Network Working Group
Request for Comments: 4255
Category: Standards Track

J. Schlyter
OpenSSH
W. Griffin
SPARTA

January 2006

Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints

Status of This Memo

Copyright Notice

Abstract

This document describes a method of verifying Secure Shell (SSH) host keys using Domain Name System Security (DNSSEC). The document defines a new DNS resource record that contains a standard SSH key fingerprint.

Table of Contents

1. Introduction

The SSH [6] protocol provides secure remote login and other secure network services over an insecure network. The security of the connection relies on the server authenticating itself to the client as well as the user authenticating itself to the server.

If a connection is established to a server whose public key is not already known to the client, a fingerprint of the key is presented to the user for verification. If the user decides that the fingerprint is correct and accepts the key, the key is saved locally and used for verification for all following connections. While some security-conscious users verify the fingerprint out-of-band before accepting the key, many users blindly accept the presented key.

The method described here can provide out-of-band verification by looking up a fingerprint of the server public key in the DNS [1][2] and using DNSSEC [5] to verify the lookup.

In order to distribute the fingerprint using DNS, this document defines a new DNS resource record, "SSHFP", to carry the fingerprint.

Basic understanding of the DNS system [1][2] and the DNS security extensions [5] is assumed by 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 RFC 2119 [3].

2. SSH Host Key Verification

2.1. Method

Upon connection to an SSH server, the SSH client MAY look up the SSHFP resource record(s) for the host it is connecting to. If the algorithm and fingerprint of the key received from the SSH server match the algorithm and fingerprint of one of the SSHFP resource record(s) returned from DNS, the client MAY accept the identity of the server.

2.2. Implementation Notes

Client implementors SHOULD provide a configurable policy used to select the order of methods used to verify a host key. This document defines one method: Fingerprint storage in DNS. Another method defined in the SSH Architecture [6] uses local files to store keys for comparison. Other methods that could be defined in the future might include storing fingerprints in LDAP or other databases. A

configurable policy will allow administrators to determine which methods they want to use and in what order the methods should be prioritized. This will allow administrators to determine how much trust they want to place in the different methods.

One specific scenario for having a configurable policy is where clients do not use fully qualified host names to connect to servers. In this scenario, the implementation SHOULD verify the host key against a local database before verifying the key via the fingerprint returned from DNS. This would help prevent an attacker from injecting a DNS search path into the local resolver and forcing the client to connect to a different host.

2.3. Fingerprint Matching

The public key and the SSHFP resource record are matched together by comparing algorithm number and fingerprint.

The public key algorithm and the SSHFP algorithm number MUST match.

A message digest of the public key, using the message digest algorithm specified in the SSHFP fingerprint type, MUST match the SSHFP fingerprint.

2.4. Authentication

A public key verified using this method MUST NOT be trusted if the SSHFP resource record (RR) used for verification was not authenticated by a trusted SIG RR.

Clients that do validate the DNSSEC signatures themselves SHOULD use standard DNSSEC validation procedures.

Clients that do not validate the DNSSEC signatures themselves MUST use a secure transport (e.g., TSIG [9], SIG(0) [10], or IPsec [8]) between themselves and the entity performing the signature validation.

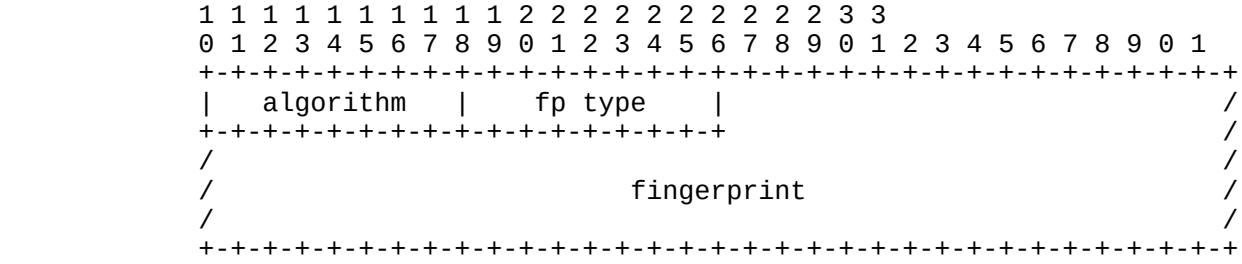
3. The SSHFP Resource Record

The SSHFP resource record (RR) is used to store a fingerprint of an SSH public host key that is associated with a Domain Name System (DNS) name.

The RR type code for the SSHFP RR is 44.

3.1. The SSHFP RDATA Format

The RDATA for a SSHFP RR consists of an algorithm number, fingerprint type and the fingerprint of the public host key.



3.1.1. Algorithm Number Specification

This algorithm number octet describes the algorithm of the public key. The following values are assigned:

Value	Algorithm name
0	reserved
1	RSA
2	DSS

Reserving other types requires IETF consensus [4].

3.1.2. Fingerprint Type Specification

The fingerprint type octet describes the message-digest algorithm used to calculate the fingerprint of the public key. The following values are assigned:

Value	Fingerprint type
0	reserved
1	SHA-1

Reserving other types requires IETF consensus [4].

For interoperability reasons, as few fingerprint types as possible should be reserved. The only reason to reserve additional types is to increase security.

3.1.3. Fingerprint

The fingerprint is calculated over the public key blob as described in [7].

The message-digest algorithm is presumed to produce an opaque octet string output, which is placed as-is in the RDATA fingerprint field.

3.2. Presentation Format of the SSHFP RR

The RDATA of the presentation format of the SSHFP resource record consists of two numbers (algorithm and fingerprint type) followed by the fingerprint itself, presented in hex, e.g.:

```
host.example.  SSHFP 2 1 123456789abcdef67890123456789abcdef67890
```

The use of mnemonics instead of numbers is not allowed.

4. Security Considerations

Currently, the amount of trust a user can realistically place in a server key is proportional to the amount of attention paid to verifying that the public key presented actually corresponds to the private key of the server. If a user accepts a key without verifying the fingerprint with something learned through a secured channel, the connection is vulnerable to a man-in-the-middle attack.

The overall security of using SSHFP for SSH host key verification is dependent on the security policies of the SSH host administrator and DNS zone administrator (in transferring the fingerprint), detailed aspects of how verification is done in the SSH implementation, and in the client's diligence in accessing the DNS in a secure manner.

One such aspect is in which order fingerprints are looked up (e.g., first checking local file and then SSHFP). We note that, in addition to protecting the first-time transfer of host keys, SSHFP can optionally be used for stronger host key protection.

If SSHFP is checked first, new SSH host keys may be distributed by replacing the corresponding SSHFP in DNS.

If SSH host key verification can be configured to require SSHFP, SSH host key revocation can be implemented by removing the corresponding SSHFP from DNS.

As stated in Section 2.2, we recommend that SSH implementors provide a policy mechanism to control the order of methods used for host key verification. One specific scenario for having a configurable policy is where clients use unqualified host names to connect to servers. In this case, we recommend that SSH implementations check the host key against a local database before verifying the key via the fingerprint returned from DNS. This would help prevent an attacker from injecting a DNS search path into the local resolver and forcing the client to connect to a different host.

A different approach to solve the DNS search path issue would be for clients to use a trusted DNS search path, i.e., one not acquired through DHCP or other autoconfiguration mechanisms. Since there is no way with current DNS lookup APIs to tell whether a search path is from a trusted source, the entire client system would need to be configured with this trusted DNS search path.

Another dependency is on the implementation of DNSSEC itself. As stated in Section 2.4, we mandate the use of secure methods for lookup and that SSHFP RRs are authenticated by trusted SIG RRs. This is especially important if SSHFP is to be used as a basis for host key rollover and/or revocation, as described above.

Since DNSSEC only protects the integrity of the host key fingerprint after it is signed by the DNS zone administrator, the fingerprint must be transferred securely from the SSH host administrator to the DNS zone administrator. This could be done manually between the administrators or automatically using secure DNS dynamic update [11] between the SSH server and the nameserver. We note that this is no different from other key enrollment situations, e.g., a client sending a certificate request to a certificate authority for signing.

5. IANA Considerations

IANA has allocated the RR type code 44 for SSHFP from the standard RR type space.

IANA has opened a new registry for the SSHFP RR type for public key algorithms. The defined types are:

0 is reserved
1 is RSA
2 is DSA

Adding new reservations requires IETF consensus [4].

IANA has opened a new registry for the SSHFP RR type for fingerprint types. The defined types are:

0 is reserved
1 is SHA-1

Adding new reservations requires IETF consensus [4].

6. Normative References

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

[2] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[4] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[5] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

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

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

7. Informational References

[8] Thayer, R., Doraswamy, N., and R. Glenn, "IP Security Document Roadmap", RFC 2411, November 1998.

[9] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[10] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[11] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

8. Acknowledgements

The authors gratefully acknowledge, in no particular order, the contributions of the following persons:

Martin Fredriksson

Olafur Gudmundsson

Edward Lewis

Bill Sommerfeld

Authors' Addresses

Jakob Schlyter
OpenSSH
812 23rd Avenue SE
Calgary, Alberta T2G 1N8
Canada

EMail: jakob@openssh.com
URI: http://www.openssh.com/

Wesley Griffin
SPARTA
7075 Samuel Morse Drive
Columbia, MD 21046
USA

EMail: wgriffin@sparta.com
URI: http://www.sparta.com/

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).