Berner
Fachhochschule

Sicherer Umgang mit dem SSH-Serverdienst von OpenSSH

Semesterarbeit CAS IT Security Management FS23

27. September 2023

Mauro Guadagnini

Zielsetzung

- SSH-Serverdienst von OpenSSH verstehen und absichern
- Wie sicher ist die Standardkonfiguration?
- Anwendungsfälle
  - Kommandozeilenzugriff
  - Dateiübertragungen
  - Jumphost
- Agent Forwarding

OpenSSH Software

OpenSSH-Server `sshd` [1]

- Konfigurationsdatei `/etc/ssh/sshd_config`
- Debugging-Parameter `-d`, `-dd` oder `-ddd`
- Test-Modus `-t`
- Erweiterter Test-Modus mit Konfigurationsausgabe `-T`

Ausgabe von `sshd`-Testmodus mit `-t` bei Schreibfehler in `PubkeyAuthentication`

```
/etc/ssh/sshd_config: line 50: Bad configuration option:
PubkexAuthentication
/etc/ssh/sshd_config: terminating, 1 bad configuration options
```

Wie konfigurieren?

Minimalstandard

Empfehlungen / „Best Practices“

Bundesbehörden

- SiO01 Version 5.0 *IT-Grundschutz in der Bundesverwaltung*, Feb. 2022 [2]
- BSI TR-02102-1 *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*, Jan. 2023 [3]
- NIST SP 800-175B Rev. 1 *Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms*, März 2020 [4]

Weitere Empfehlungen

- Center for Internet Security (CIS) Benchmarks [5] für u.a. Debian 11 [6] und RHEL 9 [7] sowie Distributions-unabhängige Linux-OS-Empfehlungen [8]
- Bücher *SSH, The Secure Shell: The Definitive Guide, 2nd Edition* (2005) [9] und *SSH Mastery - Second Edition* (2018) [10]

Vergleich Minimalstandard mit Standardkonfiguration [12]

- Algorithmen-Wahl orientiert sich am aktuellen Stand der Technik [11]
  - Bei Vorgaben überschreiben und manuell pflegen
- Einfache Passwortauthentisierung, kein 2FA
- Keine Benutzereinschränkung (ausser `root`)
- Forwarding von Agent, TCP-Ports und Sockets erlaubt
  Aufruf geöffneter Port über Loopback-Adresse
- Allgemein gültige Optionen

Laborumgebung

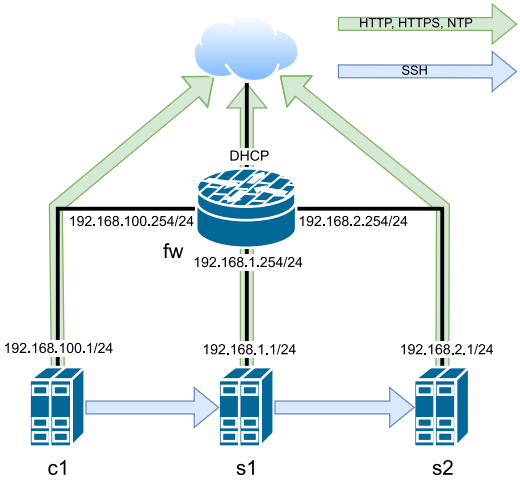

Abbildung: Laborumgebung mit IP-Adressen und zugelassenen Datenflüssen

- SSH-Client mit User-Keys auf Client `c1`
- Haupt-Konfiguration auf Server `s1`
- Server `s2` nur via Server `s1` erreichbar

Grundkonfiguration auf Server

Anpassungen an Standardkonfiguration [12]

- `LoginGraceTime` auf 1 Minute
- `RequiredRSASize` von 1024 auf 3072 Bits
- Algorithmen-Wahl gemäss ermitteltem Minimalstandard
- Deaktivierung Benutzer-Skripts, Kompression, Forwarding
- **Einschränkung des Logins** auf Benutzer der Gruppe `sshaccess`
- Forcierung der **kombinierten Authentisierung** mittels Public-Key und Passwort
- Forcierung der Eingabe eines PIN-Codes bei Verwendung eines FIDO Authenticators
- Nicht benötigte Authentisierungsmethoden deaktiviert
- Allozieren eines Pseudo-Terminals unterbunden und den Befehl `exit` forciert
- Ausgabe Login-Zeitstempel und „Message of the day“ deaktiviert

Kommandozeilenzugriff

Kommandozeilenzugriff auf Server `s1`

```
# User cmd ------------------------------------------------------------
Match User cmd
    PermitTTY                   yes
    ForceCommand                none
```

Benutzer mit Kommandozeilenzugriff hat viel Macht und kann…

- … auch ohne SFTP Dateien kopieren
  `ssh cmd@192.168.1.1 cat /tmp/test.tar.gz > test.tar.gz`
- … je nach Berechtigungen Anpassungen vornehmen

Einschränkung mittels `ForceCommand` [12]

Dateiübertragungen

Dateiübertragungen auf Server `s1`

```
# User file ----------------------------------------------------------
Match User file
    ForceCommand                internal-sftp
    ChrootDirectory             /data/sftp
```

- Mit `ChrootDirectory` wird ein Pfad als „root“-Verzeichnis hinterlegt, der Benutzer kann nur innerhalb dieses Pfades operieren [12]
- Der Benutzer erhält so keinen Kommandozeilenzugriff
  Antwort von Server: `This service allows sftp connections only.`

Jumphost

Jumphost auf Server `s1`

```
# User jump ------------------------------------------------------------
Match User jump
    DisableForwarding          no
    AllowTcpForwarding         yes
    PermitOpen                 192.168.2.1:22 # Server s2
    MaxSessions                0
```

- Ein Jumphost-User erhält so keinen Kommandozeilenzugriff auf dem Server `s1`
- Zugriff auf `s2` hängt von dessen Konfiguration ab

Agent Forwarding

Agent-Forwarding und Kommandozeilenzugriff auf Server `s1`

```
# User agent ----------------------------------------------------------
Match User agent
    AllowAgentForwarding      yes
    PermitTTY                 yes
    ForceCommand              none
```

Einschränkung der Key-Nutzung beim Agent

```
ssh-add -h 'agent@192.168.1.1' -h '192.168.1.1>cmd@192.168.2.1' \
~/.ssh/id_ed25519
```

- Verwendung von Jumphosts als sicherere Alternative gegenüber Agent-Forwarding empfohlen (siehe `ssh`-Manpage [13])

Weitere Implementationen

- Zertifikatsauthentisierung
  - CA-Public-Schlüssel hinterlegen
  - Host- und Client-Keys vertrauen
  - Einsatz „Principals“, z.B. nur Zertifikate mit Principal `db` zulassen
  - Forcierung Zertifikatsauthentisierung
  - Weitere Einschränkungen möglich: Ablaufdatum, Revocation Listen, etc.
- SSHFP-DNS-Records mit Fingerprint des Server-Public-Keys
  - Wird nur mit aktiver Client-Option und FQDN geprüft
  - Alert nur bei Mismatch

Weitere Implementationen

- FIDO2 Authentisierung mit YubiKey
  - Schlüssel auf YubiKey erstellen
  - Schlüssel-Referenzdateien von YubiKey auslesen (`ssh-keygen -K`)
    Verwenden wie „klassische“ Keys

Verbindungsaufbau mit FIDO2 und YubiKey

```
c1$ ssh cmd@192.168.1.1
Enter passphrase for key '/home/user/.ssh/id_ed25519_sk':
Confirm user presence for key ED25519-SK SHA256:3PpQ...SrgI
Enter PIN for ED25519-SK key /home/user/.ssh/id_ed25519_sk:
Confirm user presence for key ED25519-SK SHA256:3PpQ...SrgI
User presence confirmed
cmd@192.168.1.1's password:
s1$
```


Fazit

- SSH-Serverdienst von OpenSSH nach Auseinandersetzung greifbarer
- Software nahe am Stand der Technik und schnell gepatcht
- Standardkonfiguration Kompromiss zwischen Sicherheit und Komfort
  ➔ Schnellstmöglich absichern
- Algorithmen-Wahl vertrauen oder gemäss Vorgaben manuell pflegen
- Forwarding kontrolliert einsetzen (z.B. als Jumphost)
- Diverse Absicherungsmöglichkeiten (Zertifikate, FIDO-PIN, etc.)
- Infrastruktur und Dokumentation aktuell halten
- Arbeitsflüsse definieren und regelmässig prüfen

Ausblick

Punkte zur zukünftigen Vertiefung

- Logging und Auswertung Logs
- SSH-Client-Konfiguration
- Automatisierung
- Weitere Authentisierungsmethoden
- Zertifikate mit Revocation Listen und weiteren Features
- Forwarding Unix Sockets
- Tunneling

- Der Umgang mit dem SSH-Serverdienst von OpenSSH ist nun sicherer
- Fragen?

Bonus: Minimalstandard

Kombination Empfehlungen

- Sym. Verschlüsselung [14][3][15][4]
  - **AES** (AES-CBC oder AES-CTR) mit **128 Bit** Schlüssellänge
  - AEAD Betriebsmodi (**AES-GCM** oder **AES-CCM**)
- Asym. Verschlüsselung: **RSA** mit **3072 Bit** Schlüssellänge [14][3][15][4]
- Hashfunktionen: **SHA2 und SHA3** mit **256 Bit** [14][3][15][4]
- Datenauthentifizierung: **HMAC mit SHA2 und SHA3** [14][3][15][4][16]
- Digitale Signaturen
  - **RSA** mit **3072 Bit** Schlüssellänge / **ECDSA** und **EdDSA** [1] [19][14][3][15][4]
- Schlüsselaustausch
  - **DHKE** mit **3072 Bit** Gruppengrösse / **ECDH (Elliptic Curve Diffie-Hellman)** [1] [19][14][3][15][4]

[1] Elliptische Kurven aus Menge von „Safe Curves“ [17], Brainpool-Kurven [18] oder NIST-Kurven mit Bitlänge von min. 255 [19][14][3][15][4]

Bonus: Minimalstandard

Kombination Empfehlungen

- **Authentisierung**
  - 2-Faktoren-Authentisierung (2FA) verwenden [2][3][20][21]
  - Filtern Benutzer und Gruppen inkl. `root` [22][9][10]
  - Host-basierte Authentisierung deaktivieren [9][7]
  - Nicht verwendete Authentisierungsmethoden deaktivieren [23]
  - Reine Password-Authentisierung deaktivieren [22][23][9][10]
  - Private-Keys ohne Passwörter nicht verwenden, für automatisierte Abläufe Authentisierungs-Agenten verwenden [9][10]
  - Public-Key-Authentisierung verwenden [22][9][21]
  - Zertifikat-Authentisierung verwenden [16][10]
- **Forwarding / Weiterleitungen** jeglicher Art deaktivieren, sofern nicht benötigt [16][23][7]
- **Spezifische Konfiguration** mittels `Match` einschränken [23][10]

Literatur I

[1] *sshd — OpenSSH daemon*. 10. Feb. 2023. URL: https://man.openbsd.org/sshd.8 (besucht am 15. 07. 2023).

[2] Informatiksicherheit Bund SEC. *IT-Grundschutz in der Bundesverwaltung*. Si001 – IT-Grundschutz in der Bundesverwaltung - Version 5.0. Feb. 2022. URL: https://www.ncsc.admin.ch/dam/ncsc/de/dokumente/dokumentation/vorgaben/sicherheit/si001/Si001-IT-Grundschutz_V5-0-d.pdf.download.pdf/Si001-IT-Grundschutz_V5-0-d.pdf.

Literatur II

[3] Bundesamt für Sicherheit in der Informationstechnik. *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*. BSI - Technische Richtlinie, Kryptographische Verfahren: Empfehlungen und Schlüssellängen. Jan. 2023. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf.

[4] Elaine Barker. *Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms*. NIST SP 800-175B Rev. 1. März 2020. DOI: 10.6028/NIST.SP.800-175Br1. URL: https://csrc.nist.gov/Pubs/sp/800/175/b/r1/Final.

[5] *CIS Benchmarks*. URL: https://learn.cisecurity.org/benchmarks (besucht am 15. 08. 2023).

Literatur III

[6] Center for Internet Security Inc. (CIS). *CIS Debian Linux 11 Benchmark*. Version 1.0.0. 22. Sep. 2022.

[7] Center for Internet Security Inc. (CIS). *CIS Red Hat Enterprise Linux 9 Benchmark*. Version 1.0.0. 28. Nov. 2022.

[8] Center for Internet Security Inc. (CIS). *CIS Distribution Independent Linux*. Version 2.0.0. 16. Juli 2019.

[9] Daniel J. Barrett, Richard E. Silverman und Robert G. Byrnes. *SSH, The Secure Shell: The Definitive Guide, 2nd Edition*. Mai 2005.

[10] Michael W. Lucas. *SSH Mastery - Second Edition. OpenSSH, PuTTY, Tunnels and Keys*. 6. Feb. 2018.

Literatur IV

[11] *OpenSSH Release Notes*. URL: https://www.openssh.com/releasenotes.html (besucht am 16. 09. 2023).

[12] *sshd_config — OpenSSH daemon configuration file*. 3. März 2023. URL: https://man.openbsd.org/sshd_config (besucht am 15. 07. 2023).

[13] *ssh — OpenSSH remote login client*. 21. Juni 2023. URL: https://man.openbsd.org/ssh.1 (besucht am 15. 07. 2023).

[14] FUB ZEO KRYPT. *Empfehlungen zu kryptografischen Verfahren für den Grundschutz*. Jan. 2023.

Literatur V

[15] Elaine Barker und Allen Roginsky. *Transitioning the Use of Cryptographic Algorithms and Key Lengths*. NIST SP 800-131A Rev. 2. März 2019. DOI: 10.6028/NIST.SP.800-131Ar2. URL: https://csrc.nist.gov/pubs/sp/800/131/a/r2/final.

[16] *sshd_config - How to Configure the OpenSSH Server?* URL: https://www.ssh.com/academy/ssh/sshd_config (besucht am 23. 07. 2023).

[17] *SafeCurves: choosing safe curves for elliptic-curve cryptography*. URL: https://safecurves.cr.yp.to (besucht am 22. 07. 2023).

[18] Johannes Merkle und Manfred Lochter. *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*. RFC 5639. März 2010. DOI: 10.17487/RFC5639. URL: https://www.rfc-editor.org/info/rfc5639.

Literatur VI

[19] Lily Chen u. a. *Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters*. NIST SP 800-186. Feb. 2023. DOI: 10.6028/NIST.SP.800-186. URL: https://csrc.nist.gov/Pubs/sp/800/186/Final.

[20] Joint Task Force. *Assessing Security and Privacy Controls in Information Systems and Organizations*. NIST SP 800-53A Rev. 5. Jan. 2022. DOI: 10.6028/NIST.SP.800-53Ar5. URL: https://csrc.nist.gov/pubs/sp/800/53/a/r5/final.

[21] Michael Kofler u. a. *Hacking & Security, 3., aktualisierte und erweiterte Auflage. Das umfassende Handbuch*. Dez. 2022.

[22] *Eight ways to protect SSH access on your system*. 29. Okt. 2020. URL: https://www.redhat.com/sysadmin/eight-ways-secure-ssh (besucht am 23. 07. 2023).

Literatur VII

[23] *How To Harden OpenSSH on Ubuntu 20.04*. 8. Nov. 2021. URL: https://www.digitalocean.com/community/tutorials/how-to-harden-openssh-on-ubuntu-20-04 (besucht am 23. 07. 2023).