Tutorium Fortgeschrittene

Netzwerk

Einleitung

Um sich sicher in einem Netzwerk zu bewegen, muss man es kennen.
Zu diesem Zwecke existieren unter Linux einige nützliche Tools, welche es ermöglichen, herauszufinden ob ein Computer im Netzwerk aktiv ist, welche offenen Ports er hat und wie man sich mit diesem verbindet.

Ein kleiner Teil der Möglichkeiten soll Inhalt dieses Tutoriums sein.

ping

ping

Ein ping ist ein kleines Programm, mit welchem man überprüfen kann, ob ein PC im Netzwerk aktiv ist. Zur Überprüfung wird ein ICMP-Paket gesendet und eine Antwort erwartet. Ist der PC aktiv, so wird ein Echo-Reply empfangen. Die Überprüfung kann allerdings fehlschlagen, sofern eine Firewall ICMP-Pakete blockt.

Befehl:
ping -Option Servername/IP-Adresse

z.B.: ping fh-schmalkalden.de
ping -a 192.168.0.1 # -a gibt bei Erfolg ein akustisches Signal aus

nmap

nmap (Network Mapper)

nmap ist ein Portscanner mit dessen Hilfe man Computer oder ganze Netzwerke auf offene Ports scannen kann. Er wird zur Diagnose von Schwachstellen im eigenen Netzwerk eingesetzt.

Befehl:
nmap -Optionen IP-Adresse(n)

z.B.: nmap 192.168.0.1
nmap -OsP 192.168.0.0/16 #aktive Rechner im Netzwerk finden

nmap

nmap (Network Mapper)

Optionen:

-sS	"SYN-Stealth-Scan": keine komplette TCP-Verbindung wird aufgebaut
-sU	Scannt UDP-Ports statt TCP
-sP	Ping-Scan: Prüft nur auf Erreichbarkeit (Ping)
-sV	Versucht den Dienst auf jedem offenen Port zu identifizieren
-O	Versucht das Betriebssystem des Zieles zu identifizieren
-A	Kurzform für -sV -O

nmap

nmap (Network Mapper)

Ports:

-p X	Scannt nur Port X
-p X-Y	Scannt nur Port X bis Y
-p X,Y,Z	Scannt die Ports X, Y und Z
-p-	Alle Ports scannen
-F	scannt nur die bekannten Ports, welche in der Datei **nmap-services** aufgeführt sind.
-r	Scannt alle Ports in numerischer Reihenfolge

nmap

nmap (Network Mapper)

weitere Optionen:

-P0	Überprüfung auf aktiven Port 80 deaktivieren und scannen
-e eth0	nmap nutzt die Netzwerkschnittstelle eth0
-oN DATEI	Protokolliert den Scan in DATEI
-v	gibt zusätzliche Informationen während des Scans aus

SSH

Einleitung

SSH dient zum Herstellen einer sicheren Verbindung zu einem enfernten Computer.
Die IANA (Internet Assigned Numbers Authority) hat dem SSH-Protokoll den TCP-Port 22 vergeben.

SSH steht dabei als Abkürzung für „SecureShell“ und ist eine Sammlung von Programmen.
Unter ihnen gibt es z.B. den SSH-Client und den SSH-Server.

SSH

Einleitung

Nachdem SSH unter einer kommerziellen Lizenz angeboten wurde, nahmen sich die Entwickler von OpenBSD des Quellcodes an und entwickelten eine freie Version namens „OpenSSH“.

Diese ist in fast allen Linux- / UNIX Distributionen verfügbar.

SSH

Installation

SSH-Client:
- ist bereits verfügbar und muss nicht nachinstalliert werden

SSH-Server:
apt-get install openssh-server

Konfiguration

SSH-Client:
~/.ssh/config (Einstellungen für jeden Benutzer)
/etc/ssh/ssh_config (Globale Einstellungen für alle Nutzer)

SSH-Server:
/etc/ssh/sshd_config

SSH

Verbindungsaufbau

ssh IP-Adresse
ssh Servername

z.B.: ssh 192.168.0.1
ssh ispost.informatik.fh-schmalkalden.de

Verbindung mit anderen Benutzernamen

ssh -l Benutzername Servername
ssh Benutzername@servername
ssh -o User=Benutzername Servername

z.B.: ssh -l kurt 192.168.0.1

SSH

Verbindung über einen anderen Port

ssh Servername -p Port

z.B.: ssh 192.168.0.1 -p 2233

Befehle automatisch ausführen

ssh Servername Befehl

z.B.: ssh Servername cat/etc/issue

SSH

Config-Datei-Client

```
Host Servername
HostName 192.168.0.1          #IP-Adresse des Servers
Port 2233                     #geänderter Port
User karl                     #Benutzername
Protocol 2                    #Protokollversion
ForwardAgent yes              #öffentliche Schlüssel weiterreichen
StrictHostkeyChecking ask     #Überprüfung der Schlüssel
ForwardX11 yes                #GUI weiterleiten
Compression yes               #Kompression einschalten
Cipher 3des                   #verwendete Verschlüsselung
CheckHostIP no
EscapeChar ~                  #Abbruch Zeichen
```

SSH

Auszug Config-Datei-Server

```
Port 2233                          #Standard Port ändern
PermitRootLogin no                 #Benutzer root verweigern
PermitEmptyPasswords no            #leere Passwörter ablehnen
X11Forwarding yes                  #GUI Weiterleiten zulassen
PasswordAuthentication yes         #Einloggen per Passwort zulassen
UsePAM yes                         #Einloggen per Passwort zulassen
PrintLastLog yes                   #letzten Login anzeigen
TCPKeepAlive yes                   #Verbindung aufrecht halten
```

SSH

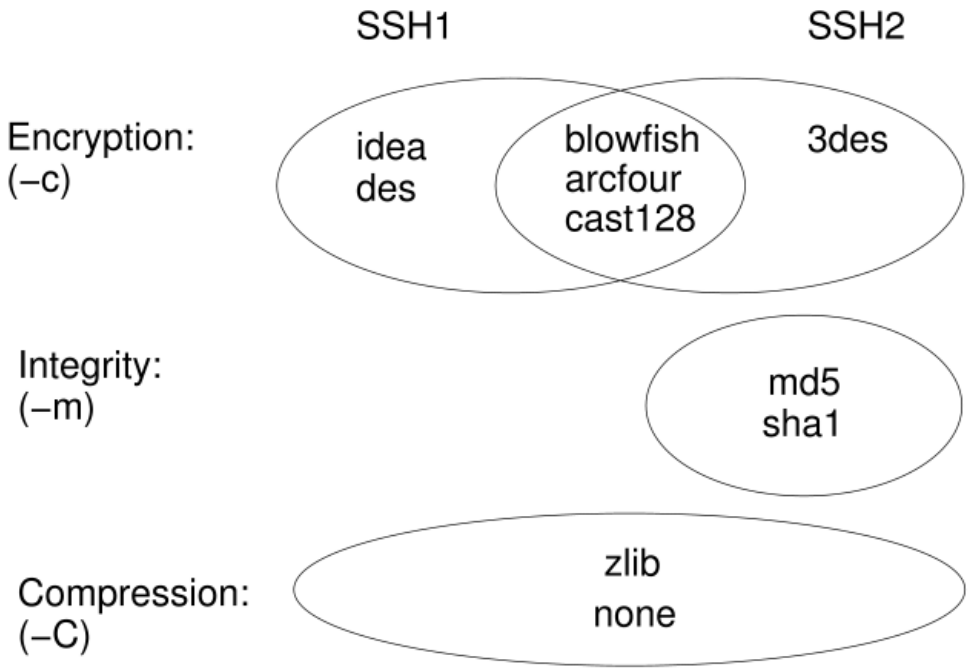
Sicherheit

Authentifizierung:
- Server: RSA-Zertifikat
- Client: Kennwort Authentifizierung (Standard Einstellung)
- Client: „Public-Key-Authentifizierung“ (öffentlicher Schlüssel wird auf dem Server hinterlegt)

Verschlüsselung:
- nach Authentifizierung -> Erzeugung eines geheimen Schlüssels; Gültigkeit: Dauer der Sitzung
- SSH2 Standard: AES 128-Bit-Schlüssellänge
- weitere mögliche Verfahren: 3DES, Blowfish, Twofish, CAST, IDEA, Arcfour, SEED, AES verschiedener Schlüssellängen (3DES: besonders sicher -> viel Rechenzeit; Blowfish: besonders schnell)

SSH

Sicherheit



SSH

Authentifizierung über Public-Keys

Schlüssel erzeugen:

ssh-keygen -t das

Hinweis: alle Fragen mit Enter bestätigen; Passwort vergeben!

Schlüssel in den Server einbinden:

ssh-copy-id -i ~/.ssh/id_dsa.pub Servername

SSH

Authentifizierung über Public-Keys

Schlüssel entfernen:

ssh-keygen -R Servername

Schlüssel für eine Sitzung speichern:

ssh-add

Hinweis: Speicherung nur bis zum Ausloggen aus dem Computer!

SSH

X-Forwarding

- grafische Programme anderer Computer werden auf eigenen Computer weitergeleitet (angezeigt)

- Voraussetzung: Paket „**xauth**“ auf Server installiert; „X11Forwarding **yes**“ in Config-Datei-Server

Aufruf:

ssh-X Servername

z.B.: ssh -X Servername firefox &

Hinweis: „ssh -Y Servername“ öffnet den Tunnel in beide Richtungen!

SSH

Dateitransfer

SCP "Secure Copy":

scp -Optionen woher:/Pfad/Datei wohin:/Pfad/Datei

z.B.: scp -C -P 223 karl@Servername:/home/ftp/Dateiname /home/karl/Dateiname
scp /home/karl/scripts/*.sh karl@Servername:/home/user/scripts

SSH

Dateitransfer

FUSE (Filesystem in USErspace) über SSH:

apt-get install sshfs (universe Quellen)
mkdir ~/mp3_server
adduser Benutzername fuse

Einbinden:

sshfs Benutzername@servername:/Pfad_auf_dem_Server/ /Pfad_auf_den_Client

z.B.: sshfs -C Servername:/home/user/mp3 ~/mp3_server

SSH

Dateitransfer

Ausbinden:

fusermount -u /Pfad_auf_den_Client

z.B.: fusermount -u ~/mp3_server

timeout verhindern:

sshfs -o ServerAliveInterval=15 Servername:/Pfad_auf_dem_Server/ /Pfad_auf_den_Client

Quellen

Quellen:

http://wiki.ubuntuusers.de/nmap
http://wiki.ubuntuusers.de/ssh
http://www.jfranken.de/homepages/johannes/vortraege/ssh1.de.html

Bildquellen:

http://openssh.org/images/openssh.gif
http://www.jfranken.de/homepages/johannes/vortraege/ssh/protokolle.png

Diskussion - Fragen

Bei konkreten Problemen sofort nachfragen

Linux-Stammtisch (Jeden 2. Mittwoch im Monat um 19.00 Uhr)

Dr. TUX die wöchtentliche Sprechstunde für Linux-Interessiert

Sprechzeiten: Mittwoch von 15:00 bis 17:00 Uhr

Ort: Database Competence Center
(Gebäude F, Raum F0001)

ubuntu