DNSSEC and SSH Key Fingerprints

Like any connection mechanism that is secured by cryptographic keys, you must have some way to verify that the key presented to you by the target host actually belongs to that host when you are using SSH. There are several ways to do this:

- **TOFU (Trust on First Use)** - trust the key you get the first time you connect. The keys are persisted in your `~/.ssh/known_hosts` file.
- **known_hosts** - trust a system-wide file of known host keys (typically `/etc/known_hosts2`).
- **out-of-band validation** - validate the key fingerprint via some other means. This is where SSHFP records signed with DNSSEC come in.

Standards

- **RFC-4255: Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints**

Operational Information

**SSH Key Hashes**
SSH generally displays the MD5 has of an SSH key. SSHFP records use a SHA-1 hash.

**Resolver Bugs**
On some OSs, the system resolver library does not properly do EDNS0 queries and SSHFP records will not be interpreted as secure even though they are.

### DNS SSHFP RR RDATA

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Fingerprint Type</th>
<th>Fingerprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 RSA</td>
<td>1 SHA-1</td>
<td>fingerprint</td>
</tr>
<tr>
<td>2 DSS</td>
<td>1 SHA-1</td>
<td>fingerprint</td>
</tr>
</tbody>
</table>

Sample Records:

```
; A fingerprint for a RSA key
host.example.edu. IN SSHFP 1 1 85b5a41a365859a4309235353a4bbc083cf9ee9a4
; A fingerprint for a DSS key
host.example.edu. IN SSHFP 2 1 b19551581d689de0ec087b529454bf892295b062
```

### Building the SSHFP RR

If the `sshfp` command available for your OS:

```
$ sshfp host.example.edu.
host.example.edu. IN SSHFP 1 1 85b5a41a365859a4309235353a4bbc083cf9ee9a4
host.example.edu. IN SSHFP 2 1 b19551581d689de0ec087b529454bf892295b062
```

If you are logged into a host and can read the public key files:
$ ssh-keygen -r $(hostname --fqdn) -f /etc/ssh/ssh_host_rsa_key.pub
host.example.edu. IN SSHFP 1 1 840ad415fb782bf704b871be57b98c5b9869b9d

$ ssh-keygen -r $(hostname --fqdn) -f /etc/ssh/ssh_host_dsa_key.pub
host.example.edu. IN SSHFP 2 1 8373cf31c903c5d58ec31bf853b3a313f1cb8e5d

Obtain the fingerprint of the key remotely with ssh-keyscan:

$ ssh-keyscan -tssh-rsa host.example.edu | cut -d ' ' -f 3 | openssl base64 -d -A | openssl sha1

$ ssh-keyscan -tssh-dss host.example.edu | cut -d ' ' -f 3 | openssl base64 -d -A | openssl sha1

OS Support

SSH Client Support

Testing

• ssh -oVerifyHostKeyDNS=yes hostname
• RES_OPTIONS=edns0 ssh -v -oVerifyHostKeyDNS=yes hostname
  • /etc/resolv.conf

# Working DNSSEC and working client.

$ ssh -oVerifyHostKeyDNS=yes -v host.example.edu
OpenSSH_5.9p1, OpenSSL 1.0.0j-fips 10 May 2012
debug1: found 2 secure fingerprints in DNS
debug1: matching host key fingerprint found in DNS
debug1: ssh_rsa_verify: signature correct
# Without -v flag, you will just get the password prompt.
user@host.example.edu's password:

# Resolver does not do DNSSEC, but host keys are in dns

$ ssh -oVerifyHostKeyDNS=yes -v host.example.edu
OpenSSH_5.9p1, OpenSSL 1.0.0j-fips 10 May 2012
debug1: found 2 insecure fingerprints in DNS
debug1: matching host key fingerprint found in DNS
The authenticity of host 'host.example.net (10.10.10.1)' can't be established.
Matching host key fingerprint found in DNS.
# You get prompted to accept the key.
Are you sure you want to continue connecting (yes/no)?
# No SSHFP record at all, with -v option
DNS lookup error: data does not exist