP servers don’t support this mode, making it impossible to
1. Introduction

[RFC0959] specifies two modes of operation for FTP: active mode, in which the server connects back to the client on port 20 or a client-provided port number, and active mode, where the server opens a port for the client to connect to. Without additional action, active mode doesn’t work through NATs or firewalls. And in both cases, an IPv4 address is specified, making both modes incompatible with IPv6. These issues were solved in [RFC2428], which specifies the EPSV (extended passive) mode that only specifies a port number and the EPRT (extended port) command which allows the client to supply an IPv6 address to the server.

A survey of 25 randomly picked and/or well-known FTP sites reachable over IPv4 showed that only 12 of them supported EPSV over IPv4. Additionally, only 2 of those 12 indicated that they supported EPSV in response to the FEAT command ([RFC2389]), while one supported EPSV but not FEAT. In 5 cases, issuing the EPSV command to the server led to a significant delay, in 3 cases followed by a control channel reset. It appears that in these cases, the server did support EPSV but a middlebox didn’t. All 25 servers were able to successfully complete a transfer in PASV mode as required by [RFC1123].

Based on the survey, an FTP ALG should be considered a necessary part of any NAT64 deployment. Since all servers in the survey supported PASV passive mode, NAT64 implementers SHOULD implement EPSV to PASV translation. NAT64 implementers MAY also implement EPRT to PORT translation. However, as many hosts reside behind firewalls, often unbeknownst to the FTP clients running on those hosts, active FTP is relatively likely to fail with or without translation.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Control channel translation

The NAT64 FTP ALG intercepts all sessions towards IPv4 destinations port 21. The FTP ALG implements the Telnet protocol ([RFC0854]) used
for control channel interactions to the degree necessary to interpret commands and responses and re-issue those commands and responses, optionally modifying them. Option negotiation attempts by the client except for those allowed by [RFC1123] SHOULD be rejected by the FTP ALG without relaying those attempts to the server. This avoids the situation where the client and the server negotiate options unknown to the FTP ALG.

If the client issues the AUTH command and the server responds with code 234 or 334, the client and server are negotiating [RFC2228] security mechanisms which are likely to be incompatible with the FTP ALG function. In this situation, the FTP ALG MUST switch to transparently forwarding all data on the control channel in both directions until the end of the control channel session.

4. EPSV to PASV translation

Although many IPv4 FTP servers support the EPSV command, some servers react adversely to this command, and there is no reliable way to detect in advance that this will happen. As such, a NAT64 FTP ALG SHOULD translate all occurrences of the EPSV command issued by the the client to the PASV command, and reformat a 227 response as a corresponding 229 response.

For instance, if the client issues EPSV, this is translated to the PASV command. If the server with address 192.0.2.31 then responds with:

227 Entering Passive Mode (192,0,2,31,237,19)

The FTP ALG reformats this as:

229 Entering Extended Passive Mode (|||60691|)

If the server’s 227 response contains an IPv4 address that doesn’t match the destination of the control channel, the FTP ALG SHOULD reply with:

425 Can’t open data connection.

It is important that the response is in the 4xx range to indicate a temporary condition.

5. EPRT to PORT translation

Should the IPv6 client issue an EPRT command, the FTP ALG MAY
translate this EPRT command to a PORT command. In that case, there are three possibilities: the address specified in the EPRT command is the client’s IPv6 address, it’s another IPv6 address or it’s an IPv4 address. If it’s an IPv6 address within the range that the translator is prepared to serve, even if it’s not the client’s address, the NAT64 selects an unused port number in combination with the IPv4 address used for the control channel towards the FTP server, and sets up a mapping from that transport address to the one specified by the client in the EPRT command. The PORT command is only issued towards the server once the mapping is created. Initially, the mapping is such that either any transport address or the FTP server’s IPv4 address with any port number is accepted as a source, but once the three-way handshake is complete, the mapping is narrowed to only match the negotiated TCP session.

If the address in the EPRT command is an IPv6 address that the NAT64 is not prepared to translate for, the EPRT command is passed along to the server unmodified. If the address in the EPRT command is an IPv4 address, the FTP ALG reformats the EPRT command to the equivalent PORT command without changing the transport address. In these cases, the NAT64 doesn’t create a mapping. This behavior retains compatibility with the server-to-server transfer option in FTP.

Note that there is the corner case where the client doesn’t specify either EPSV or EPRT because it wants to use active FTP on the default port. This case isn’t handled and will result in failure.

6. Timeouts

Wherever possible, control channels SHOULD NOT time out while there is an active data channel. A timeout of at least 30 seconds is recommended for mappings created by the FTP ALG that are waiting for initial packets.

7. IANA considerations

None.

8. Security considerations

In the majority of cases, FTP is used without further security mechanisms. This allows a passive attacker to obtain the login credentials, and an attacker that can modify packets to change the data transferred. However, FTP can be used with TLS in order to solve these issues. NAT64 translation and the FTP ALG don’t impact
the security issues in the former case nor the use of TLS in the latter case.

9. Normative References


Appendix A. Document and discussion information

The latest version of this document will always be available at http://www.muada.com/drafts/. Please direct questions and comments to the BEHAVE mailinglists or directly to the author.

Appendix B. Acknowledgement

Iljitsch van Beijnum is partly funded by Trilogy, a research project supported by the European Commission under its Seventh Framework Program.
Author’s Address

Iljitsch van Beijnum
IMDEA Networks
Avda. del Mar Mediterraneo, 22
Leganés, Madrid 28918
Spain

Email: iljitsch@muada.com