

SSH – Grundlagen & Tricks

Andreas Schiermeier

LUG Frankfurt, 01.08.2007

SSH – Grundlagen & Tricks

- ...mehr als nur ein sicherer Login
- Protokollgrundlagen (SSHv2)
- täglicher Einsatz
- Serverkonfiguration & -absicherung
- Was gibt's noch?

...mehr als nur ein sicherer Login

- sicherer Dateitransfer (sftp, scp)
- TCP-Weiterleitung
- X11-Weiterleitung
- sicherer Kanal für diverse Anwendungen (rsync, rdiff-backup, tar)

Protokollgrundlagen (SSHv2)

- transport layer
- user authentication layer
- connection layer

transport layer

- Datenkomprimierung
- Verschlüsselung
- Integritätsprüfung
 - wer ist mein Gegenüber?
 - kann ich ihm (noch) vertrauen?
 - *The authenticity of host 'mein.server.de (127.6.3.5)' can't be established.
RSA key fingerprint is f0:b8:e4:5f:3d:05:e0:68:f7:83:35:ee:f1:4b:fa:64.
Are you sure you want to continue connecting (yes/no)?*

user authentication layer

- Benutzeranmeldung über verschiedene Methoden
 - password
 - Benutzername und Passwort werden zur Prüfung an den Server geschickt
 - keyboard-interactive
 - Server kann Eingabeaufforderungen zum Client schicken (PAM, OTP, Tokens)
 - publickey
 - Anmeldung durch einen hinterlegten öffentlichen Schlüssel

connection layer

- Aufteilung der Verbindung in virtuelle Kanäle
 - shell
 - direct-tcpip
 - forwarded-tcpip
- jeder Kanal kann mehrmals vorhanden sein
 - praktisch bei langsamen GPRS-Verbindungen
 - ControlMaster/ControlPath
- X11- & Agentweiterleitung

täglicher Einsatz

- Anmeldung durch public/private-Key
- ssh-agent & Agent-Forwarding
- ~/.ssh/config
- X11-Forwarding
- Portweiterleitungen, SOCKS-Proxy
- „Punching holes in HTTP(s) proxies“
- tar, rsync, rdiff-backup
- Zeichensatzprobleme

Anmeldung durch public/private-Key

- erzeugen:
 - `gast@movingmind:~> ssh-keygen -b 1024 -t dsa`
Generating public/private dsa key pair.
Enter file in which to save the key (/home/gast/.ssh/id_dsa): <enter>
Created directory '/home/gast/.ssh'.
Enter passphrase (empty for no passphrase): <...>
Enter same passphrase again: <...>
Your identification has been saved in /home/gast/.ssh/id_dsa.
Your public key has been saved in /home/gast/.ssh/id_dsa.pub.
The key fingerprint is:
7b:85:c9:73:13:a8:7c:98:59:20:10:95:4d:58:ec:44 gast@movingmind
- auf den Server kopieren
 - `ssh-copy-id benutzername@mein.server.de`
 - oder mit einem Texteditor `~/.ssh/authorized_keys` anlegen und den Inhalt von `~/.ssh/id_dsa.pub` (lokal) einfügen
`chmod 700 ~/.ssh`
`chmod 600 ~/.ssh/authorized_keys`
- Fingerprint nachträglich anzeigen
 - `ssh-keygen -l -f ~/.ssh/id_dsa` (funktioniert auch mit Serverkeys unter `/etc/ssh/`)

ssh-agent & Agent-Forwarding

- zügiges Arbeiten: pro Sitzung nur einmal Passphrase eingegeben
- in zahlreichen Distributionen vorinstalliert oder alternativ zahlreiche HowTos (=> Google)
- Keys werden mit ssh-add dem Agent bekannt gemacht
- mit ssh -A benutzer@mein.server.de „wandert“ der Agent mit
- Achtung: bei unbekanntem oder möglicherweise komprimierten Zielsystemen Agentforwarding deaktivieren!
ssh -a benutzer@mein.server.de

~/.ssh/config

- häufig benötigte Server oder Optionen können fest hinterlegt werden

```
- Host srv1  
  Hostname mein.server.de  
  User admin
```

```
Host srv2  
  Hostname 127.4.7.1  
  User root
```

```
Host *  
  Protocol 2  
  HashKnownHosts yes
```

- Achtung: evtl. „Serverdatenbank“ für Angreifer (vs. HostKnownHosts)

X11-Forwarding

- X-Anwendungen auf dem Server starten und auf dem Client anzeigen
 - ssh -X benutzer@mein.server.de xeyes
- Nur bei vertrauenswürdigen Servern verwenden – X11-Zugriff auf den Client möglich!

Portweiterleitungen, SOCKS-Proxy

- Port auf dem Client öffnen und zum Server weiterleiten (LocalForward)
 - ssh -L8080:127.0.0.1:80 benutzer@mein.server.de
- Port auf dem Server öffnen und zum Client weiterleiten (RemoteForward)
 - ssh -R8080:127.0.0.1:80 benutzer@mein.server.de
- SOCKS-Proxy (für Browser und E-Mailclients) auf dem Client öffnen (DynamicForward)
 - ssh -D1080 benutzer@mein.server.de

„Punching holes in HTTP(s) proxies“

- durchbrechen von HTTPS-Proxys & Firewalls mit <http://proxytunnel.sourceforge.net/>
 - Host *
 - ProtocolKeepAlives 15
 - ProxyCommand proxytunnel -p proxyserver:8080 -u proxyuser -s proxypasswort -d %h:%p
 - ggf. nur Port 443 erlaubt; ggf. Server (zusätzlich) auf anderen Port legen – oder via iptables umleiten (REDIRECT-Target)
- Alternativen:
 - <http://www.nocrew.org/software/httpptunnel.html>
 - <http://sebsauvage.net/punching/>

tar, rsync, rdiff-backup

- tar cvzf - | ssh benutzer@mein.server.de „cd /ziel/verzeichnis; tar xzf -“
- integrierter SSH-Support
 - rsync: Dateien/Ordner auf zwei Systemen synchron halten
 - rdiff-backup: bandbreitenschonendes, inkrementelles Backup

Zeichensatzprobleme

- Problem: lokal wird ein anderer (z.B. UTF-8) Zeichensatz verwendet als auf dem Server (ISO8859-15)
- Lösung:
 - luit -encoding ISO8859-15 ssh user@mein.server.de

Serverkonfiguration & -absicherung

- Konfigurationsdatei: `/etc/ssh/sshd_config`
 - nicht verwechseln mit `/etc/ssh/ssh_config`
- root-Login?
- Passwortlogin deaktivieren?
 - Achtung: UsePAM!
- Port umkonfigurieren?
- Portknocking?

root-Login?

- was spricht dagegen?
 - nach dem Login voller Systemzugriff
 - besonders gefährlich bei schwachen Passwörtern
- ...und was dafür?
 - bequemes Kopieren von Dateien zwischen Rechnern unter Beibehalt der UIDs, GIDs und Rechte
 - Backups...
 - Portforwarding von Ports <1024

root-Login?

- Lösung

- root-Login nur mit Keys erlauben:

- PermitRootLogin without-password

- oder

- PermitRootLogin forced-commands-only

- in `/root/.ssh/authorized_keys`

- command="rdiff-backup --server --restrict-read-only /",no-port-forwarding,no-X11-forwarding,no-agent-forwarding,from="127.6.9.3"*
 - ssh-dss AAAAB3Nza...*

Passwortlogin deaktivieren?

- können alle Benutzer mit Keys umgehen?
- von wo aus benötige ich Serverzugriff
 - ist überall dort mein Key verfügbar?
- wenn nicht benötigt:
 - PubkeyAuthentication yes
 - PasswordAuthentication no
 - ChallengeResponseAuthentication no
 - UsePAM no
 - wichtig!

Port umkonfigurieren?

- contra: konsequenter Angreifer findet auch den Alternativport; „Security through obscurity“
- contra: bricht Kompatibilität zu Anwendungen, die SSH als Backend verwenden
- pro: hält das Log mangels fehlgeschlagener BruteForce-Logins sauberer
- wer's braucht...:
 - Port 12345 statt Port 22

Portknocking?

- zusätzliche Fehlerquelle
- macht den Serverzugang im Notfall unnötig kompliziert oder unmöglich
- besser: System auf einem aktuellen Stand halten!

Was gibt's noch?

- einzelne Public-Keys weiter einschränken (per Quell-IP, auszuführendem Befehl, Forwardingberechtigungen)
- natives VPN über SSH
- PPP-over-SSH
 - pppd persist noauth nodetach silent 172.16.0.1:172.16.0.2 pty "ssh -t root@mein.server.de pppd noauth nodetach"
 - (im Vorfeld Route für mein.server.de setzen)
- SSH-Jumpgates vor Serverfarmen
- Serverfingerprints im DNS ablegen
- Fragen?

Danke für euere Aufmerksamkeit!

Andreas Schiermeier
rh-tec Business GmbH