Secure Shell

Uwe Grossu / Matthias Rumitz

AS-SYSTEME

DOAG 18.11.2015

Grundlagen

Fingerprint überprüfen

Anwendungs-Beispiele

Man-in-the-Middle Attack

Einloggen ohne Passwort

Einloggen ohne Passphrase

SSH-Tunnel Anwendungen

Version des SSH-Packages

Name des Packages enthält die Version

pkg list pkg:/service/network/ssh@0.5.11-0.175.1.0.0.24.2

ssh -V OpenSSH_5.11

strings /usr/lib/ssh/sshd | grep OpenSSH OpenSSH_5.1p

SSH-Server als SMF-Service

```
server$ svcs -p svc:/network/ssh:default
STATE          STIME          FMRI
online         Sep_23         svc:/network/ssh:default
               Sep_23         353 sshd

server$ ps -ef | grep sshd
    root   353     1   0   Sep 23 ?           0:00 /usr/lib/ssh/sshd
    root  4006  3904   0   16:29:36 pts/1     0:00 grep sshd

server$ svcadm enable svc:/network/ssh:default
```

Server Keys

Eigenes Public/Private-Schlüsselpaar:

server$ ls -l /etc/ssh/ssh_host_rsa_key*

-rw------- 1 root root 1675 ... 14:50 /etc/ssh/ssh_host_rsa_key

-rw-r--r-- 1 root root 396 ... 14:50 /etc/ssh/ssh_host_rsa_key.pub

Public/Private-Schlüsselpaar erzeugen:

server$ ssh-keygen -q -f /etc/ssh/ssh_host_rsa_key -N ""

Asymmetrische Public und Private Keys

server$ cat /etc/ssh/ssh_host_rsa_key.pub

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAs0tRAZXy5TeD4T

.... gekürzt ....

server$ cat /etc/ssh/ssh_host_rsa_key

-----BEGIN RSA PRIVATE KEY-----

MIIEowIBAAKCAQEAs0tRAZXy5TeD4TiX+ewKEwqLUmsR6rvm6w

.... gekürzt ....

+SHcIdDX12Diqzncq4fi2pfcT3Rq7k8dyI2z2N5luaSTZnUchcUm

-----END RSA PRIVATE KEY-----

Konfigurations-Dateien des SSH-Client

client$ ls -l /etc/ssh/ssh_config (einzige Datei nach Installation)

-rw-r--r-- 1 root root 2705 2015-05-09 14:21 /etc/ssh/ssh_config

Notwendige Dateien:

/etc/ssh/ssh_config

~/.ssh/config

~/.ssh/id_rsa

~/.ssh/id_rsa.pub

~/.ssh/known_hosts

Aufruf des SSH-Client

client$ ssh <host> (Host- / DNS-Name oder IP-Adresse)

Falls Benutzername auf <host> nicht bekannt ist:

client$ ssh user2@<host>

ODER

client$ ssh -l user2 <host>

Grundlagen

Fingerprint überprüfen

Anwendungs-Beispiele

Man-in-the-Middle Attack

Einloggen ohne Passwort

Einloggen ohne Passphrase

SSH-Tunnel Anwendungen

Erster Kontakt mit SSH-Server

client$ ls -l ~/.ssh/known_hosts

ls: cannot access ~/.ssh/known_hosts: No such file or directory

client$ ssh <host>

The authenticity of host '<host> (x.x.x.x)' can't be established.

RSA key fingerprint is

d9:d8:8a:31:6f:36:14:5d:08:b9:6b:57:78:9e:4c:32.

Are you sure you want to continue connecting (yes/no)?

Man-in-the-Middle ?

Fingerprint: eine durch eine Streuwertfunktion erstellte Prüfsumme

server$ ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
2048 d9:d8:8a:31:6f:36:14:5d:08:b9:6b:57:78:9e:4c:32
/etc/ssh/ssh_host_rsa_key.pub (RSA)

Sicher (!) übertragen und auf Client zur Verfügung stellen

?

Darstellung des Fingerprint

Enthält /etc/ssh/ssh_host_rsa_key.pub den “Public Key”?

server$ ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
2048 d9:d8:8a:31:6f:36:14:5d:08:b9:6b:57:78:9e:4c:32
/etc/ssh/ssh_host_rsa_key.pub (RSA)

Beide – Datei und Kommando – liefern ASCII-Darstellung.

Warum?

Darstellung des Fingerprint

Enthält /etc/ssh/ssh_host_rsa_key.pub den “Public Key”?

server$ ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
2048 d9:d8:8a:31:6f:36:14:5d:08:b9:6b:57:78:9e:4c:32
/etc/ssh/ssh_host_rsa_key.pub (RSA)

Beide – Datei und Kommando – liefern ASCII-Darstellung.

Warum? **Public-Key ist Binär**

Berechnung des Fingerprint

Public Key extrahieren:

**server$** sed -e 's/^ssh-rsa //' -e 's/==.*$/==/' \

/etc/ssh/ssh_host_rsa_key.pub > /tmp/key.pub

Public Key in numerischen Wert (zurück-)verwandeln:

server$ base64 -d /tmp/key.pub > /tmp/key.num

Prüfsumme berechnen:

server$ openssl md5 -c /tmp/key.num

MD5(key.num)=d9:d8:8a:31:6f:36:14:5d:08:b9:6b:57:78:9e:4c:32s

Grundlagen

Fingerprint überprüfen

Anwendungs-Beispiele

Man-in-the-Middle Attack

Einloggen ohne Passwort

Einloggen ohne Passphrase

SSH-Tunnel Anwendungen

SSH-Verbindungen

SSH-Server: (ohne Verbindung)

host1$ netstat -f inet -an | grep -w 22

```
tcp      0    0    0.0.0.0:22     0.0.0.0:*          LISTEN
```

SSH-Server: (mit Verbindung)

host1$ netstat -f inet -an | grep -w 22

```
tcp      0    0    0.0.0.0:22     0.0.0.0:*          LISTEN
tcp      0    0    host1:22       host2:48801    ESTABLISHED
```

Client ist "host2"

Kommandos via SSH ausführen

client$ ssh <host> <command ...>

Einloggen, Kommando ausführen, ausloggen.

Passwort beim Einloggen wird abgefragt:

client$ ssh <host> date

Password:

Tue Sep 15 15:06:12 CEST 2015

Secure File Copy

"scp"-Kommando ist ähnlich UNIX-"cp"

Syntax erlaubt u.a.

Übertragung mehrerer Dateien,

ganzer Dateibäumen (Option -r: recursively),

Beibehalten Datum/Zugriffsrechte (Option -p)

scp [[user1@]host1:]file1 ... [[user2@]host2:]file2

Secure File Transfer Protocol

"sftp"-Kommando ist ähnlich UNIX-"ftp"

client$ sftp user2@<host>
Connecting to <host>...
Password:
sftp> get -P wichtig.*
Fetching /home/user2/wichtig.txt to wichtig.txt
/home/user2/wichtig.txt 100% 6727 6.6KB/s 00:00
sftp> exit

X11-Forwarding

X-Client ist der X-Server

client$ ssh <host2>

…

server$ echo $DISPLAY

Localhost:10.0

server$ xclock

Grundlagen

Fingerprint überprüfen

Anwendungs-Beispiele

Man-in-the-Middle Attack

Einloggen ohne Passwort

Einloggen ohne Passphrase

SSH-Tunnel Anwendungen

Man-in-the-Middle Attack I

client$ ssh <host>

The authenticity of host '<host> (x.x.x.x)' can't be established.

RSA fingerprint is 8c:02:d6:e2:74:36:4a:d7:19:5a:3c:17:ba:c3:61:3d.

Are you sure you want to continue connecting (yes/no)? Yes

Password:

Last login: Wed Sep 16 15:06:57 2015 from <client>

....

server$ exit

logout

Connection to <host> closed.

Änderung des Public Key des SSH-Servers

client$ ssh <host>

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!

Someone could be eavesdropping on you right now ...!

It is also possible that the RSA host key has just been changed.

… gekürzt ….

Host key verification failed.

Der Verbindungsaufbau wird vom SSH-Clienten abgelehnt.

Debug-Modus I

client$ ssh -v <host2>

OpenSSH_5.1p1, OpenSSL 1.6.1f 28 March 2015

.... gekürzt ....

debug1: Trying private key: /home/user1/.ssh/id_rsa

debug1: Authentication succeeded (keyboard-interactive).

.... gekürzt ....

Last login: Thu Sep 17 17:05:21 2015 from <host1>

....

Connection to <host2> closed.

debug1: Exit status 0

Debug-Modus II

Debug-Modus III

client$ ssh -v -v -v <host2> 2> /tmp/ssh-protocol-3v

Password:

Last login: Thu Sep 17 17:26:16 2015 from <host1>

....

<host2>$ exit

logout

client$ cat /tmp/ssh-protocol-3v

OpenSSH_5.1p1, OpenSSL 0.9.8h 28 May 2008

.... gekürzt ....

debug3: check_host_in_hostfile: match line 1

Grundlagen

Fingerprint überprüfen

Anwendungs-Beispiele

Man-in-the-Middle Attack

Einloggen ohne Passwort

Einloggen ohne Passphrase

SSH-Tunnel Anwendungen

Einloggen ohne Passwort (ohne Passphrase) I

Erzeugen der User Keys (ohne Passphrase)

client$ ssh-keygen -q -N "" -C test.howtodo -f ~/.ssh/id_rsa

client$ ls -l ~/.ssh/id_rsa{,.pub}

-rw------- 1 <user1> staff 1675 ... /home/<user1>/.ssh/id_rsa

-rw-r--r-- 1 <user1> staff 393 ... /home/<user1>/.ssh/id_rsa.pubs

Public Key auf SSH-Server kopieren ….

Einloggen ohne Passwort (ohne Passphrase) II

Public Key auf SSH-Server in Datei ~/.ssh/authorized_keys.

client$ scp ~/.ssh/id_rsa.pub <host2>:\~/.ssh/authorized_keys

Password:

id_rsa.pub 100% 393 0.4KB/s 00:00

client$ cat ~/.ssh/id_rsa.pub | ssh <host2> 'cat >> \

~/.ssh/authorized_keys'

client$ ssh-copy-id -i ~/.ssh/id_rsa.pub <host2>

SS-Server Datei ~/.ssh/authorized_keys

server$ ls -l ~/.ssh/authorized_keys

-rw-r--r-- 1 <user1> users 393 .. /home/<user1>/.ssh/authorized_keys

… bzw. vom SSH-Clienten (ohne Passwort!)

client$ ssh <host2> ls -l \~/.ssh/authorized_keys

-rw-r--r-- 1 <user1> users 393 /home/<user1>/.ssh/authorized_keys

client$ ssh <host2>

Last login: Fri Sep 18 18:25:15 2015 from <client>

server$

Einloggen ohne Passwort (mit Passphrase)

Erzeugen der User Keys (mit Passphrase)

client$ ssh-keygen -q -f ~/.ssh/id_rsa -C howtodo3.ssh

Enter passphrase (empty for no passphrase): <passphrase>

Enter same passphrase again: <passphrase>

Public Key auf SSH-Server kopieren ….

BEISPIEL Einloggen ohne Passwort mit Passphrase

Grundlagen

Fingerprint überprüfen

Anwendungs-Beispiele

Man-in-the-Middle Attack

Einloggen ohne Passwort

Einloggen ohne Passphrase

SSH-Tunnel Anwendungen

Passphrase im SSA-Agent

client$ ps -ef | grep ssh-agent

user1 4729 4662 0 18:14 ? 00:00:00 /usr/bin/ssh-agent

client$ ssh-add -l

The agent has no identities.

client$ ssh-add

Enter passphrase for /home/<user1>/.ssh/id_rsa: <passphrase>

Identity added: ~/.ssh/id_rsa (~/.ssh/id_rsa)

Private Key liegt (entschlüsselt) Hauptspeicher des SSH-Agenten

Einloggen ohne Passphrase

SSH-Server: Datei ~/.ssh/authorized_keys nicht eingerichtet:

client$ ssh <server>

Password:

Last login: Tue Oct 20 11:22:22 2015

SSH-Server: Datei ~/.ssh/authorized_keys eingerichtet:

client$ ssh <server>

Last login: Tue Oct 20 12:11:44 2015 from <server>s

Grundlagen

Fingerprint überprüfen

Anwendungs-Beispiele

Man-in-the-Middle Attack

Einloggen ohne Passwort

Einloggen ohne Passphrase

SSH-Tunnel Anwendungen

SSH-Tunnel I

client$ ssh -L 1111:<host2>:25 <host2> Lokaler Port 1111

client$ # netstat -f inet -an | egrep -w '22|1111'

```
tcp    0  0    127.0.0.1:1111          0.0.0.0:*              LISTEN
tcp    0  0    192.168.2.100:58367  192.168.2.101:22   ESTABLISHED
```

Unverschlüsselte Daten der Applikation nutzen Tunnel

client$ telnet 127.0.0.1 1111

Sendmail 8.15.2 ...

SSH-Tunnel II

client$ ssh -L 127.0.0.1:1111:<host2>:2555 <host2>

Applikation "horcht":

server$ netcat -l -p 2555

Tunnelausgang sichtbar:

server$ netstat -an | egrep -w '22|2555'

```
tcp    0  0    0.0.0.0:2555          0.0.0.0:*                LISTEN
tcp    0  0    0.0.0.0:22            0.0.0.0:*                LISTEN
tcp    0  0    192.168.2.101:22 192.168.2.100:53141  ESTABLISHED
```

SSH-Tunnel III

SSH-Server als "Sprungbrett":

client$ ssh -L 127.0.0.1:1111:<host3>:25 <host2>

client$ ssh -L <host1>:1111:<host2>:25 <host2>

Verbindung zu einem durchgängigen Tunnel:

client$ ssh -L <host1>:1111:<host3>:25 <host2>

Reverse SSH-Tunnel (bei NAT)

Aufbau des Tunnel “von innen nach außen”

Rechner “außen”: TCP-Port als Tunnel-Eingang

server$ grep AllowTcpForwarding /etc/ssh/sshd_config

AllowTcpForwarding yes

client$ ssh -R 2222:127.0.0.1:22 <host2>

server$ netstat -an | egrep '22|2222'

```
tcp    0  0    0.0.0.0:2222          0.0.0.0:*                LISTEN
tcp    0  0    0.0.0.0:22            0.0.0.0:*                LISTEN
tcp    0  0    192.168.2.101:22      192.168.2.100:53024      ESTABLISHED
```

Reverse SSH-Tunnel (Nutzung)

Daten via Tunneleingang und Tunnel "zurück" an
sshd des Rechners, der den Tunnel aufgebaut hat

server$ ssh -p 2222 127.0.0.1
Password:
Last login: Fri Oct 2r34 12:15:44 2015 from <host1>
…
client$

Tunnel besteht, bis: ssh -R ...

Vielen Dank für Ihre Aufmerksamkeit!

FRAGEN ?

Uwe Grossu / Matthias Rumitz

AS-SYSTEME

DOAG 18.11.2015