SSH host key verification fingerprints in the DNS

A large-scale analysis of an unknown feature and its implications.

Sebastian Neef @ TechCamp 2022

TU Berlin

2022-09-29

Who am I?

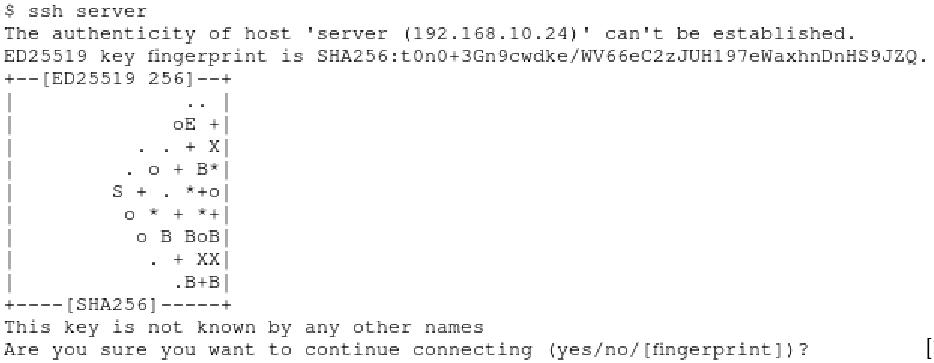
Sebastian Neef

- PhD candidate @ TU Berlin
- IT-Sec Freelancer, CTF-Player, Bughunter, etc.
- @gehaxelt

What is on our agenda?

1. What are SSH host key verification fingerprints and SSHFP records?
2. Our large-scale analysis results
3. Call to action!

SSH host key verification: What is it?



```
$ ssh server
The authenticity of host 'server (192.168.10.24)' can't be established.
ED25519 key fingerprint is SHA256:t0n0+3Gn9cwdke/WV66eC2zJUH197eWaxhnDnHS9JZQ.
+--[ED25519 256]--+
|              .. |
|             oE +|
|          . . + X|
|         . o + B*|
|        S + . *+o|
|         o * + *+|
|          o B BoB|
|           . + XX|
|             .B+B|
+----[SHA256]-----+
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

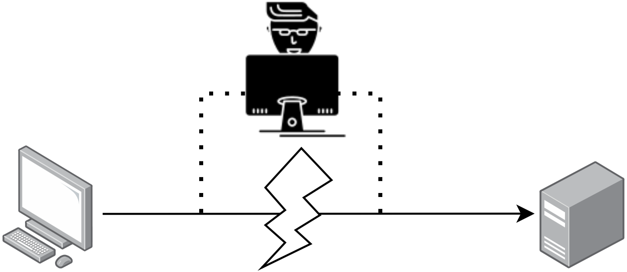
[1]

SSH host key verification: Why do we need this?

- In short: Verify that we connect to the correct server.

SSH host key verification: Why do we need this?

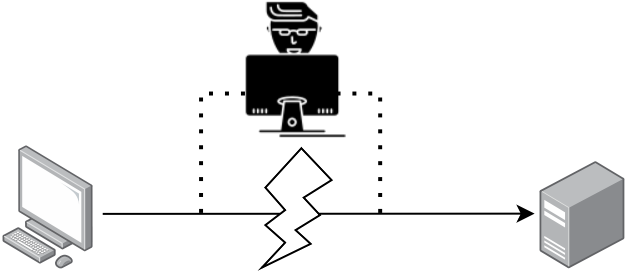
- In short: Verify that we connect to the correct server.



- If not, malice-in-the-middle attacks are possible:
  - Steal usernames and passwords from password-based logins.
  - Hijack pubkey-based logins.

SSH host key verification: Why do we need this?

- In short: Verify that we connect to the correct server.

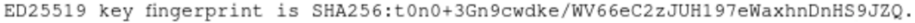


- If not, malice-in-the-middle attacks are possible:
  - Steal usernames and passwords from password-based logins.
  - Hijack pubkey-based logins.

⇒ Verifying the host key is a crucial security feature and should always be done.

SSH host key verification: How to do it?

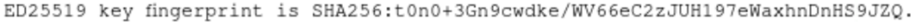
- Before typing "YES", obtain the server's host key fingerprint and verify it (RFC 4251)



```
ED25519 key fingerprint is SHA256:t0n0+3Gn9cwdke/WV66eC2zJUH197eWaxhnDnHS9JZQ.
```

SSH host key verification: How to do it?

- Before typing "YES", obtain the server's host key fingerprint and verify it (RFC 4251)



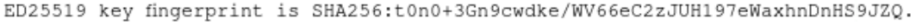
```
ED25519 key fingerprint is SHA256:t0n0+3Gn9cwdke/WV66eC2zJUH197eWaxhnDnHS9JZQ.
```

Manual process

1. Ask the admin for the fingerprints
2. Manually compare both fingerprints
3. Continue or abort connecting

SSH host key verification: How to do it?

- Before typing "YES", obtain the server's host key fingerprint and verify it (RFC 4251)



```
ED25519 key fingerprint is SHA256:t0n0+3Gn9cwdke/WV66eC2zJUH197eWaxhnDnHS9JZQ.
```

Manual process

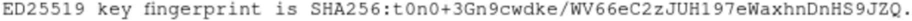
1. Ask the admin for the fingerprints
2. Manually compare both fingerprints
3. Continue or abort connecting

DNS-based process

1. Let the admin publish the fingerprints in the DNS (using DNSSEC!)
2. Let the openssh-client do the comparison
3. Continue or abort connecting

SSH host key verification: How to do it?

- Before typing "YES", obtain the server's host key fingerprint and verify it (RFC 4251)



```
ED25519 key fingerprint is SHA256:t0n0+3Gn9cwdke/WV66eC2zJUH197eWaxhnDnHS9JZQ.
```

Manual process

1. Ask the admin for the fingerprints
2. Manually compare both fingerprints
3. Continue or abort connecting

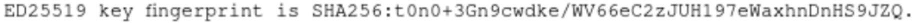
DNS-based process

1. Let the admin publish the fingerprints in the DNS (using DNSSEC!)
2. Let the openssh-client do the comparison
3. Continue or abort connecting

⇒ One method requires manual work and is error prone, the other requires a little more administrative work.

SSH host key verification: How to do it?

- Before typing "YES", obtain the server's host key fingerprint and verify it (RFC 4251)



```
ED25519 key fingerprint is SHA256:t0n0+3Gn9cwdke/WV66eC2zJUH197eWaxhnDnHS9JZQ.
```

Manual process

1. Ask the admin for the fingerprints
2. Manually compare both fingerprints
3. Continue or abort connecting

DNS-based process

1. Let the admin publish the fingerprints in the DNS (using DNSSEC!)
2. Let the openssh-client do the comparison
3. Continue or abort connecting

⇒ One method requires manual work and is error prone, the other requires a little more administrative work.

But be honest: Who does this?

SSHFP DNS records: Theory

- RFCs 4255, 6594, 7479, 8709 define how to store host key fingerprints in the DNS.
- Format: SSHFP <KEY-ALGO> <HASH-TYPE> <FINGERPRINT>

SSHFP DNS records: Theory

- RFCs 4255, 6594, 7479, 8709 define how to store host key fingerprints in the DNS.
- Format: SSHFP <KEY-ALGO> <HASH-TYPE> <FINGERPRINT>

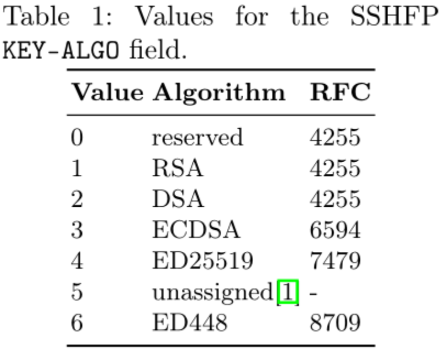


Table 1: Values for the SSHFP `KEY-ALGO` field.

Value	Algorithm	RFC
0	reserved	4255
1	RSA	4255
2	DSA	4255
3	ECDSA	6594
4	ED25519	7479
5	unassigned[1]	-
6	ED448	8709

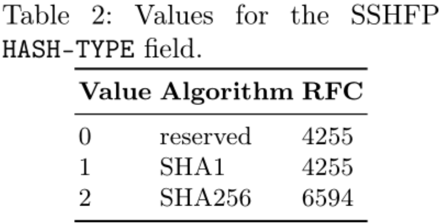
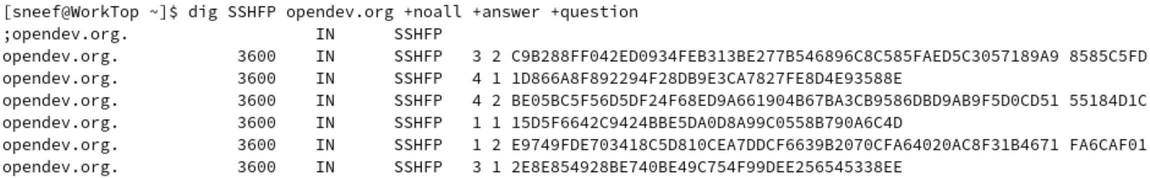


Table 2: Values for the SSHFP `HASH-TYPE` field.

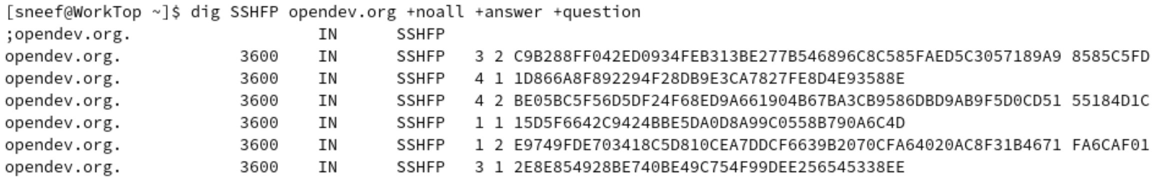
Value	Algorithm	RFC
0	reserved	4255
1	SHA1	4255
2	SHA256	6594

SSHFP DNS records: Theory

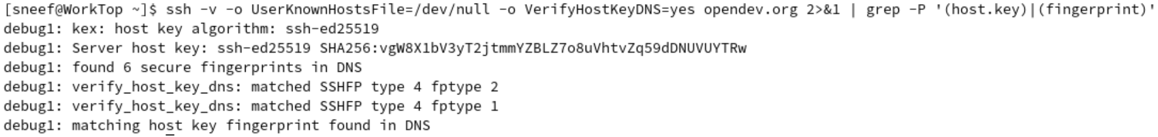


```
[sneef@WorkTop ~]$ dig SSHFP opendev.org +noall +answer +question
;opendev.org.                   IN      SSHFP
opendev.org.            3600    IN      SSHFP   3 2 C9B288FF042ED0934FEB313BE277B546896C8C585FAED5C3057189A9 8585C5FD
opendev.org.            3600    IN      SSHFP   4 1 1D866A8F892294F28DB9E3CA7827FE8D4E93588E
opendev.org.            3600    IN      SSHFP   4 2 BE05BC5F56D5DF24F68ED9A661904B67BA3CB9586DBD9AB9F5D0CD51 55184D1C
opendev.org.            3600    IN      SSHFP   1 1 15D5F6642C9424BBE5DA0D8A99C0558B790A6C4D
opendev.org.            3600    IN      SSHFP   1 2 E9749FDE703418C5D810CEA7DDCF6639B2070CFA64020AC8F31B4671 FA6CAF01
opendev.org.            3600    IN      SSHFP   3 1 2E8E854928BE740BE49C754F99DEE256545338EE
```

SSHFP DNS records: Theory



```
[sneef@WorkTop ~]$ dig SSHFP opendev.org +noall +answer +question
;opendev.org.                   IN      SSHFP
opendev.org.            3600    IN      SSHFP   3 2 C9B288FF042ED0934FEB313BE277B546896C8C585FAED5C3057189A9 8585C5FD
opendev.org.            3600    IN      SSHFP   4 1 1D866A8F892294F28DB9E3CA7827FE8D4E93588E
opendev.org.            3600    IN      SSHFP   4 2 BE05BC5F56D5DF24F68ED9A661904B67BA3CB9586DBD9AB9F5D0CD51 55184D1C
opendev.org.            3600    IN      SSHFP   1 1 15D5F6642C9424BBE5DA0D8A99C0558B790A6C4D
opendev.org.            3600    IN      SSHFP   1 2 E9749FDE703418C5D810CEA7DDCF6639B2070CFA64020AC8F31B4671 FA6CAF01
opendev.org.            3600    IN      SSHFP   3 1 2E8E854928BE740BE49C754F99DEE256545338EE
```



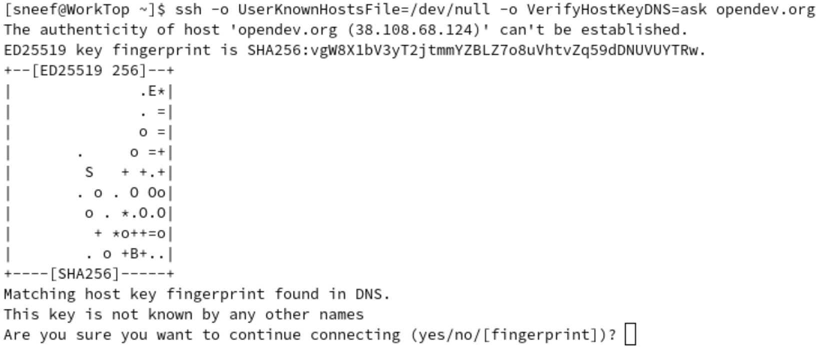
```
[sneef@WorkTop ~]$ ssh -v -o UserKnownHostsFile=/dev/null -o VerifyHostKeyDNS=yes opendev.org 2>&1 | grep -P '(host.key)|(fingerprint)'
debug1: kex: host key algorithm: ssh-ed25519
debug1: Server host key: ssh-ed25519 SHA256:vgW8X1bV3yT2jtmmYZBLZ7o8uVhtvZq59dDNUVUYTRw
debug1: found 6 secure fingerprints in DNS
debug1: verify_host_key_dns: matched SSHFP type 4 fptype 2
debug1: verify_host_key_dns: matched SSHFP type 4 fptype 1
debug1: matching host key fingerprint found in DNS
```

SSHFP DNS records: Live Demo

Live-Demo

SSHFP DNS records: Live Demo

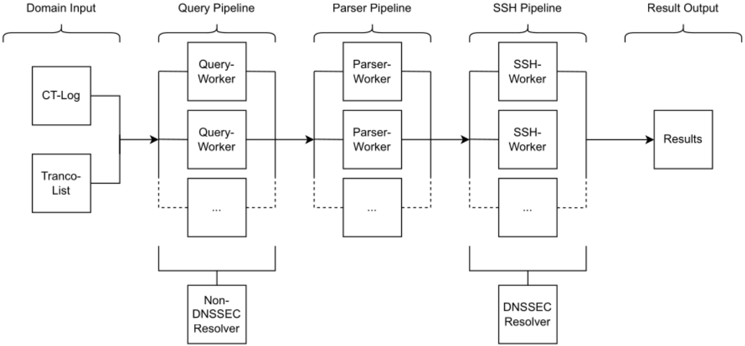
Live-Demo



```
[sneef@WorkTop ~]$ ssh -o UserKnownHostsFile=/dev/null -o VerifyHostKeyDNS=ask opendev.org
The authenticity of host 'opendev.org (38.108.68.124)' can't be established.
ED25519 key fingerprint is SHA256:vgW8X1bV3yT2jtmmYZBLZ7o8uVhtvZq59dDNUVUYTRw.
+--[ED25519 256]--+
|              .E*|
|              . =|
|              o =|
|       .     o =+|
|        S   + +.+|
|       . o . O Oo|
|        o . *.O.O|
|         + *o++=o|
|        . o +B+..|
+----[SHA256]-----+
Matching host key fingerprint found in DNS.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Large-scale analysis: Methodology

1. Query the Tranco 1M list **and** $\geq$ 500M certificate transparency logs for SSHFP records
2. Query a domain's A records to find possible hosts
3. Obtain server-side host key fingerprints using SSH
4. Compare DNS-hosted and server-side host key fingerprints
5. Check whether the records are DNSSEC-secured



Large-scale analysis: Results (1)

Tranco 1M

- 1M domains scanned
- 105 domains use SSHFP (0.011%)
- 75 servers run SSH
- 66 with $\geq$ 1 matching fingerprint
- 28 use DNSSEC

Large-scale analysis: Results (1)

Tranco 1M

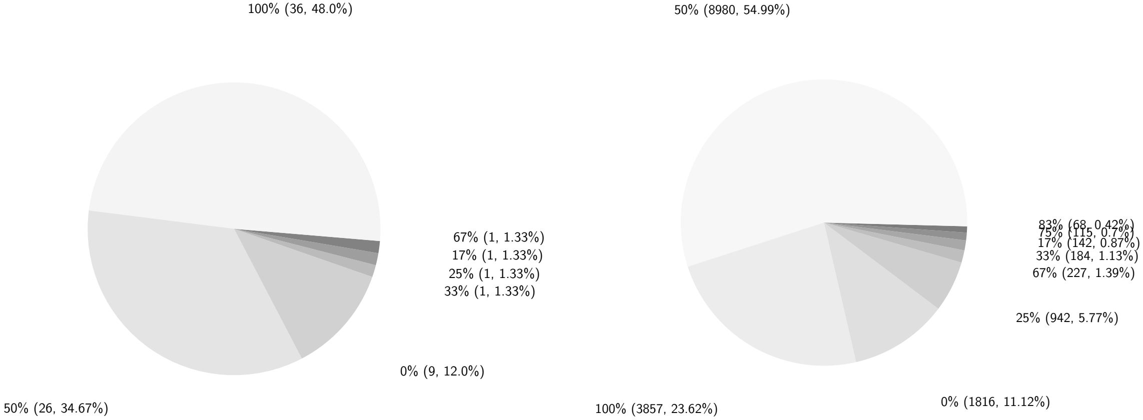
- 1M domains scanned
- 105 domains use SSHFP (0.011%)
- 75 servers run SSH
- 66 with $\geq$ 1 matching fingerprint
- 28 use DNSSEC

Certificate Transparency Logs

- 515M domains scanned (136M unique)
- 17,672 SSHFP sets (11,524 unique domains)
- 16,331 servers run SSH
- 14,515 with $\geq$ 1 matching fingerprint
- 3,896 unique domains use DNSSEC

Large-scale analysis: Results (2)

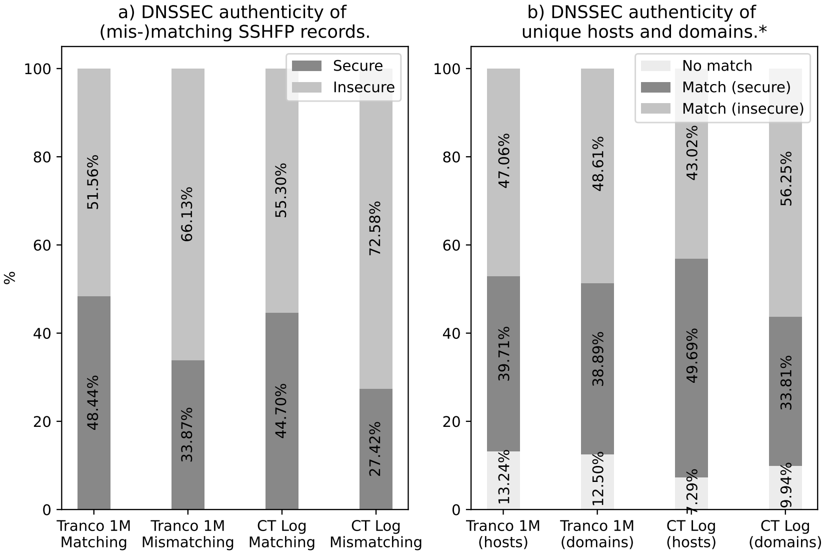
- Less than 50% domains have 100% DNS-vs-server matching ratios



(a) Tranco 1M

(b) Certificate Transparency Log

Large-scale analysis: Results (3)



Call to action!

⇒ Security benefits wait for you! DNS-based host key verification is not hard :-)

Call to action!

⇒ Security benefits wait for you! DNS-based host key verification is not hard :-)

- If you use SSH, consider using SSHFP DNS records.

Call to action!

⇒ Security benefits wait for you! DNS-based host key verification is not hard :-)

- If you use SSH, consider using SSHFP DNS records.
- If you use SSHFP records, do **not** forget to use DNSSEC (or other secure channels)!

Call to action!

⇒ Security benefits wait for you! DNS-based host key verification is not hard :-)

- If you use SSH, consider using SSHFP DNS records.
- If you use SSHFP records, do **not** forget to use DNSSEC (or other secure channels)!
- If you got this far, tell openssh to use the records:

```
$> ssh -o VerifyHostKeyDNS=yes <...>
```

Call to action!

⇒ Security benefits wait for you! DNS-based host key verification is not hard :-)

- If you use SSH, consider using SSHFP DNS records.
- If you use SSHFP records, do **not** forget to use DNSSEC (or other secure channels)!
- If you got this far, tell openssh to use the records:

```
$> ssh -o VerifyHostKeyDNS=yes <...>
```

* If you want to know more, read the paper[1] ;-)

Questions & Answers

Thanks for listening!

Questions?

Feel free to reach out: neef@tu-berlin.de

References

1. SSHFP DNS paper - TBD Springer LNCS or `https://arxiv.org/abs/2208.08846`
2. Repo with code & data - `https://github.com/gehaxelt/sshfp-dns-measurement`
3. Tranco 1M - `https://tranco-list.eu/`