13. Konfiguration von SSH und VPN

Secure Shell

Merkmale

- Verschlüsselte Login-Übertragung
- Verschlüsselte Datenübertragung
- Public-Key-Verfahren
- RSA-Schlüssel
- DSA-Schlüssel
- Nutzer-Schlüssel
- Host-Schlüssel

Protokolle

- SSH-Protokoll Version 1 (bei uns nicht)
- SSH-Protokoll Version 2

Vorteile

- Offene Quellen
- Kaum noch Sicherheitslöcher

Zentrale Konfigurationsfiles

in /etc/ssh oder /etc/openssh

```
moduli
ssh_config              - Konfigurationsfile für ssh-Client
ssh_host_dsa_key        - dsa-Host-Key geheim
ssh_host_dsa_key.pub    - dsa-Host-Key öffentlich
ssh_host_key            - Host-Key geheim
ssh_host_key.pub        - Host-Key öffentlich
ssh_host_rsa_key        - Host-Key geheim
ssh_host_rsa_key.pub    - Host-Key öffentlich
sshd_config             - Konfigurationsfile für sshd-Server
```

/etc/ssh/ssh_config

```
#       $OpenBSD: ssh_config,v 1.26 2010/01/11 01:39:46 dtucker Exp $

# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# ssh_config(5) man page.

Host *
ForwardAgent yes
ForwardX11 yes
ForwardX11Trusted yes
...

#   IdentityFile ~/.ssh/id_dsa
#   Port 22
    Protocol 2
...
```

/etc/ssh/sshd_config

```
#       $OpenBSD: sshd_config,v 1.82 2010/09/06 17:10:19 naddy Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.
...
Port 22
....
# The default requires explicit activation of protocol 1
Protocol 2
....
PermitRootLogin yes
UsePAM yes
.....
X11Forwarding yes
.....
```

```
Keys
====

/etc/ssh/ssh_host_key.pub
-------------------------
1024 35 121198783766069637887004787724215468355123156533511963754530433
978751839273854335683077972181150834238340086196592634128068149378840009
775120921552327858536448415038876795831503330263304796980633693757234324
432119374212756230546376003740235173897387526228609504329559210103598914
435958736158450228587495432529527 root@wll193-10

/etc/ssh/ssh_host_dsa_key.pub
-----------------------------
ssh-dss AAAAB3NzaC1kc3MAAACBAJGuUw/R9rzsmCh7dT4yjad0gjifhy4ax2mrLSgH7Tc
hSGJWrtDoj37YuwJP7eRIka51Y95ktLHN8s16ox5vdz+bEV/QO++e3DFbxJ9DR12J9V9tQW
To7NU1NopExsfTnSVOVbJ+3kZ1lTx1Xr2JAhX601alxDMtGFTERWXzd7g/AAAAFQDUX2ryi
HX1IWE10YpYtoiUnbhn9QAAAIBzQWYUwuWyvSfrqwGI5KW+fGn0ePYwBBzSbSkIOErCW9S0
mFNlABeMV2KLz0SF+Z5YrW5rvYDoLcDlvOUbY8ZnceRhH7jM/OOfuw+m39KHbpv4KAIpwgb
PzmR/yHafpt/XTo/Z5dSc6gTruVeV5kZMplZgC+1Irmuzg+/jTIDvlAAAAIALEAj/i9KM9o
o7pVw1/PEpCb3OetSUPBIs8yX7D5ehciZUqTy7VAfAkHoD5ZzGRjtj4mXWquxT5LHU0jVz/
4Q3IZ40x6h63CrZadczgGkgfHvoKNPlKm5wwjCeNrDr3svSH60P4Kn+zjB6QyzkPl/+s9Ec
Vn34OsQ+AO0QcqyYqg== root@wll193-10

/etc/ssh/ssh_host_rsa_key.pub
-----------------------------
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAw9Ma43YOE2wzboXDsCzgewp6vsp1APwHNg3
BZLiimDzErmx9dGwLqL3ekoG7Ea4UXL7EKhzbndQJCDqsrHon8DxyoFpSjiyowJ9MfiT9DL
0wRpJzin5kNado3bjwcgvcTxutdHQHS8Cs8BPpJN7CEzMGaW3YfREU2Zxl9XdIcW0= root@wll193-10
```

Private Konfigurationsfiles:

in ~/.ssh:

authorized_keys	**– Zugelassene Keys**
authorized_keys2	
known_hosts	**– bekannte Host–Keys**
known_hosts2	
id_dsa	**– privater DSA–Key**
id_dsa.pub	**– öffentlicher DSA–Key**
id_rsa	**– privater RSA–Key**
id_rsa.pub	**– öffentlicher RSA–Key**
identity	**– privater Key**
identity.pub	**– öffentlicher Key**

```
ssh – OpenSSH SSH client (remote login program)

     ssh [–l login_name] hostname | user@hostname [command]

     ssh [–afgknqstvxACNTVX1246] [–b bind_address] [–c cipher_spec]
         [–e escape_char] [–i identity_file] [–l login_name] [–m mac_spec]
         [–o option] [–p port] [–F configfile] [–L port:host:hostport]
         [–R port:host:hostport] [–D port] hostname | user@hostname [command]

       Beispiel:
         ssh tbell@rabe


scp – secure copy (remote file copy program)

     scp [–pqrvBC1246] [–F ssh_config] [–S program] [–P port] [–c cipher]
         [–i identity_file] [–l limit] [–o ssh_option] [[user@]host1:]file1
         [...] [[user@]host2:]file2

       Beispiel:
         scp –r * bell@data:tmp
```

```
sftp  – secure file transfer program

     sftp [–vC1] [–b batchfile] [–o ssh_option] [–s subsystem | sftp_server]
          [–B buffer_size] [–F ssh_config] [–P sftp_server path]
          [–R num_requests] [–S program] host
     sftp [[user@]host[:file [file]]]
     sftp [[user@]host[:dir[/]]]
     sftp –b batchfile [user@]host

       Beispiel:
         sftp bell@rabe

ssh–keygen – authentication key generation, management and conversion
     ssh–keygen [–q] [–b bits] –t type [–N new_passphrase] [–C comment]
                [–f output_keyfile]
     ssh–keygen –p [–P old_passphrase] [–N new_passphrase] [–f keyfile]
     ssh–keygen –i [–f input_keyfile]
     ssh–keygen –e [–f input_keyfile]
     ssh–keygen –y [–f input_keyfile]
     ssh–keygen –c [–P passphrase] [–C comment] [–f keyfile]
     ssh–keygen –l [–f input_keyfile]
     ssh–keygen –B [–f input_keyfile]
     ssh–keygen –D reader
     ssh–keygen –U reader [–f input_keyfile]
     ssh–keygen –r hostname [–f input_keyfile] [–g]
     ssh–keygen –G output_file [–b bits] [–M memory] [–S start_point]
     ssh–keygen –T output_file –f input_file [–a num_trials] [–W generator]
```

```
ssh-agent - authentication agent

     ssh-agent [-a bind_address] [-c | -s] [-t life] [-d] [command [args ...]]
     ssh-agent [-c | -s] -k

     Bespiel:
       ssh-agent -k    - beenden
       ssh-agent -s    - starten mit Shell-Ausgabe

ssh-add - adds RSA or DSA identities to the authentication agent

     ssh-add [-lLdDxXc] [-t life] [file ...]
     ssh-add -s reader
     ssh-add -e reader

     Bespiel:
          ssh-add

ssh-keyscan - gather ssh public keys

     ssh-keyscan [-v46] [-p port] [-T timeout] [-t type] [-f file]
                 [host | addrlist namelist] [...]

     Bespiel:
       ssh-keyscan -t rsa rabe
       ssh-keyscan -t dsa rabe
```

ssh-copy-id

Beispiel passwortloses sichere Einloggen mit SSH:

1. **Privaten Key erzeugen**
   ssh-keygen -b 4096 -t dsa
   ssh-keygen -b 4096 -t rsa

2. **Public-Key auf das Zielsystem übertragen**

3. **Zielsystem initialisieren**
   ssh <Quellsystem>
   es entsteht ~/.ssh/known_hosts

4. **Public-Key eintragen**
   in ~/.ssh/authorized_keys und ~/.ssh/authorized_keys2
   Linux Solaris

5. **Starten eines ssh-agent auf Quellsystem (bei gnome nicht notwendig)**
   ssh-agent -k
   ssh-agent -s
   Shellvariable merken und setzen

6. **Privaten Key auf dem Quellsystem an SSH-Agenten uebergeben**
   (bei gnome nicht notwendig)
   ssh-add

7. **ssh ohne Passworteingabe möglich**

   (wenn Umgebungsvariable für ssh-agent gesetzt oder gnome)

Virtuelle Private Netze – OpenVPN

Was ist VPN?

- ein Verfahren zur sicheren Kommunikation zwischen Rechnern an zwei oder mehreren Standorten.
- Alle beteiligten Rechner "bilden sich ein", in einem lokalen Rechnernetz zu sein.
- Die Kommunikation erfolgt verschlüsselt. Verschiedene kryptographische Verfahren werden unterstützt.
- Die Kommunikationspartner müssen sich authentifizieren.
- Nachfolger von direkten statischen WAN-Verbindungen, die sehr teuer waren und sind.

Anwendungsmöglichkeiten:
- Anbindung externer Clienten an ein Firmennetz
- Verbindung von unterschiedlichen Teilen eines Firmennetzes zu einem gemeinsamen Firmennetz
- Verbindung eines internen Firmennetzes mit dem Internet

Ziele:
- Vertraulichkeit
- Integrität
- Authentizität
- Verifizierbarkeit

Methoden
- symmetrische und asymmetrische Schlüssel
- digitale Signaturen
- Nutzerverifikation mittels Zertifikaten oder Nutzername/Passwort

Am Institut für Informatik werden mehrere OpenVPN-Server betrieben.
Für die Studenten steht der OpenVPN-Server

wlanin.informatik.hu-berlin.de 141.20.21.20

zur Verfügung.
Er ermöglicht den Aufbau von sicheren Verbindung zwischen Rechnern
im Internet und dem Netz der HUB.
Weiterhin ermöglicht er den Zugang aus dem offenen HU-WLAN-Netz "HU-VPN"
zum Netz der HUB. Dadurch ist die Nutzung von für die HU lizensierter
Software und Dienste aus dem Internet möglich.

Die Konfiguration der entsprechenden Cliente ist in

https://www2.informatik.hu-berlin.de/rbg/Openvpn_SSL/index.shtml

beschrieben.