Network Working Group
Internet-Draft
Expires: December 3, 2005

J. Salowey
Cisco Systems
S. Suehring
June 2005

Uniform Resource Identifier (URI) Scheme for Secure File Transfer Protocol (SFTP) and Secure Shell (SSH) draft-ietf-secsh-scp-sftp-ssh-uri-03.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the Uniform Resource Identifiers used to locate resources for the Secure File Transfer Protocol (SFTP) and the Secure Shell (SSH) protocols. The document describes the generic syntax involved in URI definitions as well as specific definitions for each protocol. These specific definitions may include user credentials such as username and also may include other parameters such as host key fingerprint. In addition, security considerations

and examples are also provided within this document.

Table of Contents

1. Introduction

This document describes the Uniform Resource Identifiers (URIs) to be used with the Secure File Transfer Protocol (SFTP) [I-D.ietf-secsh-filexfer] and Secure Shell (SSH) [I-D.ietf-secsh-architecture] protocols.

2. General Syntax

A hierarchical URI shall consist of the scheme and the scheme specific portion separated by a colon ":" followed by the hierarchical part, as discussed in [RFC3986]. This specification uses the definitions "port", "host", "scheme", "userinfo", "path-empty", "path-abempty" and "authority" from [RFC3986]. This document follows the ABNF notation defined in [RFC2234].

2.1 Secure Shell (SSH) URI

The Secure Shell (SSH) scheme shall consist of the scheme name "ssh" followed by a colon ":" followed by hier-part defined in [RFC3986]. This URL does not designate a data object, but rather an interactive service. The SSH URI ABNF definition follows.

```
sshURI        =  "ssh:" hier-part
hier-part     =  "//" authority ( path-empty / path-abempty )
authority     =  [ [ userinfo-ssh ] "@" ] host [ ":" port ]
host          =  <as specified in [RFC3986]>
port          =  <as specified in [RFC3986]>
userinfo-ssh  =   [ userinfo ] [";" c-param *("," c-param)]
userinfo      =  <as specified in [RFC3986]>
path-empty    =  <as specified in [RFC3986]>
path-abempty  =  <as specified in [RFC3986]>
c-param       =  paramname "=" paramvalue
paramname     =  *( ALPHA / DIGIT / "-" / "." / ":" )
paramvalue    =  *( ALPHA / DIGIT / "-" / "." / ":" )
```

The first component of the scheme specific portion MAY include credentials (userinfo-ssh) consisting of a username followed by optional parameters. The convention of optionally including the password separated from the username by a ":" in the URI is NOT RECOMMENDED and is deprecated in accordance with [RFC3986].

One or more optional connection parameters (conn-parameters) may be specified within the userinfo section of the URI. These conn-parameters are separated from the userinfo by a semi-colon ";". The only connection parameter defined in this document is for the host-key fingerprint described in section Section 3.1. It is possible

that additional parameters be defined in the future. If a connection parameter is not understood it SHOULD be ignored.

If the userinfo or connection parameters are present the at-sign "@" shall precede the authority section of the URI. Optionally, the authority section MAY also include the port preceded by a colon ":". If the port is not included, the default port is assumed.

2.2 Secure File Transfer Protocol (SFTP) URI

The SFTP URL scheme is used to designate files and directory listings to retrieve on Internet hosts accessible using the SFTP protocol described in [I-D.ietf-secsh-filexfer]. For Secure File Transfer Protocol (SFTP), the scheme portion shall consist of the scheme name "sftp". SFTP URIs ABNF definition is given below.

```
sftpURI       = "sftp:" hier-part
hier-part     = "//" authority [ path ] [ ";" s-param *("," s-param)]
path          = <as specified in [RFC3986]>
authority     = [ userinfo-ssh "@" ] host [ ":" port ]
host          = <as specified in [RFC3986]>
port          = <as specified in [RFC3986]>
userinfo-ssh =  [ userinfo ] [";" c-param *("," c-param)]
userinfo      = <as specified in [RFC3986]>
c-param       =  paramname "=" paramvalue
paramname     =  *( ALPHA / DIGIT / "-" / "." / ":" )
paramvalue    =  *( ALPHA / DIGIT / "-" / "." / ":" )
s-param       = paramname "=" paramvalue
```

The authority portion of the SFTP URL is the same as for the SSH URL defined in section Section 2.1. The URIs for SFTP are hierarchical URIs where each component of the path consists of path elements separated by a '/'. This formatting is meant to represent the path information as in Section 5 of [I-D.ietf-secsh-filexfer]. If a path starts with a %2F (a URL-encoded "/") then it is relative to the root of the file system. Paths starting with any other character are relative to the user's home or default directory. Note that the characters "/" and ";" are reserved and must be encoded. Path segments SHOULD be represented in the UTF-8 [RFC3629] character set and clients SHOULD NOT disable UTF-8 translation on the server with the filename-translation-control extension. The shortest valid UTF-8 encoding of the UNICODE data MUST be used. Note that dot segments "." and ".." are only interpreted within the URI path hierarchy and are removes as part of the URL resolution process defined in [RFC3986].

Following the path additional sftp specific parameters may be

specified. These are described in section Section 3.2. It is possible that additional parameters be defined in the future. If a sftp parameter is not understood it SHOULD be ignored.

3. Parameters

3.1 SSH connection parameters

The following parameters are associated with an SSH connection and are applicable to SSH and SFTP. All parameters are optional and MUST NOT overwrite configured defaults. Individual parameters are separated by a comma (",").

fingerprint

The fingerprint parameter contains the fingerprint of the host key for the host specified in the URL. The fingerprint is encoded as host-key-alg-fingerprint. Host-key-alg is host public key algorithm defined in [I-D.ietf-secsh-transport] and the fingerprint format is [I-D.ietf-secsh-publickeyfile]. For use in a URI, the fingerprint shall use a single dash "-" as a separator

instead of the colon ":" as described in [I-D.ietf-secsh-publickeyfile]. This parameter MUST NOT overwrite a key that is already configured for the host. The fingerprint MAY be used to validate the authenticity of the host key if the URL was obtained from an authenticated source with its integrity protected. If this parameter is not included then the validity of the host key is validated using another method. See Security Considerations section for additional considerations. There MUST be only one fingerprint parameter per host-key-alg for a given URL.

3.2 SFTP Parameters

The SFTP parameters determine how to handle the file transfer character translation. Additional parameters MAY be used.

typecode

The typecode identifies the type of file which determines how it will be treated. The name for the typecode attribute is "type". The value "i" indicates that a file should be transmitted without character conversion performed. The value "a" indicates that the file should be opened with the SSH_FXF_ACCESS_TEXT_MODE flag set so it is converted to the canonical newline convention in use. The value "d" indicates that the path is a directory and should be opened using SSH_FXP_OPENDIR.

4. Examples

The following section shows basic examples of URLs for each protocol. This section should not be considered to include all possible combinations of URLs for each protocol.

An SSH connection to the host host.example.com on the standard port using username user.

```
ssh://user@host.example.com
```

An SSH connection to the host host.example.com on port 2222 using username user.

```
ssh://user@host.example.com:2222
```

An SSH connection to the host having the specified host-key fingerprint at host.example.com on the standard port using username user.

```
ssh://user;fingerprint=ssh-dss-c1-b1-30-29-d7-b8-de-6c-97-
     77-10-d7-46-41-63-87@host.example.com
```

Retrieve file.txt from the user's home directory on the host at host.example.com using SFTP using username user.

```
sftp://user@host.example.com/file.txt
```

Retrieve file.txt from the absolute path /dir/path on the host at host.example.com using SFTP using username user.

```
sftp://user@host.example.com/%2Fdir/path/file.txt
```

Retrieve the directory listing of user's home directory on the host having the specified host-key fingerprint at host.example.com using SFTP.

```
sftp://user;fingerprint=ssh-dss-c1-b1-30-29-d7-b8-de-6c-97-
      77-10-d7-46-41-63-87@host.example.com:2222/;type=d
```

5. IANA Considerations

This document provides the information required in the URL registration template in accordance with [RFC2717].

6. Security Considerations

Passwords SHOULD NOT be included within the URI it should be noted

that doing so poses a security risk. Since URIs are usually sent in the clear with no encryption or other security, any password or other credentials included in the userinfo could be seen by a potential attacker.

Although the host-key fingerprint is not confidential information, care must be taken in handling fingerprints associated with URIs because URIs transmitted or stored without protection may be modified

by an attacker. In general an implementation cannot determine the source of a URI so a fingerprint received in a URI should have no more trust associated with it than a raw public key received in the SSH protocol itself. If a locally configured key exists for the server already it MUST NOT be automatically overwritten with information from the URI. If the host is unknown then the implementation should treat the fingerprint received with the same caution that it does with any unknown public key. The client MAY offer the fingerprint and URI for external validation before allowing a connection based on this information. If the client chooses to make a connection based on the URI information and it finds that the public key in the URI and the public key offered by the server do not match then it SHOULD provide a warning and provide a means to abort the connection. Sections 4.1 and 9.2.4 of [I-D.ietf-secsh-architecture] provide a good discussion of handling public keys received in the SSH protocol.

7. Acknowledgements

Ben Harris has provided much useful feedback in the preparation of this document.

8. References

8.1 Normative References

[I-D.ietf-secsh-architecture]
Ylonen, T. and C. Lonvick, "SSH Protocol Architecture", draft-ietf-secsh-architecture-22 (work in progress), March 2005.

[I-D.ietf-secsh-filexfer]
Galbraith, J., "SSH File Transfer Protocol", draft-ietf-secsh-filexfer-09 (work in progress), June 2005.

[I-D.ietf-secsh-publickeyfile]
Galbraith, J. and R. Thayer, "SSH Public Key File Format", draft-ietf-secsh-publickeyfile-09 (work in progress),

August 2005.

[I-D.ietf-secsh-transport]
Lonvick, C., "SSH Transport Layer Protocol", draft-ietf-secsh-transport-24 (work in progress), March 2005.

[RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[RFC2717] Petke, R. and I. King, "Registration Procedures for URL Scheme Names", BCP 35, RFC 2717, November 1999.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

8.2 Informative References

[RFC2718] Masinter, L., Alvestrand, H., Zigmond, D., and R. Petke, "Guidelines for new URL Schemes", RFC 2718, November 1999.

[RFC3305] Mealling, M. and R. Denenberg, "Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations", RFC 3305, August 2002.

Authors' Addresses

Joseph Salowey
Cisco Systems
2901 3rd Ave
Seattle, WA 98121
US

Email: jsalowey@cisco.com

Steve Suehring
PO BOX 1033
Stevens Point, WI 54481
US

Email: suehring@braingia.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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