SSH - Secure Shell
SFTP - SSH File Transfer Protocol

Erik Borsos

Inhaltsverzeichnis

1 Was ist SSH?

SSH, auch bekannt als **Secure Shell**, ist ein Netzwerkprotokoll auf der **7. Schicht des OSI-Modells** mit dem TCP Port **22**. Es soll **telnet ersetzen**. Secure Shell bietet eine starke Authentifizierung mit öffentlichen Schlüsseln und **eine sichere Verschlüsselung**. SSH verwendet das **Client-Server-Modell** und verbindet einen SSH-Client mit einem SSH-Server. Daher kann SSH für eine **Remote-Administration** verwendet werden. Dies ist auch die häufigste Anwendung.[1]

2 Geschichte von SSH

SSH wurde **1995** von **Tatu Ylönen** entwickelt. Er war ein Forscher an der **Universität Helsinki** und entwickelte SSH als Antwort auf eine **Password-Sniffing-Attacke**. Im **Juli 1995** wurde SSH veröffentlicht und erreichte bis Dezember mehr als 2000 Benutzer. Daraufhin gründete er eine Firma, die SSH Communications Security, Inc, um an Secure Shell weiterzuarbeiten. **1996** wurde **SSH-2** entwickelt, nachdem Schwachstellen in **SSH-1** entdeckt wurden. Die beiden Versionen sind **nicht miteinander kompatibel**. **OpenSSH** ist eine Open-Source-Version von SSH und wurde 1999 veröffentlicht. **2006** wurde diese Protokoll (Version 2) von der IETF als Internetstandard **RFC 4251** vorgeschlagen.[2]

Abbildung 1: Tatu Ylönen

3 SSH vs telnet

	SSH	**telnet**
Sicherheit	hoch gesichert	wenig gesichert
Erstveröffentlichung	1995	1969
Port	TCP Port 22	TCP Port 23
Datenformat	verschlüsselt mithilfe eines Secure-Channels	Unverschlüsselt
Authentifizierung	öffentliche Schlüssel	Passwort
RFC	RFC 4253 (Server)	RFC 15 →854

[3]

4 SSH Layers

- **Application Layer**: SSH

  ssh-connection: Session-Multiplexing, Remote-Befehle, Dateien übertragen, etc.

  ssh-userauth: Benutzerautorisierung mittels public key, password, etc.

  ssh-transport: Key-Austausch, Serverauthentifizierung, Verschlüsselung, etc.

- **Transport Layer**: TCP
- **Network Layer**: IP
- **Link Layer**: Ethernet

Wie man sehen kann, ist SSH nicht mehr dafür zuständig, wie die Daten übertragen werden. Das macht dann TCP.

5 SSH Verbindungsablauf

1. Ein kryptografischer Handshake wird durchgeführt.
2. Die Verbindung zwischen Client und Server wird verschlüsselt, indem eine Diffie-Hellman-Key-Exchange durchgeführt wird.
3. Client authentifiziert sich mit einem Passwort oder einer public key.
4. Client kann nun sicher und verschlüsselt mit Server interagieren.

[4]

6 SSH Befehle

ssh username@hostname

- **ssh** ist ein Programm, das eine Verbindung zwischen Client und Server aufbaut.
- **username** ist der Benutzername, mit dem der Client sich mit dem Server authentifiziert.
- **hostname** ist der Hostname oder IP-Adresse des Servers.

6.1 Argumente

- **ssh -p** Ändert die Portnummer, an der die Verbindung gestartet werden soll.
- **ssh -1** verwendet SSH-1.
- **ssh -2** verwendet SSH-2.
- **ssh -4** verwendet IPv4.
- **ssh -6** verwendet IPv6.
- **ssh -V** zeigt die Version des Protokolls an.
- **ssh -D** Ändert die Portnummer, an die der Server die Verbindung überwachen soll.
- **ssh -E log_file** speichert die Log-Datei in log_file.

6.2 Andere SSH-Befehle

- **ssh-keygen** erzeugt einen neuen SSH-Key.
- **ssh-copy-id** speichert einen SSH-Key als äuthorisedän einem Server.
- **ssh-agent** erzeugt einen SSH-Agent.
- **ssh-add** speichert einen SSH-Key im SSH-Agent.
- **scp** überträgt Dateien mit einem RCP-ähnlichen Command-Interface.
- **sftp** überträgt Dateien mit einem FTP-ähnlichen Command-Interace.

[5]

7 SSH Keys

7.1 Asymetrische Kryptografie

Bekannte asymetrische Verschlüsselungsverfahren sind RSA und DSA. Im Gegensatz zu symmetrischen Verschlüsselungsverfahren arbeiten diese Algorithmen mit zwei verschiedenen Schlüsseln. Diese beiden Schlüsseln bilden einen Schlüsselpaar, das für jeden Benutzer spezifisch ist.

7.2 Schlüsselpaar - Public und Private

Im Anwendungsfall der SSH-Authentifizierung mit öffentlichem Schlüssel ist es ziemlich typisch, dass die Benutzer das Schlüsselpaar selbst erstellen. SSH-Implementierungen enthalten hierfür einfach zu verwendende Dienstprogramme(ssh-keygen etc.).

Ein Schlüsselpaar besteht aus:

- **Public Key** - öffentlicher Schlüssel.

  Ein öffentlicher Schlüssel wird auf den Server gesendet und dort gespeichert. Damit können die Benutzer die Daten verschlüsselt übertragen. Diese Daten können nur vom Benutzer mit dem dazugehörigen privaten Schlüssel entschlüsselt werden. Sobald ein SSH-Server einen öffentlichen Schlüssel von einem Benutzer erhält und diesen als vertrauenswürdig einstuft, markiert der Server den Schlüssel in seiner Datei authorized_keys als autorisiert.

- **Private Key** - privaten Schlüssel.

  Der private Schlüssel, bleibt nur beim Benutzer. Der Besitz dieses Schlüssels ist der Beweis für die Identität des Benutzers. Nur ein Benutzer, der im Besitz eines privaten Schlüssels ist, der mit dem öffentlichen Schlüssel des Servers übereinstimmt, kann sich erfolgreich authentifizieren.
  Sorgfältig aufbewahren!

[6]

8 SSH Fingerprints

Wenn sich ein Client das erste mal mit einem Server verbindet, wird ein sogenannter **Fingerprint** erzeugt. Dieser Fingerprint ist ein Hashwert, der aus dem Public Key des Servers besteht. Wenn sich bei der nächsten Verbindung die Fingerprint-Werte ändern, wird darauf hingewiesen, dass die Verbindung nicht vertrauenswürdig ist. Dieses Prinzip nennt man **Trust on first use**. Allerdings ist es möglich, das Ganze mit einem **Man in the Middle** Angriff auszunutzen, wenn der Fingerprint noch nicht bekannt ist.[7]

9 Port-Forwarding

SSH-Port-Forwarding nutzt man zum Tunneln von Anwendungsports vom Client-Rechner zum Server-Rechner oder umgekehrt. Es kann dazu verwendet werden, Firewalls zu umgehen, und Einige nutzen ihn, um Hintertüren in das interne Netz von ihren eigenen Rechnern aus zu öffnen. Es kann jedoch auch von Hackern und Malware missbraucht werden.

9.1 Local Forwarding

Es wird verwendet, um einen Port vom Client-Rechner zum Server-Rechner weiterzuleiten. Grundsätzlich lauscht der SSH-Client auf Verbindungen an einem konfigurierten Port, und wenn er eine Verbindung erhält, tunnelt er die Verbindung zu einem SSH-Server. Der Server stellt eine Verbindung zu einem konfigurierten Zielport her, der sich auch auf einem anderen Rechner als der SSH-Server befinden kann.
Anwendungen:

- Verbindung zu einem Dienst in einem internen Netzwerk von außen
- Verbinden mit einer entfernten Dateifreigabe

Syntax:

ssh -L local_port:remote_host:remote_port username@hostname

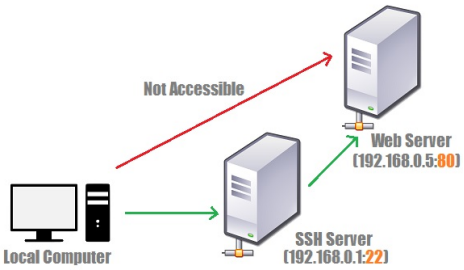

Abbildung 2: Local Forwarding

9.2 Remote Forwarding

Mit **Remote Forwarding** kann jeder in dem entfernten Netzwerk eine Verbindung zum **remote_port** auf dem Server herstellen. Die Verbindung wird dann zurück zum Client getunnelt, und der Client stellt dann eine TCP-Verbindung zum **local_port** auf localhost her.

Zum Beispiel kann man **Remote Forwarding** verwenden, um jemandem von außen Zugang zu einem internen Dienst zu geben.

Syntax:

ssh -R remote_port:localhost:local_port username@hostname

[8]

10 Sicherheit

SSH kann mit folgenden Attacken gefährdet werden:

- **Man-in-the-middle**

  Dies geht nur bei der ersten Authentifizierung

  Private- und Public- Keys schützen davor

- **DDoS**

  Beispielweise mit Cloudflare kann man sich davor schützen

- **Bruteforce**

  Fail2Ban benutzen

  Bei zu vielen falschen Passwörtern wird die IP-Adresse gebannt.

  Starkes Passwort benutzen

  SSH-Keys!

11 SFTP

11.1 Was ist SFTP?

SFTP steht für Secure File Transfer Protocol und ist ein Netzwerkprotokoll auf der 7. Schicht, um die Übertragung von Dateien zwischen verschiedenen Computersystemen zu regeln. Wie auch beim FTP findet die Kommunikation nach einem Client-Server-Prinzip statt. Ein Client interagiert dabei mit einem Server, um Dateien herunter- bzw. hochzuladen und auch die Ordnerstruktur auf dem Server zu verändern.

11.2 SFTP vs FTP

Wie der Name bereits verrät, handelt es sich bei einem SFTP um eine gesicherte Version des ursprünglichen FTPs. Für die Übertragung von Daten setzt SFTP auf SSH, wodurch sich der Client am Server authentifizieren muss. Bei der gesamten Kommunikation sind die Befehle und Daten somit verschlüsselt. Ein weiterer Unterschied liegt an der Anzahl Verbindungen. Während der FTP zwei Verbindungen (eine Datenübertragungs- und Steuerverbindung) benötigt, reicht beim SFTP alleine eine Verbindung.

11.3 Wie kann man SFTP benutzen?

Um SFTP zu benutzen, braucht man als erstes SSH-Zugang. Nun kann man SFTP entweder textuell im Terminal benutzen, oder man lädt sich ein Programm wie FileZille, WinSCP, etc. runter.[9]

Literatur

[1] Michael Cobb Peter Loshiski. Secure shell (ssh). https://www.computerweekly.com/de/definition/Secure-Shell-SSH, 02 21.

[2] OmniSecu. History of ssh protocol. https://www.omnisecu.com/tcpip/history-of-ssh-secure-shell.php.

[3] Tutorial Point. Differences between ssh and telnet. https://www.tutorialspoint.com/difference-between-ssh-and-telnet.

[4] Red Hat Enterprise Linux. Die abfolge einer ssh-verbindung. https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-de-4/s1-ssh-conn.

[5] SSH Academy. Ssh command - usage, options, configuration. https://www.ssh.com/academy/ssh/command.

[6] SSH Academy. What is ssh public key authentication? https://www.ssh.com/academy/ssh/public-key-authentication.

[7] Wikipedia. Schlüssel (kryptologie). https://de.wikipedia.org/wiki/Schl%C3%BCssel_(Kryptologie)#Schl%C3%BCssel_bei_asymmetrischen_Verfahren.

[8] SSH Academy. Ssh port forwarding - example, command, server config. https://www.ssh.com/academy/ssh/tunneling/example.

[9] IP Switch. Sftp-server. https://www.ipswitch.com/de/ressourcen/best-practices/sftp-server.