Network Working Group
Request for Comments: 3617
Category: Informational

E. Lear
Cisco Systems

October 2003

Uniform Resource Identifier (URI) Scheme and Applicability Statement for the Trivial File Transfer Protocol (TFTP)

Status of this Memo

Copyright Notice

Abstract

The Trivial File Transfer Protocol (TFTP) is a very simple TRIVIAL protocol that has been in use on the Internet for quite a long time. While this document discourages its continued use, largely due to security concerns, we do define a Uniform Resource Identifier (URI) scheme, as well as discuss the protocol's applicability.

1. Introduction

The Trivial File Transfer Protocol (TFTP) has been around for quite some time. Its common uses are to initially configure devices or to load new versions of operating system code [1]. As devices begin to adopt use of Uniform Resource Identifiers (URIs) and Uniform Resource Locators (URLs), for completeness we specify a way to reference files that is still quite common. Use of a URI is a convenient way to indicate underlying mechanism, server name or address, and file name.

WHILE WE DEFINE THE TFTP URI TYPE, WE STRONGLY RECOMMEND AGAINST THE CONTINUED USE OF TFTP, FOR REASONS LISTED IN SECTION 5 (amongst others). The definition of a URI merely allows tools that currently use protocols such as TFTP to have a standard name space and structure where one can understand the process used to resolve that name. Indeed it is hoped that the definition of this URI will ease transition to modern file transfer mechanisms.

2. Syntax of a TFTP URI

A TFTP URI has the following ABNF syntax [2]:

```
tftpURI         = "tftp://" host "/" file [ mode ]
mode            = ";"  "mode=" ( "netascii" / "octet" )
file            = *( unreserved / escaped )
host            = <as specified by RFC 2732 [3]>
unreserved      = <as specified in RFC 2396 [4]>
escaped         = <as specified in RFC 2396>
```

A TFTP URI specifies a file that is to be found or placed on a TFTP server. The "mode" option is an option indicating how the file is to be transferred. If left unspecified, the mode is assumed to be "octet". A third "mail" mode was deprecated at the time RFC 1350 was adopted, and is not specified.

2.1. Encoding Rules

Aside from syntax as described above, the TFTP protocol does not specify length limits to either file names or file sizes. In the case of file names, they may contain any character so long as those characters are properly escaped as described above.

3. Semantics and Operations

As previously stated the TFTP URI is a reference to a file. The allowed operations on a TFTP URI are read and write. When a TFTP URI is read the underlying mechanisms retrieve the named file via the TFTP protocol from the specified host with the optionally specified mode. When a TFTP URI is written the underlying mechanisms transmit a file via TFTP to a specified server to either the specified file using the optionally specified mode. No other operations are supported.

Note that it is not possible to retrieve file size information prior to retrieval, nor is it possible to determine file existence or permissions prior to transfer. Files transferred may or may not arrive intact, as there is no guarantee of reliability or even completeness. See the TFTP standard for more details. For more robust file transfer, consider using either FTP or HTTP [5, 6].

4. Examples

tftp://example.com/myconfigurationfile;mode=netascii

This example references file "myconfigurationfile" on server "example.com" and requests that the transfer occur in netascii mode.

tftp://example.com/mystartupfile

This file references file "mystartupfile" on server "example.com". The transfer should occur in octet mode, since no other mode was specified.

5. Security Considerations and Concerns about TFTP's use

Use of TFTP has been historically limited to those devices where a more full protocol stack is impractical due to either memory or CPU constraints. While this still may be the case with a toaster, it is unlikely to be the case for even the simplest piece of network support hardware, such as simple routers or switches. There are a myriad of reasons to use some protocol other than TFTP, only a few of which are listed below.

TFTP has no mechanism for access control within the protocol, and there is no protection from a man in the middle attack. Implementations are left to their own devices in this area. Because TFTP has no way to determine file sizes in advance, implementations should be prepared to properly check the bounds of transfers so that neither memory nor disk limitations are exceeded.

TFTP is not well suited to large files for the following reasons. TFTP has no inherent integrity check. There is no way to determine what one side sent is what the other received. There is no way to restart TFTP transfers from anywhere other than the beginning. TFTP is a lock step protocol. Only one packet may be in flight at any one time. There is no slow start or smart backoff mechanism in TFTP, but very simple timeouts.

TFTP is not well suited to file transfers across administrative domains. For one thing, TFTP utilizes UDP, and many NATs will not either support or allow TFTP transfers. More likely firewalls will prohibit transfers.

There are no caching semantics within TFTP. There is no safe way to cache information using the TFTP protocol.

In summary, use of TFTP is strongly discouraged except in the most limited of circumstances where memory and CPU are at the highest premium.

6. IANA Considerations

The IANA has registered the URL registration template found in Appendix A in accordance with RFC 2717 [7].

7. Acknowledgments

The author thanks Larry Masinter, Randy Presuhn, Phil Schafer, and Bill Fenner for their help in developing this document.

8. Intellectual Property Statement

Appendix A. Registration Template

URL scheme name: tftp
URL scheme syntax: Section 2
Character encoding considerations: Section 2
Intended usage: Section 1
Applications and/or protocols which use this scheme: [1]
Interoperability considerations: None
Security considerations: Section 5
Relevant publications: [1]
Contact: The author, Section 8
Author/Change Controller: IESG

References

[1] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, RFC 1350, July 1992.

[2] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[3] Hinden, R., Carpenter, B. and L. Masinter, "Format for Literal IPv6 Addresses in URL's", RFC 2732, December 1999.

[4] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[5] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[6] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[7] Petke, R. and I. King, "Registration Procedures for URL Scheme Names", BCP 35, RFC 2717, November 1999.

Author's Address

Eliot Lear
Cisco Systems, Inc.
170 W. Tasman Dr.
San Jose, CA 95134-1706

Phone: +1 (408) 527 4020
EMail: lear@cisco.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.