SSH im praktischen Einsatz

Christian Rode
Rechenzentrum Uni Würzburg

SSH im praktischen Einsatz - Übersicht

- SSH-Grundlagen (Konfigurationsdateien, Identitäten)
- Schlüssel generieren und prüfen (ssh-keygen)
- Zugriffskontrolle (3-stufig: iptables, hosts.allow, authorized_keys)
- Authentication Agent (ssh-agent, ssh-add)
- SSH Port Tunneling (ssh -L, ssh -R)
- weitere Anwendungsbeispiele (Shell-Skripte, rsync-Backups, ..)
- typische Fallstricke (Identitäten, Host Keys, ..)

SSH-Grundlagen

Client (ssh(1), scp(1)):

- /etc/ssh/ssh_config (ssh_config(5))
- ~/.ssh/known_hosts
- Verbose-Option (ssh -v)

SSH-Identitäten (ssh -i, scp -i):

- Server: Host Keys (/etc/ssh/*_key*)
- Client: User Keys (~/.ssh/id_*)
- ssh -i <user_key>
- ~/.ssh/authorized_keys:

  from="..", command=".."

Server (sshd(8)):

- /etc/ssh/sshd_config (sshd_config(5))
- ~/.ssh/authorized_keys
- /etc/hosts.allow
- Subsysteme (sftp(1))

X11-Forwarding:

- Client: ssh -X, $DISPLAY
- Server: X11Forwarding

wichtige Tools:

- ssh-agent(1), ssh-add(1),
- ssh-keygen(1)

SSH-Schlüssel generieren

```
test01:~ # ssh-keygen -t rsa -b 2048 -f /root/.ssh/id_rsa_test -N ''
Generating public/private rsa key pair.
Your identification has been saved in /root/.ssh/id_rsa_test.
Your public key has been saved in /root/.ssh/id_rsa_test.pub.
The key fingerprint is:
73:f5:2a:20:2a:2b:ab:fe:de:c8:74:c8:ca:d8:3b:46 root@test01
```

```
test01:~ # ssh-keygen -p -f /root/.ssh/id_rsa_test
Key has comment '/root/.ssh/id_rsa_test'
Enter new passphrase (empty for no passphrase): ********
Enter same passphrase again: ********
Your identification has been saved with the new passphrase.
```

```
test01:~ # cat /root/.ssh/id_rsa_test.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAwlus0XceRTDEaM/NVH+MFmVxV8NinCziwf34
...
FwWpa4/Jkzx09iVN3qnF5QlyZa25keCvFkWWeEN9PCGpWIxQ+NGnaAN7FXLiG4P4n6aX2Ne5
fcc9VDGZOcS6aX4t6Sc0WQyon5gA1w== root@test01
```

SSH-Schlüssel prüfen (1/2)

Ein neuer Host Key (RSA oder DSA) sollte nur dann akzeptiert werden ...

```
test01:~ # ssh -l root bkup01
The authenticity of host 'bkup01 (172.25.1.140)' can't be established.
RSA key fingerprint is fe:2c:c5:a9:c8:f9:aa:dd:4a:b8:8e:89:df:45:15:99.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'bkup01,172.25.1.140' (RSA) to the list of known
hosts.
```

... wenn der zugehörige Host Key Fingerprint verifiziert wurde:

```
bkup01:~ # ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
1024 fe:2c:c5:a9:c8:f9:aa:dd:4a:b8:8e:89:df:45:15:99 /etc/ssh/ssh_host_
rsa_key.pub
```

SSH-Schlüssel prüfen (2/2)

Nur in Ausnahmefällen sollte auf diese Prüfung verzichtet werden, z.B. bei nicht-interaktiven Shell-Skripten, die in einer sicheren Umgebung laufen:

```
#!/bin/bash
...
export SSH_AUTH_SOCK=
scp -q -o StrictHostKeyChecking=no -i $identity -l $user $host $src $dest
...
```

SSH-Zugriffskontrolle (hier: 3-stufig)

- Personal Firewall (iptables)

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp -s 132.187.4.0/24 --dport 22 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp -s 132.187.3.56 --dport 22 -m state --state NEW -j ACCEPT
iptables -P INPUT DROP
```

- TCP-Wrappers (/etc/hosts.allow, /etc/hosts.deny)

```
# cat /etc/hosts.deny
sshd:   ALL EXCEPT LOCAL

# cat /etc/hosts.allow
sshd:   132.187.4.0/255.255.255.0
sshd:   132.187.3.56
```

- SSH-Identität (~/.ssh/authorized_keys)

```
# cat ~/.ssh/authorized_keys
from="132.187.4.*,wrzx56.rz.uni-wuerzburg.de",command="rsync --server --sender -lo
gDtprz –delete-excluded --numeric-ids . /",no-pty,no-port-forwarding,no-agent-forw
arding,no-X11-forwarding ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAw6rGM01v5lUQutc82Wsg1
GLhOTj/ruqsGAa7xPJmOXsSbNSi4VztEblxtcfK642JYAVJlu94D1m8hbT6yGYZNGwE= root@bkupserv
...
```

SSH Authentication Agent (1/4)

den SSH Agenten starten ...

```
$ eval `ssh-agent -s`
Agent pid 9455
```

... und den Private User Key bekannt geben:

```
$ ssh-add
Enter passphrase for /home/rode/.ssh/id_rsa: ********
Identity added: /home/rode/.ssh/id_rsa (/home/rode/.ssh/id_rsa)
```

Nun funktioniert ssh überall dort ohne Eigabe eines Passworts, wo der Public User Key in ~/.ssh/authorized_keys eingetragen ist, und zwar über beliebig viele Hops:

```
$ ssh root@test01
Last login: Mon Jan 30 11:49:50 2006 from wrsx42.rz.uni-
wuerzburg.de
Have a lot of fun...
```

SSH Authentication Agent (2/4)

Die im SSH Agenten gespeicherten Schlüssel lassen sich auf einem beliebigen solchen 'SSH-Hop' anschauen ...

```
$ ssh-add -l
4096 bd:bf:fc:a9:9d:a3:ea:8f:13:13:e3:46:77:7c:be:78 /home/rode
/.ssh/id_rsa (RSA)
```

... oder sperren ...

```
$ ssh-add -x
Enter lock password: *****
Again: *****
Agent locked.
```

... und wieder entsperren.

```
$ ssh-add -X
Enter lock password: *****
Agent unlocked.
```

SSH Authentication Agent (3/4)

Der SSH Agent kann z.B. automatisch beim interaktiven Login auf dem Sprungbrett-Rechner gestartet werden ...

```
$ cat ~/.bash_login

mytty="$(tty)"
if [ "${mytty:0:9}" = '/dev/pts/' ]; then
        eval `ssh-agent -s` && ssh-add
fi
```

... oder über das PAM-Modul “pam_ssh(8)” direkt in das graphische Login integriert werden (“UsePAM yes” in sshd_config(5) sowie “auth sufficient pam_ssh.so” in /etc/pam.d/xdm).

SSH Authentication Agent (4/4)

Fazit: Der SSH Agent läuft auf dem ursprünglichen Client-Host und funktioniert völlig transparent über alle SSH-Verbindungen, die das Agent Forwarding unterstützen (“ForwardAgent yes” in ssh_config(5) bzw. “ssh -A”).

Allerdings kann sich ein Angreifer mit Root-Rechten auf irgend einem dieser Hosts mit Hilfe des vom SSH Agent verwendeten Unix Sockets ebenfalls für alle anderen Hosts authentifizieren!

Deshalb sollte der Forwarding-Mechanismus des SSH Agents für unsichere Hosts gezielt unterdrückt werden (“ForwardAgent no” in ssh_config(5) bzw. “ssh -a”)

SSH Port Tunneling

zu Hause (Root erforderlich, da Port 443 ein privilegierter Port (<1024) ist):

```
# cat /etc/hosts
...
127.0.0.1       localhost www.rz.uni-wuerzburg.de

# ssh -L 443:localhost:1027 -l rode 132.187.4.42
```

auf 132.187.4.42 (beliebiges Sprungbrett, Root nicht erforderlich, da Port 1027):

```
$ ssh -L 1027:www.rz.uni-wuerzburg.de:443 -l rode 132.187.4.42
```

jetzt zu Hause im Browser https://www.rz.uni-wuerzburg.de:443/ eingeben, und der Web-Server bekommt eine uni-interne Verbindung (Alternativen: VPN, w3m)

Beachte: Damit 'ssh -L' nicht nur auf 127.0.0.1 (localhost) sondern auf 0.0.0.0 (alle Interfaces) hört, muß auf 132.187.4.42 in /etc/ssh/sshd_config 'GatewayPorts yes' eingetragen sein, sonst funktioniert's nicht!

weitere Anwendungsbeispiele

- Shell-Skripte

```
#!/bin/bash
...
export SSH_AUTH_SOCK=
tar czf - data | ssh -i $identity $user@$host “cd /bkup && tar xzf -”
...
```

- rsync-Backups

```
/usr/bin/rsync --stats --numeric-ids --exclude '/proc'            \
         --exclude '/sys' --exclude '/media/*' --delete-excluded  \
         -aze 'ssh -i /root/.ssh/id_rsa_mirror -l root'           \
         test01:/ /bkup/test01
```

- sicherer Zugang zum CVS-Repository

```
# cat ~/.bashrc

export CVSROOT=:ext:cvsuser@cvshost:/var/cvs
export CVS_RSH=ssh
```

SSH-Fallstricke (1/2)

- auf dem Linux-System gibt es andere SSH-Konfigurationsdateien als die für SuSE9.x/SLES9 beschrieben (ssh(1), sshd(8)), z.B. Red Hat 7.3:

```
$ ls -l .ssh/authorized_keys2
-rw-r--r--  1 ben  dattrans  3342 Aug  1  2003 .ssh/authorized_keys2

$ ls -l .ssh2/*
-rw-------  1 ben  dattrans    20 Dez 13 10:15 .ssh2/identification
-rw-------  1 ben  dattrans  1555 Dez 13 10:15 .ssh2/id_rsa_2048_a
-rw-r--r--  1 ben  dattrans   547 Dez 13 10:15 .ssh2/id_rsa_2048_a.pub

.ssh2/hostkeys:
-rw-------  1 ben  dattrans   769 Dez 13 09:59 key_22_maildb.pub
-rw-------  1 ben  dattrans   773 Dez  1 18:03 key_22_mailmaster.pub

$ cat .ssh2/identification
IdKey id_rsa_2048_a
```

SSH-Fallstricke (2/2)

- die in /etc/ssh/sshd_config getroffene Einstellung wird nicht wirksam ("rcsshd restart" vergessen)
- die mit "ssh -i" angegebene Identität wird nicht verwendet (ssh-add -x, $SSH_AUTH_SOCK)
- der SSH-Login funktioniert nicht:
  - keine aussagekräftige Fehlermeldung (hosts.allow, authorized_keys)
  - Dateirechte der Private-Key-Dateien stimmen nicht
  - SSH-Version 1 muß freigeschaltet werden (/etc/ssh/sshd_config)
- X11-Forwarding klappt nicht (xauth, ssh -X, /etc/ssh/sshd_config)
- nach der Server-Neuinstallation funktionieren Shell-Skripte nicht mehr (geänderte Host Keys)
- die Public-Key-Datei liegt im falschen Format vor (ssh-keygen -i, ssh-keyconverter)