J. Twine
JRTwine Software, LLC
INTERNET-DRAFT
draft-twine-ftpmd5-00.txt
May, 2002

The 'MD5' and "MMD5" FTP Command Extensions

Status of This Document

Abstract

This document specifies two additions to the File Transfer Protocol (FTP). These additions (new Server commands) would give FTP Servers the ability to generate (or otherwise obtain) and return MD5 checksums for the files it has available for transfer.

It is the author's belief that this would provide a great benefit to the Internet community, because it would allow automated transfer agents, as well as Web Browsers and other "click-to-download" applications to be able to automatically verify the data of a downloaded file, and hence be able to detect any data tampering and/or corruption that may occurred while "on the wire", or possibly while the file was on the Server (a virus infection).

Copyright Notice

Comments

Comments should be directed to James R. Twine (jtwine@jrtwine.com).

Table of Contents

1. Introduction

This Draft is being distributed to members of the Internet community in order to solicit their reactions to the proposals contained in it.

2. Rational

FTP is still very much in use on the Internet. These days, some servers make available files that contain the checksums for some of the files that are available. These available checksums allow users to be able to verify the content of the files that they have downloaded.

However, this introduces some additional overhead: these MD5 checksums must be manually generated, put into a file, the file placed where it can be accessed. Then, users must manually download the file containing the checksum, generate an MD5 checksum from the file they just downloaded, and (usually) visually compare the two checksums to determine the validity of the file.

Having these tasks automated, by making the MD5 checksums available directly from the FTP Server proper, and having file-transfer implementations use them, alleviates some of the user intervention that would otherwise be required.

3. Server Requirements

FTP Servers would have to implement a new server-side command, called "MD5", this command would normally generate and return a MD5 for the specified file.

Optionally, the FTP Server could also implement the "MMD5" command, which is used to obtain MD5 checksums for multiple files using a single request.

(These commands impose no specific or additional syntax on the formatting of a filepath, they use the Server's existing conventions.)

The Server implementation is also free to use some form of caching to keep the generated MD5 checksums, so that the MD5 checksum values do not have to be regenerated over and over again when requested.

This also allows the Server implementations to maintain some level of security: the Server can expose administrative commands that regenerate the cache of MD5 checksums on command, thus allowing

for "known good" checksums to be kept, and would be insensitive to things like the file becoming corrupted or otherwise tampered with after the "known good" MD5 checksum was generated.

A Server implementation could even take that approach one step further: by generating additional MD5 checksums "on the fly" and comparing them to the "known good" values that were stored earlier, the Server would now have the ability to detect file corruption and/or tampering earlier than the user would normally discover.

The command would support a full or relative path, so that a directory change would not be necessary in order to obtain the MD5 checksum of a particular file. Of course, the command should normally be restricted to the directory tree and/or files that the connected user would normally have access to.

3.1 Command Format (MD5)

The "MD5" command is used to obtain a MD5 checksum for a single file, and is specified as follows:

```
MD5 [Filepath]
```

Possible responses to this command would normally include:

```
251 [FilePath] E67DED2886048D308532042B777D53CF
500 Command Not Recognized
502 Command Not Implemented
504 Command Not Implemented for the Specified Argument
```

(Note that the returned MD5 checksum is in UPPERCASE.)

A successful response of "251" would contain the specified filepath (verbatim) followed by a space (or some amount of whitespace), and then followed by the MD5 checksum value in ASCII format.

An error return of "500" would be for an obvious reason: the FTP Server does not recognize the "MD5" command.

An error return of "502" would be appropriate if the FTP Server regocnized the command, but did not support it, or the FTP Server administrator disabled it.

An error return of "504" would be appropriate if the user requested an MD5 checksum for a directory (for example).

3.1.1 MD5 Examples

This first example demonstrates a request for a MD5 checksum of a single file ("C>" is Client input, and "S>" is Server response):

```
C> MD5 filename.ext
```

```
        S> 251 filename.ext E67DED2886048D308532042B777D53CF
```

This second example demonstrates a request for a MD5 checksum of a directory:

```
C> MD5 ".."
S> 504 Command Not Implemented for the Specified Argument
```

This third example demonstrates a request for a MD5 checksum of a file using a relative path:

```
C> MD5 "../SomeDir/A File.txt"
S> 251 "../SomeDir/A File.txt" 604E67DED8D308B777D53CF532042288
```

3.2 Command Format (MMD5)

The "MMD5" command is used to obtain MD5 checksums for multiple files by a single request. Filepaths are comma separated, and are specified as follows (it is to be considered valid to specify a single filepath with with MMD5 command):

```
MMD5 [Filepath1], [Filepath2] [...]
```

Possible responses to this command would normally include:

```
252 [FilePath1] E67DED2886048D308532042B777D53CF,[Filepath2]
        308536048D20E67D77D53CFED28842B7 [...]
500 Command Not Recognized
502 Command Not Implemented
504 Command Not Implemented for the Specified Argument
```

A successful response of "252" would contain comma separated "groups" of MD5 checksum information. Each group would contain the specified filepath (verbatim) followed by a space (or some amount of whitespace) followed by the MD5 checksum value in ASCII format.

An error return of "500" would be the same as described for the "MD5" command.

An error return of "502" would be appropriate if the "MMD5" command was not implemented or disabled.

An error return of "504" would be the same as described form the "MD5" command, with this exception: of any of the specified filepaths were invalid, the server would return this error code (i.e. it would no MD5 checksums at all).

3.2.1 MMD5 Examples

This first example demonstrates a request for a MD5 checksum of a single file:

```
      C> MMD5 filename.ext
          S> 251 filename.ext E67DED2886048D308532042B777D53CF
```

This second example demonstrates a request for the MD5 checksums for two files:

```
C> MMD5 filename.ext, "../SomeDir/A File.txt"
S> 252 filename.ext E67DED2886048D308532042B777D53CF,
   "../SomeDir/A File.txt" 604E67DED8D308B777D53CF532042288
```

This third example demonstrates a request for the MD5 checksums of a file and a directory:

```
C> MD5 filename.ext, ".."
S> 504 Command Not Implemented for the Specified Argument
```

4. References

[1] Postel, J., Reynolds J., "Instructions to RFC Authors", RFC 2223, October 1997

[2] Postel, J., Reynolds J., "FILE TRANSFER PROTOCOL (FTP)", RFC 959, October 1958

[3] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992

[4] Various, "Guidelines to Authors of Internet-Drafts", http://www.ietf.org/ietf/1id-guidelines.txt

4. Author's Address

James R. Twine
JRTwine Software, LLC
379 Shirley Hill Road
Goffstown, NH, 03045
(USA)

Phone: +1 603-644-1307
EMail: jtwine@jrtwine.com