Network Working Group
Internet-Draft
Expires: August 5, 2006

J. Salowey
Cisco Systems
S. Suehring

February 1, 2006

Uniform Resource Identifier (URI) Scheme for Secure File Transfer Protocol (SFTP) and Secure Shell (SSH)
draft-ietf-secsh-scp-sftp-ssh-uri-04.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) [RFC4251] 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 [RFC4234].

3. Secure Shell (SSH) URI

This section describes the SSH URI and contains the information necessary to register the URI according to the template in [I-D.hansen-2717bis-2718bis-uri-guidelines].

3.1 Scheme Name

The Secure Shell scheme name is "ssh".

3.2 Status

The requested status of the SSH URI is "permanent".

3.3 URI Scheme Syntax

The Secure Shell (SSH) scheme shall consist of the scheme name "ssh" followed by a colon ":" followed by hier-part defined in [RFC3986]. The SSH URI ABNF definition follows.

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

The following reserved characters from [RFC3986] are used as delimiters within the SSH URI: ";", ",", ":", and "=" . They must not be escaped when used as delimiters and must be escaped when the appear in other uses.

3.4 URI Semantics

The intended usage of the SSH URI is to establish an interactive SSH terminal session with the host defined in the authority portion of the URI. The only operation defined for the URI is to establish an SSH terminal session with a remote host.

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 ":". The host SHOULD be a non-empty string. If the port is not included, the default port is assumed.

The ssh-info portion of the URI MAY include credentials consisting of a username followed by optional parameters. The convention of 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 (c-param) 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 5.1. It is possible that additional parameters be defined in the future. If a connection parameter is not understood it SHOULD be ignored.

The SSH URI does not define a usage for a non-empty path element. If a non-empty path element is included in an SSH URI then it SHOULD be ignored.

3.5 Encoding Considerations

The encoding of the "host" portion of the URI is as defined in [RFC3986]. The encoding of the connection parameters is described in Section 5.1

3.6 Protocols using this URI scheme

This URI scheme is used by the SSH protocol version 2 defined in [RFC4251].

3.7 Security Considerations

See Section 8.

3.8 Contact

This document is a product of the SSH working group.

4. Secure File Transfer Protocol (SFTP) URI

This section describes the Secure File Transfer protocol URI and contains the information necessary to register the URI according to the template in [I-D.hansen-2717bis-2718bis-uri-guidelines].

4.1 Scheme Name

The Secure File Transfer Protocol (SFTP) scheme name is "sftp".

4.2 Status

The requested status of the SFTP URI is "permanent".

4.3 URI Scheme Syntax

The SFTP URI scheme shall consist of the scheme name "sftp" followed by a colon ":" followed by hier-part defined in [RFC3986]. The SFTP URI ABNF definition follows.

```
sftpURI        =  "sftp:" hier-part
hier-part      =  "//" authority path [";" s-param *("," s-param)]
path           =  path-abempty
path-abempty   =  <as specified in [RFC3986]>
authority      =  [ ssh-info "@" ] host [ ":" port ]
host           =  <as specified in [RFC3986]>
port           =  <as specified in [RFC3986]>
ssh-info       =  [ 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 part is the same as that defined in the SSH scheme. The following reserved characters from [RFC3986] are used as delimiters within the SFTP URI: ";", ",", ":", "=" and "/". They must not be escaped when used as delimiters and must be escaped when the appear in other uses.

4.4 URI Semantics

The intended usage of the SFTP URI is to retrieve the contents of a file or directory listing. The only operation defined for the URI is this "GET" operation.

The authority portion of the SFTP URL is the same as for the SSH URL defined in Section 3.4. 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]. The SFTP implementation determines where the root of the path in the URI is. It is RECOMMENDED that path be interpreted as an absolute path from the root of the file system. An implementation MAY use the tilde ("~") character as the first path element in the path to denote a path relative to the user's home directory. Note that dot segments "." and ".." are only interpreted within the URI path hierarchy and are removed 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 5.2. It is possible that additional parameters be defined in the future. If a sftp parameter is not understood it SHOULD be ignored.

4.5 Encoding Considerations

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. The encoding of the "host" portion of the URI is as defined in [RFC3986]. The encoding of the connection parameters is described in Section 5.1 and the encoding of SFTP parameters is described in Section 5.2.

4.6 Protocols using this URI scheme

This URI scheme is used by the SFTP protocol defined in [I-D.ietf-secsh-filexfer].

4.7 Security Considerations

The SFTP URI retrieves data from a remote host. Even though the connection is secured by SFTP care should be taken in handling and processing data retrieved from potentially unknown sources to avoid malicious content from an attacker that may have been placed on the host. For additional security considerations see Section 8.

4.8 Contact

This document is a product of the SSH working group.

5. Parameters

5.1 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 [RFC4253] 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 host key is validated using another method. See Security Considerations section for additional considerations. There MUST be only one fingerprint parameter present in a given URL.

5.2 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.

6. 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. This example assumes that the implementation supports the indication of a path relative to the home directory using a leading tilde.

```
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/dir/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
```

7. IANA Considerations

Section 3 and Section 4 provide the information required in the URL registration template in accordance with [I-D.hansen-2717bis-2718bis-uri-guidelines].

8. Security Considerations

Passwords SHOULD NOT be included within the URI as doing so poses a security risk. 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 fingerprint 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 [RFC4251] provide a good discussion of handling public keys received in the SSH protocol.

9. Acknowledgements

Ben Harris, Tom Petch and the members of the SSH working group have provided much useful feedback in the preparation of this document.

10. References

10.1 Normative References

[I-D.ietf-secsh-filexfer]
Galbraith, J. and O. Saarenmaa, "SSH File Transfer Protocol", draft-ietf-secsh-filexfer-12 (work in progress), January 2006.

[I-D.ietf-secsh-publickeyfile]
Galbraith, J. and R. Thayer, "SSH Public Key File Format", draft-ietf-secsh-publickeyfile-11 (work in progress), January 2006.

[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.

[RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC4251] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[RFC4253] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006.

10.2 Informative References

[I-D.hansen-2717bis-2718bis-uri-guidelines]
Hansen, T., "Guidelines and Registration Procedures for new URI Schemes", draft-hansen-2717bis-2718bis-uri-guidelines-06 (work in progress), October 2005.

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

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

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