Handout, 7. Mai 2019

Secure Shell (SSH)

Marvin Gaube[1], Sidney Kuyateh[1], Steffen Walter[1]

Zusammenfassung
Das vorliegende Handout soll einen kurzen Überblick über den Aufbau und die Anwendung des Protokolls „SSH“ geben. Nach einem kurzen Einstieg in die Geschichte werden hierzu die drei standardisierten Schichten mit zugehörigen RFC's betrachtet:

- SSH Transport Layer Protocol (RFC 4253)
- SSH Authentication Protocol (RFC 4252)
- SSH Connection Protocol (RFC 4254)

Abschließend wird kurz auf die Implementierungen eingegangen.

[1] *Studiengang Informationstechnik, Fakultät Technik, Duale Schule Baden-Württemberg, Stuttgart*
***Corresponding author**: inf17001@lehre.dhbw-stuttgart.de

Inhaltsverzeichnis

Einleitung

1. Vorgänger

Sowohl das Protokoll also auch die zugehörigen Werkzeuge zu SSH haben Vorgänger welche im Folgenden kurz genannt werden.

1.1 Telnet

Das Teletype Network (Telnet) Protokoll wurde 1969 im Rahmen eines Forschungsprojekts entwickelt. Es handelt sich um ein Protokoll welches den zeichenorientierten Datenaustausch über eine TCP Verbingung beschreibt. Telnet verfolgt dabei eine Client/Server-Modell. Der große Nachteil an diesem Protokoll ist es, dass Daten unverschlüsselt übers Netzwerk ausgetauscht werden. Gerade bei Zugangsdaten ist dies sehr problematisch.

1.2 rlogin, rsh, rcp

Bei „Remote Login“, „Remote Shell“ und „Remote Copy“ handelt es sich um Werkzeuge die alle das Telnet Protokoll verwenden. Während rlogin und rsh zur Fernsteuerung eines Computersystems verwendet werden können, handelt es sich bei rcp um ein Programm welches es ermöglicht Kopiervorgänge über Computergrenzen hinaus zu realisieren.

2. Geschichte von SSH

Von der Entwicklung bis heute hat sich einiges getan im Kontext des SSH Protokolls und seiner Werkzeuge.

2.1 SSH-1

Nachdem er Opfer eines Hackerangriffs wurde, wobei sein Passwort in fremde Hände gelangte entwickelte Tatu Ylönen im Jahr 1995 die ersten Implementierung von SSH. Diese Software wurde zuerst unter einer freien Lizenz veröffentlicht. Auf Grund einer Vielzahl an Support Anfragen gründete Ylönen bald die Firma SSH Communications Security Corp. (SCS) um kommerziellen Support anbieten zu können. Im selben

Jahr brachte Ylönen die Dokumentation des Protokolls auch bei der IEFT als Internet Draft ein.

2.2 SSH-2

Im Jahr 1996 veröffentlichte SCS die Version 2 des SSH Protokolls, welches um einige nützliche Funktionen für Unternehmen erweitert wurde. Die IEFT bildete daraufhin eine Arbeitsgruppe welche sich damit befasste das Protokoll zu Gunsten der Allgemeinheit zu Standardisieren. Diese Arbeitsgruppe brachte 1997 den ersten Internet Draft für SSH-2.0 ein. Im folgend n Jahr veröffentlichte SCS ihre erste Implementierung von SSH-2.0 welche allerdings unter einer weitaus restriktiveren Lizenz veröffentlicht wurde als sein Vorgänger. Aus diesem Gru: undg auf Grund des Erscheinens von OpenSSH konnte sich SSH2 am Markt nie durchsetzten.

2.3 OpenSSH

Die Lizenzpolitik von SCS motivierte einige Entwickler des OpenBSD Projekts sich zusammen zu tun und an der letzten freien Veröffentlichung von SSH1 (Version 1.2.12) weiter zu entwickeln. Diese Initiative hat bis heute großen Erfolg so schaffte es OpenSSH heute zu einer der wichtigsten SSH Implementierungen zu werden, welche neben OpenBSD auch auf Linux, Solaris, AIX, Mac OS X und weitere Plattformen portiert werden konnte. SCS entwickelt seinerseits weiter an ihrer Implementierung, ist aber neben OpenSSH zu einem eher kleinen Faktor geworden.

3. Komponenten

SSH besteht im wesentlichen aus drei Komponenten, deren Zusammenspiel im RFC4251 beschrieben wird:

- SSH Transport Layer Protocol (RFC 4253)
- SSH Authentication Protocol (RFC 4252)
- SSH Connection Protocol (RFC 4254)

Üblicherweise werden bei einem Verbindungsaufbau die verschiedenen Teilprotokolle in dieser Reihenfolge durchlaufen: Zunächst wird der Transport Layer aufgebaut, auf diesem wird der Nutzer authentifiziert, und danach die eigentliche Nutzdatenverbindung aufgebaut.
Im RFC4250 sind allgemeine Informationen definiert, zum Beispiel die zu den Kommandos gehörenden Enumerationen.

3.1 Transport Layer Protocol

Dieser Layer ist im RFC4253 definiert.
Im TCP/IP Protokollstapel läuft SSH als Application im Layer 5-7, und verwendet TCP im Layer 4. Dem Protokoll wurde der Port 22 zugewiesen.
Der Paketaufbau sieht so aus:

- uint32 packet_length (ggf. encrypted)
- byte padding_length (ggf. encrypted)
- byte[] payload; (ggf. encrypted)
- byte[] random padding; (ggf. encrypted)
- byte[m] mac (Message Authentication Code - MAC)

Der Handshake läuft vereinfacht folgendermaßen ab:
Durch KEXINIT einigen sich Client und Server auf die verwendeten Algorithmen.
Der Key Exchange selbst erfolgt üblicherweise über Diffie-Hellman in Kombination mit dem Hostkey als Authentifizierung.
Nach SSH_MSG_NEWKEYS wird auf die neu verhandelte Verschlüsselung und Message Authentication umgestellt. Der Transport Layer authentifiziert nur den Host gegenüber dem Client, nicht anders herum! Nach dem Aufbau der Verbindung geht es mit SSH_MSG_SERVICE_REQUEST service in den nächsten Layer.
„service“ ist üblicherweise „ssh-userauth“.

3.2 Authentication Protocol

Die Definition der Authentifizierung erfolgt im RFC 4252.
Ablauf einer Authentifizierung:
Client → Server: SSH_MSG_USERAUTH_REQUEST (mit Username, Service, Auth-Methode)
Server → Client: SSH_MSG_USERAUTH_FAILURE oder SSH_MSG_USERAUTH_SUCCESS.
In letzterem Fall erfolgt ein Üpgrade”, der gewünschte Service wird gestartet (ssh-connection).
Folgende Authentifizierungsmethoden sind verfügbar:

- publickey (required)
- password
- hostbased
- none

3.3 Connection Protocol

Im Layer des „Connection Protocols“ (RFC4254) liegen die eigentlichen Anwendungen.
Hier ist es möglich, verschiedene Anwendungen über die Selbe Verbindung zu multiplexen - dafür werden sog. „Channel“ verwendet.

3.3.1 Interaktive Shell
3.3.2 SSH-Tunnel
3.3.3 X-Forwarding
3.3.4 L3-Tunnel
3.3.5 SCP
3.3.6 SFTP
3.3.7 SSHFS

4. Implementierungen

4.1 SSH Tectia
4.2 OpenSSH
4.3 Putty

Putty ist eine reine Client-Implementierung von SSH, die insbesondere unter Windows beliebt ist. Die GUI ist minimal gehalten; es werden viele Versionen und auch andere Protokolle unterstützt.

4.4 Dropbear

Dropbear ist eine freie, minimale Implementierung. Daher wird Dropbear häufig ein eingebetteten Systemen, wie zum Beispiel OpenWRT, eingesetzt. Nicht benötigte Funktionen können einfach entfernt werden.