Oh SSH-it, what's my fingerprint? A Large-Scale Analysis of SSH Host Key Fingerprint Verification Records in the DNS

Sebastian Neef @ CANS 2022

Chair for Security in Telecommunications
Technische Universität Berlin, Germany
neef@tu-berlin.de

2022-11-14

Motivation

- Secure Shell (SSH) protocol is widely used to connect to remote systems
- Anecdotal evidence suggested that users do not properly verify host key fingerprints [1]
- An incomplete or incorrect verification embodies a security risk (i.e. MITM)

Motivation

- Secure Shell (SSH) protocol is widely used to connect to remote systems
- Anecdotal evidence suggested that users do not properly verify host key fingerprints [1]
- An incomplete or incorrect verification embodies a security risk (i.e. MITM)

- SSHFP records is one solution standardized with RFC 4255 in 2006
- Little research: Only few records observed by Gasser et al. [NOMS 2014] [2]

⇒ Measure its adoption almost a decade later.

Contributions

- Answers to the following questions:

RQ1 How common are DNS-based host key verification records (SSHFP)?

RQ2 Do the SSFHP records match their service counterpart?

RQ3 Are these records properly secured using DNSSEC?

- Artifacts from our large-scale analysis:
  - Python SSHFP-library
  - All analysis scripts & (intermediate) data sets

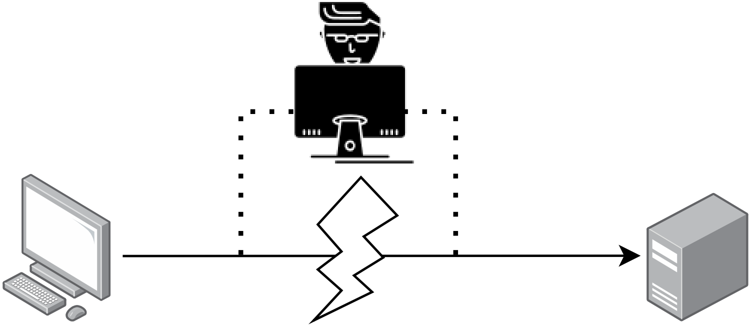
What is SSH host key verification?

- SSH uses public-key cryptography to establish the authenticity of a server
- TOFU requires the user to verify the server's host key fingerprint

```
$ ssh server
The authenticity of host 'server (192.168.10.24)' can't be established.
ECDSA key fingerprint is SHA256:jq3V6ES34fNDKdn5L1sbmhoyJ5MN9afd9wIS1Upa1dc.
+---[ECDSA 256]---+
|             o...|
|            + . E|
|           * .  .|
|          + o    |
|        S.o+ + . |
|       = =Bo* + .|
|      . **=B + o+|
|       =  *oo *.=|
|      . .  .o+.*o|
+----[SHA256]-----+
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Why is this important?

- In short: Verify that a user connects to the correct server.

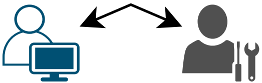


- If not, malice-in-the-middle attacks might obtain credentials or unauthorized access

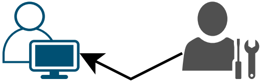
⇒ Host key verification is a crucial security feature that should always be done

How to perform host key verification?

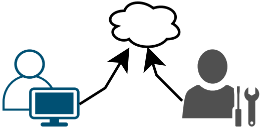
- Manually
  - A user asks the administrator for the fingerprints
  - The user manually verifies the fingerprint



- Certificate Authority
  - An administrator deploys a root-CA to the user's device(s)
  - The SSH client validates the host key's signature and verifies the fingerprint



- SSHFP DNS records
  - An administrator deploys the fingerprints as SSHFP DNS records using DNSSEC
  - The SSH client queries these records and verifies the fingerprint



RFC 4255: Using DNS to Securely Publish SSH Key Fingerprints

- RFCs 4255, 6594, 7479, 8709 define and extend SSHFP records
- Format: SSHFP <KEY-ALGO> <HASH-TYPE> <FINGERPRINT>

Table: 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	-
6	ED448	8709

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

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

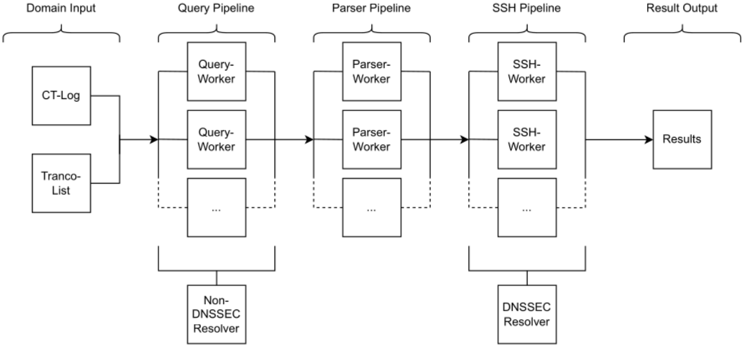
Real-world example

```
[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
```

→ RSA (1), ECDSA (3), ED25519 (4) keys with SHA1 (1) and SHA256 (2) hashes

```
[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
```

Data collection



1. Query a domain for SSHFP records and validate their format
2. Query A records and collect server-side host key fingerprints using SSH
3. Resend SSHFP query through DNSSEC resolver
4. Match SSHFP records with server-side fingerprints

Data analysis

- Empirically collected data from two domain sets:
  - Tranco 1M (ID: G8KK)
  - $>$ 515M domains observed on the certificate transparency log over 26 days
- Quantitative analysis to answer our RQs
- Focus on reproducibility: All code and (intermediate) data sets available [3,4]

Tranco 1M

- 105 domains (0.0105%) with 465 SSHFP records in total

- 75 SSH servers (72 domains) provide 380 server-side fingerprints
- 66 hosts with ≥ 1 matching fingerprint (256 fingerprints)

- 28 domains are DNSSEC secured

Certificate Transparency Log

- Scanned 515M domains over 26 days (136.5M unique; 45M registrable) $\rightarrow$ repetitions
- 23,823 unique SSHFP records from 74,740 record sets (5,961 unique) mapping to 17,672 unique domains (7,007 registrable)

- 16,331 SSH servers (11,524 unique domains) provide 72,512 server-side fingerprints
- 14,515 hosts with ≥ 1 matching fingerprint (10,378 unique domains)

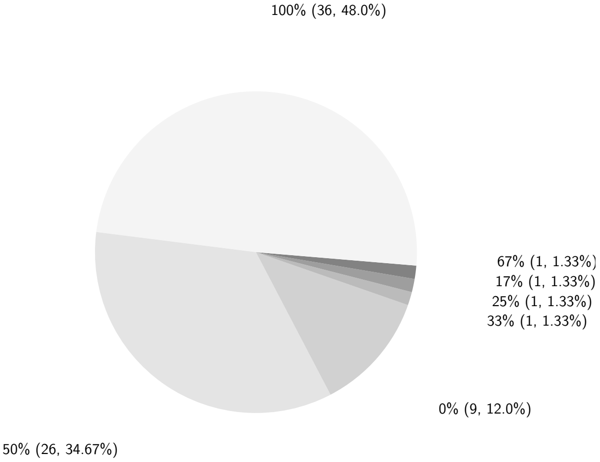
- 3,896 unique domains are DNSSEC secured

---

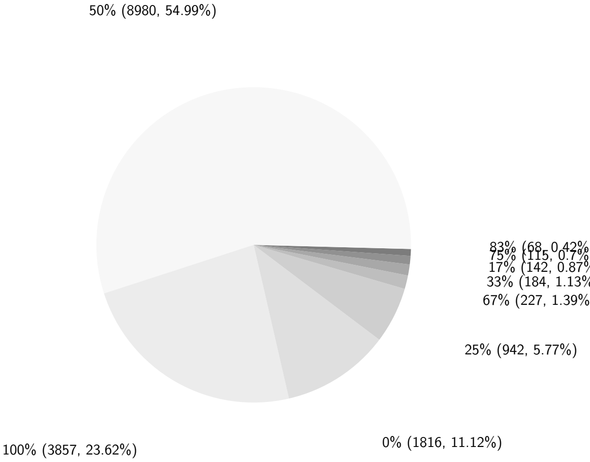
registrable domain: `example.com`; unique domain: `www.example.com`,`mail.example.com`,...

SSHFP vs. server-side host key fingerprint matching

- < 50% of hosts fulfill a 100% matching ratio required by newer OpenSSH versions [5]

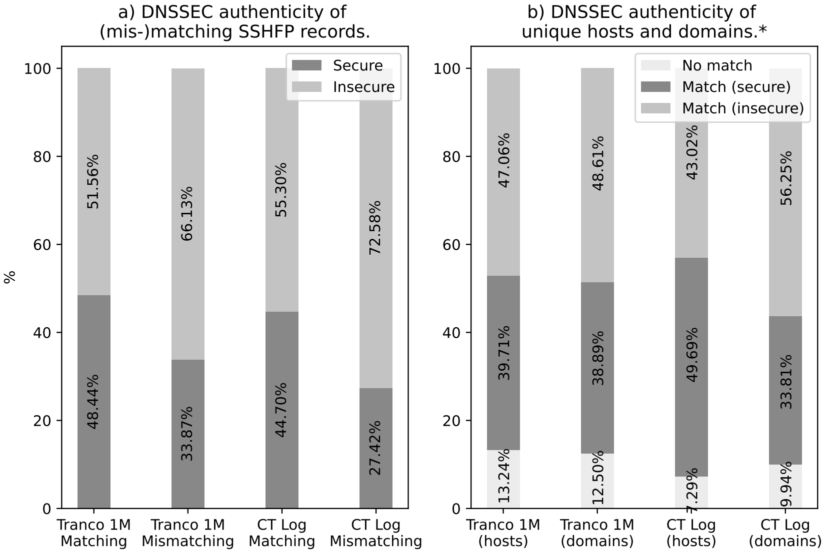


(a) Tranco 1M



(b) Certificate Transparency Log

SSHFP and DNSSEC



* Limited to the first measurement per domain.

Discussion

- Prevalence
  - Overall low (~1 in 10,000 domains)
    - Not enabled by default in OpenSSH
    - Dependency on 'secure DNS' (i.e. DNSSEC)
  - Our work and Gasser et al. can only provide a lower bound
- Security & Privacy
  - Lower matching rate than Gasser et al. (88% vs. 94%)
    - Improper deployments: Mismatching SSHFP records or wrong KEY-/HASH-/FP values
  - Increase in DNSSEC adoption (44% vs. 31.8%), but many records still insecure
  - Modern key algorithms (EC*, SHA-256) are still behind established ones (RSA, DSA, SHA-1)
  - Duplicate fingerprints disclose links between domains or potential key-reuse

Limitations & Future Work

- Limitations
  - No insight into *private* DNS servers, only public ones
  - 5% DNS resolving errors (NXDOMAIN, timeouts, ...)
  - Short disconnects from the certificate transparency log provider ($\leq$ 3% of the total time)
- Future Work
  - Find alternative and better domain sources
  - Longitudinal study to monitor changes in adoption and deployment
  - Studying causes of the low SSHFP adoption: unawareness? technical?

Conclusion

- In this work, we performed a large-scale analysis of SSHFP records
- (Still) no widespread adoption (~1 in 10,000 domains), although its standardization was $\geq$ 15 years ago
- Misconfiguration eliminates most benefits:
  - DNS and server-side fingerprints differ $\rightarrow$ broken verification for $\geq$ 50% of hosts
  - Lack of DNSSEC violates the standard $\rightarrow$ reduced security for ~50% of domains
- If used correctly, SSHFP records can mitigate many of SSH's TOFU risks!

Thank you for your time and attention!

Let's talk!

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

References

1. Gutmann, P.: Do Users Verify SSH Keys? p. 2
2. Gasser, O., Holz, R., Carle, G.: A deeper understanding of SSH: Results from Internet-wide scans. In: 2014 IEEE Network Operations and Management Symposium (NOMS). pp. 1–9 (May 2014). https://doi.org/10.1109/NOMS.2014.6838249
3. https://github.com/gehaxelt/sshfp-dns-measurement
4. https://zenodo.org/record/6993096
5. https://marc.info/?l=openssh-unix-dev&m=164700394009668&w=2

KEY-Algo and Hash-Type values

Table: Distribution of `KEY-ALGO` and `HASH-TYPE` values for the Tranco 1M list

Data From	**Key Algorithm**						**Hash Type**	
	RESERVED	RSA	DSA	ECDSA	ED25519	ED448	SHA1	SHA256
DNS	0	131	79	109	103	0	245	177
SSH	0	138	22	106	114	0	190	190
– Matching	0	93	10	74	79	0	151	105
– Mismatching	0	45	12	32	35	0	39	85

Table: Distribution of `KEY-ALGO` and `HASH-TYPE` values for the Certificate Transparency Logs

Data From	**Algorithm**						**Hash Type**	
	RESERVED	RSA	DSA	ECDSA	ED25519	ED448	SHA1	SHA256
DNS	1	7,536	2,367	6,726	7,191	2	9,054	14,769
SSH	0	26,974	5,680	19,562	20,296	0	36,256	36,256
– Matching	0	15,190	1,528	11,972	12,211	0	21,871	19,030
– Mismatching	0	11,784	4,152	7,590	8,085	0	14,385	17,226

Why is DNSSEC important?

- RFC 1035 standardizes the DNS protocol in 1987, but without security
- DNSSEC introduced with RFC 2535 in 1999 and superseded by RFC 4033ff in 2005:

  "The DNS security extensions provide origin authentication and integrity protection for DNS data, as well as a means of public key distribution. These extensions do not provide confidentiality."

⇒ Mitigate fingerprint manipulation on the DNS-level

Gasser et al. vs. our work

- Gasser et al. used PTR records:
  1. Scan IPv4 space for hostnames (PTR)
  2. Forward-resolve (A) for validation
  3. Query SSHFP records
  4. Compare the fingerprints
- We used a longitudinal approach with first SSHFP followed by A queries
  1. Query SSHFP records
  2. Query A records
  3. Compare the fingerprints
- For our over 10,000 SSHFP domains, only ~1,900 had reverse lookup information correctly set up

→ Both approaches have their (dis-)advantages