



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Labreport, Gruppe 4

Projekt Netzwerk-Infrastruktur WS 2017/18

vorgelegt von

Dewin Bagci: 5bagci@informatik.uni-hamburg.de

Karan Popat: karan.popat@outlook.de

Hanife Demircioglu: h.demircioglu@hotmail.de

MIN-Fakultät

Fachbereich Informatik

Abgabedatum: 01.03.2018

Dozent: Robert Olotu

Inhaltsverzeichnis

Abbildungsverzeichnis	3
Part 3: Network Troubleshooting Utilities	4
Exercise 6: Managing Services (Please use pnidX-svr-mu)	4
Exercise 7: Configure the following network (figure 1) using ifconfig and route add	13
Exercise 8: Configure the following network (figure 1) using ip and nmcli . .	13
Exercise 9: Configure the following network (figure 1) using GUI	13
Part 4: Network Scanning	14
Exercise 1: Configure the networks of figure 1	14
Exercise 2: NMAP	14
Exercise 3: Nessus network device identification	14
Exercise 4: OpenVAS Network device identification	14
Part 5: Sniffing, Virtual Private Network (VPN)	15
Exercise 1: Configure and set the networks shown below (figure1 and 2) . . .	15
Exercise 2: Getting started with network monitoring tools	15
Exercise 3: TCPDUMP	15
Exercise 4: Wireshark	15
Exercise 5: Experimenting with network monitoring tools	18
Exercise 6: Set up a host-to-host VPN using preshared key	19
Exercise 7: Set up a host-to-host VPN using RSA keys	19
Exercise 8: Set up a network-to-network VPN using preshared key	19
Exercise 9: Set up a network-to-network VPN using RSA secrets keys	19

Abbildungsverzeichnis

Abbildung 1: aktivierte bzw. deaktivierte Dienste eines runlevels	5
Abbildung 2: Runlevel, in denen iptables eingeschaltet bzw. ausgeschaltet sind	5
Abbildung 3:	6
Abbildung 4: Runlevel 2,3,4,5 werden deaktiviert	6
Abbildung 5: chkconfig iptables on off	6
Abbildung 6: vi /etc/yum.repos.d/local.repo	7
Abbildung 7: mounten	7
Abbildung 8: yum install tftp	7
Abbildung 9: yum install tftp	8
Abbildung 10: service xinetd start	8
Abbildung 11:	8
Abbildung 12:	9
Abbildung 13:	10
Abbildung 14: yum install vsftpd	10
Abbildung 15: Runlevel 2 von vsftpd wird deaktiviert	11
Abbildung 16:	11
Abbildung 17:	11
Abbildung 18:	11
Abbildung 19:	11
Abbildung 20:	12
Abbildung X1 Wireshark Filter 1	16

Part 3: Network Troubleshooting Utilities

Exercise 6: Managing Services (Please use `pnidX-svr-mu`)

Please type and explain the meaning of the following commands:

1) `# chkconfig`

Die folgenden Kommandos wurden auf Rechner `pnid4-svr-mu` mit dem Betriebssystem `Centos-6.5-x86_64` ausgeführt.

Zeigt an welche Dienste in ihren jeweiligen runlevels aktiviert bzw. deaktiviert sind [siehe Abb. 1]

```
[root@localhost ~]# chkconfig
NetworkManager 0:off 1:off 2:on 3:on 4:on 5:on 6:off
abrt-ccpp       0:off 1:off 2:off 3:on 4:off 5:on 6:off
abrt            0:off 1:off 2:off 3:on 4:off 5:on 6:off
acpid           0:off 1:off 2:on 3:on 4:on 5:on 6:off
atd             0:off 1:off 2:off 3:on 4:on 5:on 6:off
auditd          0:off 1:off 2:on 3:on 4:on 5:on 6:off
autofs          0:off 1:off 2:off 3:on 4:on 5:on 6:off
blk-availability 0:off 1:off 2:on 3:on 4:on 5:on 6:off
certmonger      0:off 1:off 2:off 3:on 4:on 5:on 6:off
cpuspeed        0:off 1:on 2:on 3:on 4:on 5:on 6:off
crond           0:off 1:off 2:on 3:on 4:on 5:on 6:off
cups            0:off 1:off 2:on 3:on 4:on 5:on 6:off
dnsmasq         0:off 1:off 2:off 3:off 4:off 5:off 6:off
firstboot       0:off 1:off 2:off 3:off 4:off 5:off 6:off
haldaemon       0:off 1:off 2:off 3:on 4:on 5:on 6:off
ip6tables       0:off 1:off 2:on 3:on 4:on 5:on 6:off
iptables        0:off 1:off 2:on 3:on 4:on 5:on 6:off
irqbalance      0:off 1:off 2:off 3:on 4:on 5:on 6:off
kdump           0:off 1:off 2:on 3:on 4:on 5:on 6:off
lvm2-monitor    0:off 1:on 2:on 3:on 4:on 5:on 6:off
mdmonitor       0:off 1:off 2:on 3:on 4:on 5:on 6:off
messagebus      0:off 1:off 2:on 3:on 4:on 5:on 6:off
netconsole      0:off 1:off 2:off 3:off 4:off 5:off 6:off
netfs           0:off 1:off 2:off 3:on 4:on 5:on 6:off
network         0:off 1:off 2:on 3:on 4:on 5:on 6:off
nfs             0:off 1:off 2:off 3:off 4:off 5:off 6:off
nfslock         0:off 1:off 2:off 3:on 4:on 5:on 6:off
ntpd            0:off 1:off 2:off 3:off 4:off 5:off 6:off
ntpddate        0:off 1:off 2:off 3:off 4:off 5:off 6:off
oddjobd         0:off 1:off 2:off 3:off 4:off 5:off 6:off
portreserve     0:off 1:off 2:on 3:on 4:on 5:on 6:off
postfix         0:off 1:off 2:on 3:on 4:on 5:on 6:off
psacct          0:off 1:off 2:off 3:off 4:off 5:off 6:off
quota_nld       0:off 1:off 2:off 3:off 4:off 5:off 6:off
rdisc           0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

Abbildung 1: aktivierte bzw. deaktivierte Dienste eines runlevels

2)# chkconfig -- list iptables

Zeigt an in welchen runlevel iptables eingeschaltet bzw. ausgeschaltet ist. [Abb. 2]

```
[root@localhost ~]# chkconfig --list iptables
iptables      0:off 1:off 2:on 3:on 4:on 5:on 6:off
[root@localhost ~]#
```

Abbildung 2: Runlevels, in denen iptables eingeschaltet bzw. ausgeschaltet sind

3)# chkconfig --level 2 iptables off

Deaktiviert iptables im runlevel 2. [Abb. 3]. Wir sehen, dass zuvor iptables im runlevel 2 aktiviert war.

```
[root@localhost ~]# chkconfig --list iptables
iptables      0:off  1:off  2:on   3:on   4:on   5:on   6:off
[root@localhost ~]# chkconfig --level 2 iptables off
[root@localhost ~]# chkconfig --list iptables
iptables      0:off  1:off  2:off  3:on   4:on   5:on   6:off
```

Abbildung 3: runlevel 2 wird ausgeschaltet

4)# chkconfig --level 2345 iptables off

Deaktiviert iptables im runlevel 2, 3, 4 und 5. [Abb. 4]

```
[root@localhost ~]# chkconfig --level 2 iptables off
[root@localhost ~]# chkconfig --list iptables
iptables      0:off  1:off  2:off  3:on   4:on   5:on   6:off
[root@localhost ~]# chkconfig --level 2345 iptables off
[root@localhost ~]# chkconfig --list iptables
iptables      0:off  1:off  2:off  3:off  4:off  5:off  6:off
[root@localhost ~]
```

Abbildung 4: Runlevel 2,3,4,5 werden deaktiviert

5)# chkconfig iptables on | off

Mit iptables off wird iptables auf jedem runlevel deaktiviert. Mit iptables on wird iptables auf die default Konfiguration zurückgesetzt. Das bedeutet die runlevels 2,3,4 und 5 sind wieder aktiviert.

```
[root@localhost ~]# chkconfig --list iptables
iptables      0:off  1:off  2:off  3:off  4:off  5:off  6:off
[root@localhost ~]# chkconfig iptables on
[root@localhost ~]# chkconfig --list iptables
iptables      0:off  1:off  2:on   3:on   4:on   5:on   6:off
[root@localhost ~]# chkconfig iptables off
[root@localhost ~]# chkconfig --list iptables
iptables      0:off  1:off  2:off  3:off  4:off  5:off  6:off
[root@localhost ~]#
```

Abbildung 5: chkconfig iptables on | off

6)# chkconfig tftp on

Tftp ist ein Vorgänger des FTP-Protokolls. Dieser service ist nicht automatisch auf Centos-6.5-x86_64 vorinstalliert und wird durch den Superserver xinetd, welcher ebenfalls nicht automatisch vorinstalliert ist, verwaltet. Damit wir die gewissen Pakete mit

allen Abhängigkeiten für tftp und xinetd über das Terminal mit yum (Yellow dog Updater, Modified) installieren können, müssen wir ein Quellpaket Repository einrichten. Zuerst erstellen wir einen Ordner mit `# mkdir /dvdrom` im Verzeichnis `/etc/yum.repos.d`. Danach fügen wir das Verzeichnis als neues Repository hinzu, indem wir die Konfigurationsdatei mit dem vi Editor öffnen `# vi /etc/yum.repos.d/local.repo` und das Repository hinzufügen. [Abb. 6]

```
[LocalRepo]
name=Local Repository
baseurl=file:///dvdrom
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

Abbildung 6: vi /etc/yum.repos.d/local.repo

Zuletzt mounten wir das Verzeichnis mit dem Befehl `# mount -t iso9660 /dev/sr0 /dvdrom`

```
[root@localhost yum.repos.d]# mount -t iso9660 /dev/sr0 /dvdrom
mount: block device /dev/sr0 is write-protected, mounting read-only
```

Abbildung 7: mounten

Nach dem wir den Befehl `#yum clean all` im Terminal ausgeführt haben, kann die Installation beginnen. Dies geschieht wie folgt:

Wir führen im Terminal den Befehl `#yum install tftp` aus, sodass die Installation starten kann.

```
[root@localhost yum.repos.d]# yum install tftp
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
LocalRepo
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package tftp.x86_64 0:0.49-7.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch          Version           Repository        Size
=====
Installing:
tftp              x86_64        0.49-7.el6        LocalRepo         32 k
=====
Transaction Summary
-----
Install      1 Package(s)
-----
```

Abbildung 8: yum install tftp

Nachdem tftp installiert wurde, muss außerdem xinetd installiert werden. Ansonsten kann tftp nicht verwendet werden. Mit dem Befehl `#yum install xinetd` wird xinetd installiert.

```
[root@localhost ~]# yum install xinetd
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package xinetd.x86_64 2:2.3.14-39.el6_4 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version           Repository        Size
=====
Installing:
xinetd                  x86_64        2:2.3.14-39.el6_4 LocalRepo         121 k
Transaction Summary
-----
Install      1 Package(s)

Total download size: 121 k
Installed size: 259 k
Is this ok [y/N]: y
Downloading Packages:
Running rpm check debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 2:xinetd-2.3.14-39.el6_4.x86_64
    Verifying : 2:xinetd-2.3.14-39.el6_4.x86_64
    Installed:
      xinetd.x86_64 2:2.3.14-39.el6_4
Complete!
```

Abbildung 9: yum install xinetd

Zunächst muss xinetd gestartet werden, damit wir Zugriff auf tftp haben. Dies geschieht mit dem Befehl `# service xinetd start`.

```
[root@localhost ~]# service xinetd start
Starting xinetd: [ OK ]
```

Abbildung 10: service xinetd start

Die Dateien im Verzeichnis `/etc/xinetd.d/` enthalten die Konfigurationsdateien für jeden von xinetd verwalteten Dienst. Die Konfigurationsdatei tftp muss wie in Abbildung 15 angepasst werden. Damit tftp funktioniert, muss `disable=no` sein. Disable legt fest, ob der Dienst aktiv ist oder nicht. Im Regelfall ist "disable = yes" zu Beginn. Dieser muss dann geändert werden zu "disable = no". Nach der Konfiguration kann tftp genutzt werden, wie in Abbildung 12 zu sehen ist.

```
[root@localhost ~]# vi /etc/xinetd.d/tftp
[root@localhost ~]# service xinetd start
Starting xinetd:
[root@localhost ~]# chkconfig tftp on
[root@localhost ~]# chkconfig
```

Abbildung 11


```
service tftp
{
    disable = no
    socket_type = dgram
    protocol = udp
    wait = yes
    user = root
    server = /usr/sbin/in.tftpd
    server_args = -s /var/lib/tftboot
    per_source = 11
    cps = 100 2
    flags = IPv4
}
```

Abbildung 12

Nachdem der Befehl `# chkconfig tftp on` ausgeführt wurde, kann man sich mit dem Befehl `#chkconfig` anzeigen lassen, ob der Dienst wirklich aktiviert wurde, da dieser angibt welche Dienste in ihren jeweiligen runlevels aktiviert bzw. deaktiviert sind. In der Abbildung 13 sieht man, dass tftp aktiviert ist. Tftp findet man unten im Bild bei den "xinetd based services".

```

smartd      0:off  1:off  2:off  3:off  4:off  5:off  6:off
snmpd       0:off  1:off  2:off  3:off  4:off  5:off  6:off
snmptrapd   0:off  1:off  2:off  3:off  4:off  5:off  6:off
spice-vdagentd 0:off  1:off  2:off  3:off  4:off  5:on   6:off
sshd        0:off  1:off  2:on   3:on   4:on   5:on   6:off
sssd        0:off  1:off  2:off  3:off  4:off  5:off  6:off
sysstat     0:off  1:on   2:on   3:on   4:on   5:on   6:off
udev-post   0:off  1:on   2:on   3:on   4:on   5:on   6:off
wdaemon     0:off  1:off  2:off  3:off  4:off  5:off  6:off
winbind     0:off  1:off  2:off  3:off  4:off  5:off  6:off
wpa_supplicant 0:off  1:off  2:off  3:off  4:off  5:off  6:off
xinetd      0:off  1:off  2:off  3:on   4:on   5:on   6:off
ypbind      0:off  1:off  2:off  3:off  4:off  5:off  6:off

xinetd based services:
  chargen-dgram: off
  chargen-stream: off
  daytime-dgram: off
  daytime-stream: off
  discard-dgram: off
  discard-stream: off
  echo-dgram: off
  echo-stream: off
  rsync: off
  tcpmux-server: off
  tftp: on
  time-dgram: off
  time-stream: off

```

Abbildung 13

7)# chkconfig --level 2 vsftpd off

Um diesen Befehl ausführen zu können, muss zunächst vsftpd installiert werden. Dies geschieht mit dem Befehl # yum install vsftpd. Nach der erfolgreichen Installation kann vsftpd verwendet werden.

```

[root@localhost ~]# yum install vsftpd
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
Setting up Install Process
Resolving Dependencies
--> Running Transaction check
--> Package vsftpd.x86_64 0:2.2.2-11.el6_4.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version           Repository        Size
=====
Installing:
vsftpd                 x86_64        2.2.2-11.el6_4.1  LocalRepo        151 k
Transaction Summary
-----
Install      1 Package(s)

Total download size: 151 k
Installed size: 331 k
Is this ok [y/N]: y
Downloading Packages:
Running rpm check debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : vsftpd-2.2.2-11.el6_4.1.x86_64      1/1
  Verifying  : vsftpd-2.2.2-11.el6_4.1.x86_64      1/1

Installed:
vsftpd.x86_64 0:2.2.2-11.el6_4.1

Complete!

```

Abbildung 14: yum install vsftpd

Mit dem Befehl `#chkconfig --level 2 vsftpd off` wird der Runlevel 2 von vsftpd deaktiviert.

```
[root@localhost ~]# chkconfig --level 2 vsftpd off
[root@localhost ~]# chkconfig
```

Abbildung 15: Runlevel 2 von vsftpd wird deaktiviert

8) `# chkconfig --level 2345 vsftpd off`

Mit dem Befehl `# chkconfig --level 2345 vsftpd off` werden die Runlevels 2, 3, 4 und 5 deaktiviert.

Zunächst haben wir mit dem Befehl `"# chkconfig --level 2345 vsftpd on"` die Runlevels 2, 3, 4 und 5 aktiviert, wie in Abbildung 16 zu sehen ist.

```
[root@localhost ~]# chkconfig --level 2345 vsftpd on
```

Abbildung 16

Hier sieht man, dass nach der Aktivierung die entsprechenden Runlevels aktiviert wurden.

```
|vsftpd      0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

Abbildung 17

Anschließend werden mit dem Befehl `"# chkconfig --level 2345 vsftpd off"` die Runlevels 2, 3, 4 und 5 deaktiviert.

```
[root@localhost ip nmcli]# chkconfig --level 2345 vsftpd off
```

Abbildung 18

Man erkennt, dass die aktivierten Runlevels nach dem Ausführen des Befehls ausgeschaltet wurden.

```
|vsftpd      0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

Abbildung 19

9)# Explain the function of xinetd

Bei xinetd handelt es sich um einen open source Superserver für Unix-Systeme. Dieser verwaltet verschiedene Dienste u.a. den FTP / HTTP Server.

Xinetd bietet gegenüber dem Vorgänger inetd noch weitere zusätzliche Dienste an um eine verbesserte Sicherheit zu ermöglichen. Dazu zählen Zugangskontrollen, zeitliche Beschränkung von Diensten (nach Datum und Uhrzeit), sowie einen Verteidigungsmechanismus gegen Portscanner. Sobald der xinetd Superserver eingeschaltet ist, lässt sich im Terminal nachvollziehen, welche Dienste über xinetd verwaltet werden.

The super server xinetd controlled services are automatically enabled or disabled by chkconfig.

Please type and explain the meaning of the following commands:

10)# service network stop

Der command stoppt alle konfigurierten Netzwerk interfaces. 11)# service network start

Der command aktiviert alle konfigurierten Netzwerk interfaces.

```
[root@localhost ~]# service network stop
Shutting down interface eth0:                [ OK ]
Shutting down loopback interface:             [ OK ]
[root@localhost ~]# service network start
Bringing up loopback interface:               [ OK ]
[root@localhost ~]#
```

Abbildung 20

Exercise 7: Configure the following network (figure 1) using ifconfig and route add

Exercise 8: Configure the following network (figure 1) using ip and nmcli

Exercise 9: Configure the following network (figure 1) using GUI

Part 4: Network Scanning

Exercise 1: Configure the networks of figure 1

Exercise 2: NMAP

Exercise 3: Nessus network device identification

Exercise 4: OpenVAS Network device
identification

Part 5: Sniffing, Virtual Private Network (VPN)

Exercise 1: Configure and set the networks shown below (figure1 and 2)

Exercise 2: Getting started with network monitoring tools

Exercise 3: TCPDUMP

Exercise 4: Wireshark

Question 1: Please type and examine the syntax for a Wireshark command which capture filter so that all IP datagrams with source or destination IP address equal to 10.88.X.? are recorded.

Answer 1: Mit dem Filter: *ip.addr == 10.88.40.70* können wir alle Netzwerkpakete, welche über die Schnittstelle 10.88.40.70 gesendet oer empfangen werden, abfangen und anzeigen lassen.

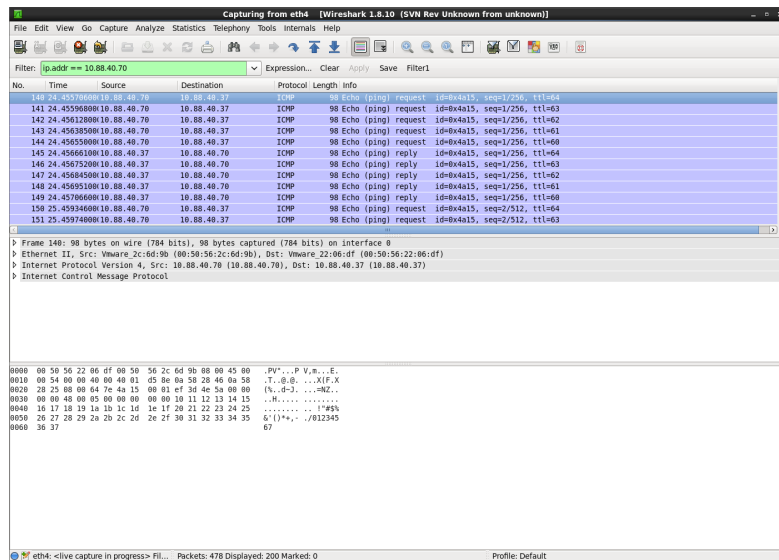


Abbildung 21: Wireshark Filter ip.addr == 10.88.40.70

Question 2: Please type and examine the syntax for a Wireshark display filter that shows IP datagrams with destination IP address equal to 10.88.X.? and frame size greater than 400 bytes.

Answer 2: Um alle Datenpakete abzufangen, die mindestens 400 Byte groß sind, bedarf eine kleine Erweiterung des vorherigen Befehls. Der Filter lautet nun: `ip.addr == 10.88.40.70 && frame.len > 400`. Mit dem Teil `frame.len > X` können wir die Datenpakete nach Bytegröße X filtern. Für X gilt, $X < 2^{32} \wedge X \in \mathbb{N}$.

Question 3: Please type and examine the syntax for a Wireshark display filter that shows packets containing ICMP messages with source or destination IP address equal to 10.88.X.? and frame numbers between 15 and 30

Answer 3: Der Filter lautet: `ip.addr == 10.88.40.70 && (frame.number > 15 && frame.number < 30)`. ICMP steht für Internet Control Message Protocol und übermittelt hauptsächlich Diagnose-informationen zwischen dem Router und dem Host.

Question 4: Please type and examine the syntax for a Wireshark display filter that

shows packets containing TCP segments with source or destination IP address equal to 10.88.X.? and using port number 23.

Answer 4: Damit wir alle TCP Pakete eines Hosts über die Port 23 abfangen können wird der folgende Filter eingesetzt: *ip.dst == 10.88.40.70 and tcp.port == 23*. Bei TCP handelt es sich um ein Übertragungsprotokoll (Transmission Control Protocol) aus der Familie der Internetprotokolle. Port 23 ist standardisiert für den Service Telnet.

Question 5: Please type and examine a Wireshark capture filter expression for Q4.

Answer 5: Der Filter ist ähnlich wie in Q4, lediglich die Konfiguration findet an einer anderen Stelle statt.

Question 6: Please type and examine the syntax for a Wireshark command which, by default, collects packets with source or destination IP address 10.88.X.? on interface eth4.

Answer 6: Innerhalb des Terminals lässt sich der Filter: *wireshark -i eth4 -k -f "host 10.88.40.70"*, anwenden. Die Argumente bedeuten dabei folgendes: *-i eth4* steht für Interface, *-k* startet das Abfangen von Paketen und *-f "host 10.88.40.70"*, ist der Paketfilter.

Question 7: : Please type and examine the syntax of a display filter which selects the TCP packets with destination IP address 10.88.X.?, and TCP port number 23.

Answer 7: Der Filter lautet: *ip.addr == 10.88.40.70 && tcp.port == 23* und fängt alle ein-/ausgehenden Pakete der Ip Adresse 10.88.40.70 über den Port 23 (Telnet) ab.

Question 8: Please login to the server pnidX-mid-hh and start an ftp client to the server pnidXcnt-bln(vsftpd daemon should be running on pnidX-cnt-bln). Please use wireshark on pnidX-mid-bln to sniff or capture the username and password of the ftp service between pnidX-mid-hh and pnidX-cnt-bln. Is this possible, show your result of the capture

Answer 8: Mithilfe von Wireshark können wir leicht das Telnet login Passwort herausfinden, da bei der Übertragung via Telnet die Pakete unverschlüsselt übertragen werden. Dazu starten wir zunächst Wireshark auf dem Host pnid4-mid-hh.

Question 9: Please login to the server pnidX-mid-hh and start an ssh client to the server pnidX-cnt-bln(sshd daemon should be running on pnidX-cnt-bln). Please use wireshark on pnidX-mid-bln to sniff or capture the username and password of the ssh service between pnidX-mid-hh and pnidX-cnt-bln. Is this possible, show the result of the capture.

Answer 9: Anders als Telnet werden bei ssh (Secure Shell) die Pakete verschlüsselt übertragen, sodass es nicht möglich ist das Passwort mitzulesen. Zuerst starten wir wireshark auf dem Host pnid4-mid-hh.

Exercise 5: Experimenting with network monitoring tools

Exercise: In this exercise you will connect to the webserver pnidX-cnt-bln from pnidX-mid-hh. Let pnidX-cnt-bln determine your IP-address and the OS you are running. Then, connect to a service of your choice (e.g. ftp, http, ssh etc.) on pnidX cnt-bln. Let pnidX-mid-hh determine which services are running on pnidX-cnt-bln.

Solution: Wir führen zunächst nmap auf dem Host pnid4-cnt-bln aus und übergeben dabei die Zieladresse des Hosts pnid4-mid-hh, damit wir sehen können welche Ports geöffnet sind bzw. welcher Service auf dem Zielhost gerade aktiv ist.

[IMAGE]

Wir sehen nun, dass die Ports 21 ftp, 22 ssh, 23 telnet, 80 http und 111 rpcbind offen sind und entscheiden uns via Telnet vom Quellhost pnid4-cnt-bln bei dem Zielhost

pnid-mid-hh anzumelden.

[IMAGE]

Die Ausgabe des letzten Logins zeigt uns, dass die Anmeldung erfolgreich war.

**Exercise 6: Set up a host-to-host VPN using
preshared key**

**Exercise 7: Set up a host-to-host VPN using RSA
keys**

**Exercise 8: Set up a network-to-network VPN
using preshared key**

**Exercise 9: Set up a network-to-network VPN
using RSA secrets keys**