Secure Shell Working Group
Internet-Draft
Expires: September 1, 2006

O. Saarenmaa
F-Secure
J. Galbraith
VanDyke Software

February 28, 2006

X.509 authentication in SSH
draft-ietf-secsh-x509-03.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 1, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document specifies how X.509 certificates and signatures are used within the Secure Shell protocol for user and server authentication.

Table of Contents

1. Introduction

The Secure Shell protocol can use public keys for both server and user authentication. However, particularly for server authentication, plain public keys lack a good method of verifying that the the key provided really does belong to the host asserting ownership. X.509v3 certificates can address this problem in environments where a PKI infrastructure is available.

2. Conventions Used in This Document

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. Certificate validation

Implementations are expected to follow the basic certificate and certificate path validation guidelines defined in [RFC3280]. This document does not define any new X.509 certificate extensions.

Users deploying certificates have often had little control over the capabilities of CAs available to them. Implementations of this specification MAY include configuration knobs to disable checks required by this specification in order to permit use with inflexible and/or noncompliant CAs. Before disabling any checks the administrators and users need to understand the purposes of those checks as well as the security implications that may raise when they are disabled.

3.1. Certificate Extensions

Implementations MUST recognize the following extensions: BasicConstraints, KeyUsage, and SubjectAltName. Implementations also MUST be able to handle all other extensions that have been marked critical or reject the certificate.

3.1.1. ExtendedKeyUsage

Certificates meant for use within the SSH protocol SHOULD NOT include the ExtendedKeyUsage extension. If the certificates require an EKU extension because of use in another protocol or application, it is RECOMMENDED to also specify the anyExtendedKeyUsage keyPurposeID [RFC3280].

Nevertheless, this document defines several ExtendedKeyUsage

keyPurposeID that MAY be used to limit a certificate's use. These are id-kp-ssh-server, for use in server certificates, id-kp-ssh-client for use in client (user) certificates, and id-kp-ssh-clientHostbased for use in server certificates that can be used with hostbased authentication [RFC4252]. The object identifiers are listed below:

```
id-kp-ssh-server OBJECT IDENTIFIER
  ::= { 1.3.6.1.4.1.2213.15.1.1 }
id-kp-ssh-client OBJECT IDENTIFIER
  ::= { 1.3.6.1.4.1.2213.15.1.2 }
id-kp-ssh-clientHostbased OBJECT IDENTIFIER
  ::= { 1.3.6.1.4.1.2213.15.1.3 }
```

3.2. Server Authentication

Implementations MUST validate the server host certificates by matching the server's fully qualified domain name [RFC1034] against the certificate's subjectAltName extension's dNSName entries. If the certificate does not contain dNSName subjectAltName extensions, the (most specific) Common Name field in the certificate Subject MUST be used. This is similar to host validation in HTTP Over TLS [RFC2818].

3.3. User Authentication

No constraints are placed on the presence of user account information in the certificates used for user authentication. The mapping of user certificates to user accounts is left as an implementation choice and configuration issue for the implementors and deployers.

4. Use in SSH Protocol

This document defines three new key formats which are in the form "x509v3-sign*". Each of the formats encodes the key type name in the beginning of the key blob.

4.1. x509v3-sign

This is the most flexible key and signature format defined by the document. It is RECOMMENDED that implementations prefer this algorithm over the two other x509v3-sign* algorithms that this document defines and may be supported. This format supports multiple certificates in a chain as well as including OCSP-responses [RFC2560] along with the certificate data. It also supports multiple different hash algorithms for signatures. Keys using this format are encoded as follows:

```
string "x509v3-sign"
uint32  number of certificates
string[1..] DER encoded X.509v3 certificate data
uint32  number of ocsp responses
string[0..] OCSP response data
```

The first certificate in the list MUST be the end-entity one, and any other certificates MUST be part of the end-entity certificate's path.

Signatures are encoded as follows:

```
string "x509v3-sign"
string hash algorithm OID
string signature data
```

Possible hash algorithms include, but are not limited to, SHA1 (1.3.14.3.2.26) [FIPS-180-2], SHA256 (2.16.840.1.101.3.4.2.1) [FIPS-180-2], MD5 (1.2.840.113549.2.5) [RFC1321] and RIPEMD160 (1.3.36.3.2.1) [RIPEMD-160].

4.2. x509v3-sign-rsa-sha1

Certificates that use the RSA public key algorithm MAY use the "x509v3-sign-rsa-sha1" key format. This key type uses the following format:

```
string   "x509v3-sign-rsa-sha1"
string   DER encoded X.509v3 certificate data
```

Signing using this key format, uses the certificate's private key, in exactly the same manner specified for "ssh-rsa" public keys in [RFC4253]. That is to say, signing and verifying using this key format is performed according to the RSASSA-PKCS1-v1_5 scheme in [RFC3447] using the SHA-1 hash [FIPS-180-2].

The signature format for x509v3-sign-rsa-sha1 certificates is the "ssh-rsa" signing format specified in [RFC4253]. This format is as follows:

```
string    "ssh-rsa"
string    rsa_signature_blob
```

The value for 'rsa_signature_blob' is encoded as a string containing s (which is an integer, without lengths or padding, unsigned and in network byte order).

4.3. x509v3-sign-dss-sha1

Certificates that use the DSA public key algorithm MAY use the "x509v3-sign-rsa-sha1" key format. This key type uses the following format:

```
string   "x509v3-sign-dss-sha1"
string   DER encoded X.509v3 certificate data
```

Signing and verifying using this key format, uses the certificate's private key, in exactly the same manner specified for "ssh-dss" public keys in [RFC4253]. That is to say, signing and verifying using this key format is done according to the Digital Signature Standard [FIPS-186-2] using the SHA-1 hash [FIPS-180-2].

The signature format for x509v3-sign-dss-sha1 certificates is the "ssh-dss" signing format specified in [RFC4253]. This format is as follows:

```
string    "ssh-dss"
string    dss_signature_blob
```

The value for 'dss_signature_blob' is encoded as a string containing r followed by s (which are 160-bit integers, without lengths or padding, unsigned and in network byte order).

5. Implementation Considerations

Implementations should be careful when using X.509v3 certificates as hostkeys. If the peer does not implement the required algorithms to validate both the end-entity certificate and all certificates in the chain, it MUST disconnect. There is no way to renegotiate the key during key exchange.

This is especially true when using the "x509v3-sign" key type, since in this case the peer has no knowledge whatsoever of required algorithms. The peer might also refuse a "x509v3-sign" key if the required intermediate certificates and OCSP responses are not included.

6. IANA Considerations

This document reserves all key types beginning with "x509v3-sign" in the SSH publickey type registry.

This document specifically adds "x509v3-sign-rsa-sha1", "x509v3-sign-

dss-sha1", and "x509v3-sign" to the SSH publickey type registry.

This document adds "x509v3-sign-rsa" and "x509v3-sign-dss" to the SSH publickey type registry as "poisoned" by historical use.

7. Security Considerations

PKI is an extremely complex topic, and care must be taken by both implementors and deployers to understand the complex interactions involved.

This document suggests that validation of the ExtendedKeyUsage extension MAY be disabled by configuration in the implementations. Disabling validation of other extensions such as KeyUsage or BasicConstraints MUST NOT be done, as that might lead into invalid trust paths being established.

Implementations should carefully validate the certificate, including but not limited to, certificate expiration, certificate signature, certification revocation status etcetera. Implementations must also be careful to validate all these properties of all certificates in the path leading to a trust anchor. For more information implementors should refer to [ITU.X509.2000] and [RFC3280].

8. References

8.1. Normative References

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[RFC4252] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Authentication Protocol", RFC 4252, January 2006.

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

[FIPS-180-2] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", Federal Information Processing Standards Publication 180-2, August 2002.

[FIPS-186-2] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", Federal Information Processing Standards Publication 186-2, January 2000.

[ITU.X509.2000] International Telecommunications Union, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", ITU-T Recommendation X.509, ISO Standard 9594-8, March 2000.

[RIPEMD-160] Dobbertin, H., Bosselaers, A., and B. Preneel, "RIPEMD-160: A Strengthened Version of RIPEMD", April 1996.

8.2. Informative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

Trademark notice

"ssh" is a registered trademark in the United States and/or other countries.

Authors' Addresses

Oskari Saarenmaa
F-Secure
Tammasaarenkatu 7
PL 24
Helsinki 00181
FI

Email: oskari.saarenmaa@f-secure.com

Joseph Galbraith
VanDyke Software
4848 Tramway Ridge Blvd
Suite 101
Albuquerque, NM 87111
US

Phone: +1 505 332 5700
Email: galb-list@vandyke.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

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

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