



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 | 5 |
| Exercise 6: Managing Services (Please use pnidX-svr-mu) | 5 |
| 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 . . | 24 |
| Exercise 9: Configure the following network (figure 1) using GUI | 24 |
| Part 4: Network Scanning | 26 |
| Exercise 1: Configure the networks of figure 1 | 27 |
| Exercise 2: NMAP | 30 |
| Exercise 3: Nessus network device identification | 31 |
| Exercise 4: OpenVAS Network device identification | 31 |
| Part 5: Sniffing, Virtual Private Network (VPN) | 32 |
| Exercise 1: Configure and set the networks shown below (figure1 and 2) . . . | 32 |
| Exercise 2: Getting started with network monitoring tools | 32 |
| Exercise 3: TCPDUMP | 32 |
| Exercise 4: Wireshark | 32 |
| Exercise 5: Experimenting with network monitoring tools | 39 |
| Exercise 6: Set up a host-to-host VPN using preshared key | 40 |
| Exercise 7: Set up a host-to-host VPN using RSA keys | 41 |
| Exercise 8: Set up a network-to-network VPN using preshared key | 41 |
| Exercise 9: Set up a network-to-network VPN using RSA secrets keys | 41 |

Abbildungsverzeichnis

| | |
|--|----|
| Abbildung 1: aktivierte bzw. deaktivierte Dienste eines runlevels | 6 |
| Abbildung 2: Runlevel, in denen iptables eingeschaltet bzw. ausgeschaltet sind | 6 |
| Abbildung 3: | 6 |
| Abbildung 4: Runlevel 2,3,4,5 werden deaktiviert | 7 |
| Abbildung 5: chkconfig iptables on off | 7 |
| Abbildung 6: vi /etc/yum.repos.d/local.repo | 8 |
| Abbildung 7: mounten | 8 |
| Abbildung 8: yum install tftp | 8 |
| Abbildung 9: yum install tftp | 9 |
| Abbildung 10: service xinetd start | 9 |
| Abbildung 11: | 9 |
| Abbildung 12: | 10 |
| Abbildung 13: | 11 |
| Abbildung 14: yum install vsftpd | 11 |
| Abbildung 15: Runlevel 2 von vsftpd wird deaktiviert | 12 |
| Abbildung 16: | 12 |
| Abbildung 17: | 12 |
| Abbildung 18: | 12 |
| Abbildung 19: | 12 |
| Abbildung 20: | 13 |
| Abbildung 21: | 15 |
| Abbildung 22: | 16 |
| Abbildung 23: | 16 |
| Abbildung 24: | 17 |
| Abbildung 25: | 17 |
| Abbildung 26: | 18 |

| | |
|---|----|
| Abbildung 27: | 18 |
| Abbildung 28: | 18 |
| Abbildung 29: | 19 |
| Abbildung 30: | 19 |
| Abbildung 31: | 20 |
| Abbildung 32: | 20 |
| Abbildung 33: | 20 |
| Abbildung 34: | 21 |
| Abbildung 35: | 21 |
| Abbildung 36: Netz: 10.88.40.32/27 | 22 |
| Abbildung 37: Netz: 10.88.40.64/27 | 23 |
| Abbildung 38: Netz: 10.88.40.96/27 | 23 |
| Abbildung 39: Netz: 10.88.40.128/27 | 24 |
| Abbildung P4 figure 1 LAN | 26 |
| Abbildung P5 ex. 1 Zenmap Subnetz 64 | 27 |
| Abbildung P5 ex. 1 Zenmap Subnetz 96 | 28 |
| Abbildung P5 ex. 1 Zenmap Subnetz 128 | 28 |
| Abbildung P5 ex. 1 Zenmap Subnetz 160 | 29 |
| Abbildung P5 ex. 1 Zenmap Subnetz 32 | 29 |
| Abbildung P5 ex. 1 Zenmap alle Subnetze | 30 |
| Abbildung P5 ex. 4 Wireshark Filter 1 | 30 |
| Abbildung P5 ex. 4 Wireshark Filter 1 | 33 |
| Abbildung P5 ex. 4 Wireshark Filter 2 | 33 |
| Abbildung P5 ex. 4 Wireshark Filter 3 | 34 |
| Abbildung P5 ex. 4 Wireshark Filter 4 | 34 |
| Abbildung P5 ex. 4 Wireshark Filter 5 | 35 |
| Abbildung P5 ex. 4 Wireshark Filter 7 | 36 |
| Abbildung P5 ex. 4 Wireshark Filter 8 | 37 |
| Abbildung P5 ex. 4 Wireshark ftp Login Passwort | 37 |
| Abbildung P5 ex. 4 Wireshark Filter 9 | 38 |
| Abbildung P5 ex. 4 Wireshark ssh Datenpaket | 39 |
| Abbildung P5 ex. 5 nmap offene Ports anzeigen | 39 |
| Abbildung P5 ex. 5 Telnet login | 40 |

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
- 2)# chkconfig --list iptables
- 3)# chkconfig --level 2 iptables off
- 4)# chkconfig --level 2345 iptables off
- 5)# chkconfig iptables on | off
- 6)# chkconfig tftp on
- 7)# chkconfig --level 2 vsftpd off
- 8)# chkconfig --level 2345 vsftpd off
- 9)# Explain the function of xinetd

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
- 11)# service network start

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-d         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
mdmonitord      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
ntpdupdate      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 | 4.0 kB 00:00 ...
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      1/1
  Verifying  : 2:xinetd-2.3.14-39.el6_4.x86_64      1/1
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
  Verifying  : vsftpd-2.2.2-11.el6_4.1.x86_64
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

You need to set the network depicted on figure 1 by doing the following:

Use the "ifconfig" and the "route add" commands to configure all the subnets 10.88.X.32/27, 10.88.X.64/27, 10.88.X.96/27 and 10.88.X.128/27. For this exercise you will use the hosts pnidX-svr-mu, pnidX-WEB-hn, pnidX-svr-blh and pnidX-svr-hh. Furthermore you have to configure the routers pnidX-rou-1, pnidX-rou-2 and pnidX-rou-3

Hint 1: Remember after rebooting the system, the ifconfig and route add configuration

will disappear

Hint 2: Do not forget to flush the firewall by issuing the command "iptables -F"

Please use Kali Linux as root:

```
# zenmap
```

Scan the networks:

10.88.X.32/27

10.88.X.64/27

10.88.X.96/27

and 10.88.X.128/27

Ziel der Aufgaben 7, 8 und 9 ist, das folgende Netzwerk [Abb. 21] mithilfe verschiedener Kommandos aufzubauen um alle zugehörigen Subnetze zu konfigurieren.

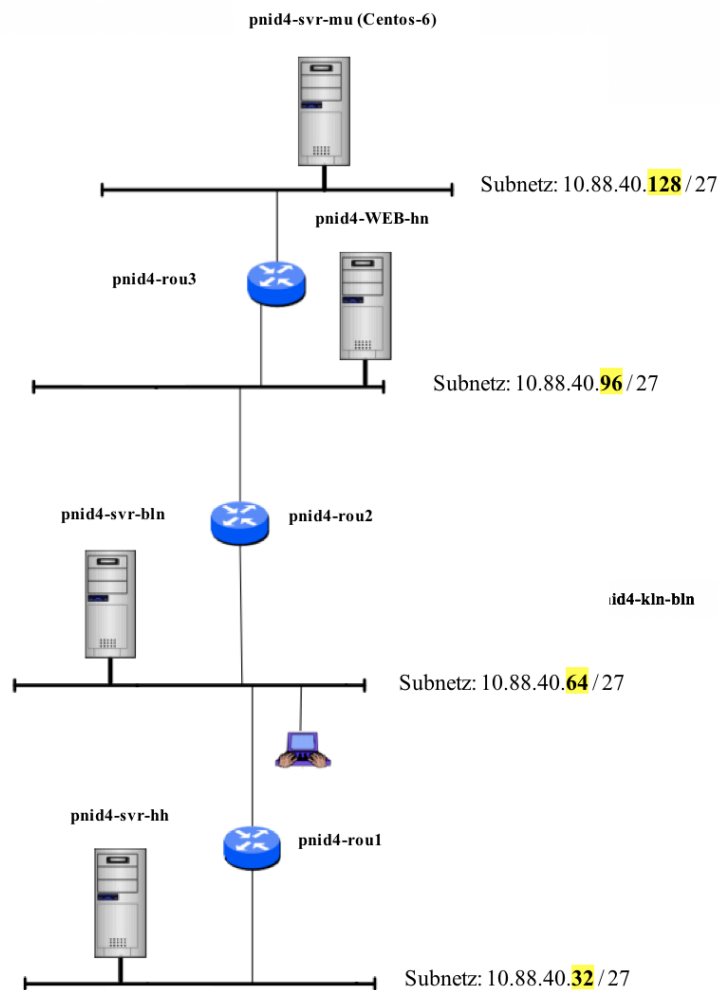


Abbildung 21

Zunächst haben wir das Netzwerk für den pnid4-svr-mu (Sever München) konfiguriert, indem wir die Ethernetkarte eth0 zustehende Adresse 10.88.40.129 zugewiesen haben und die Netmask-Adresse mit einbinden. Dieses haben wir ermöglicht, indem wir /27 im Anschluss hinzugefügt haben, jedoch ist es auch über dem Schlüsselwort netmask und die komplette Adresse realisierbar.

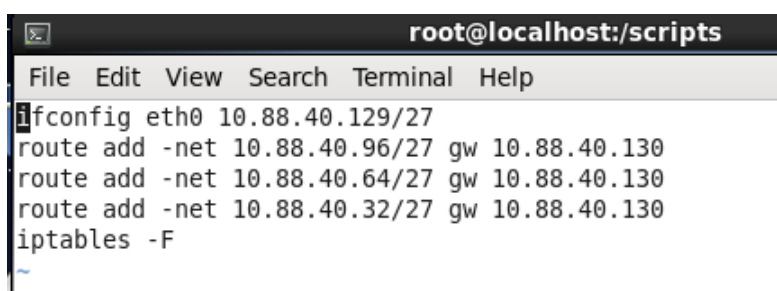
Anschließend haben wir die Routing tables über dem Kommando route add erstellt. Hier haben wir die Netze 10.88.40.96, 10.88.40.64 und 10.88.40.32 mit dem Kommando versehen, da wir über alle drei Netze eine Verbindung herstellen möchten.

Nachstehend haben wir die Firewall dieses Servers ausgeschaltet, da wir später eine Verbindung mit anderen Servern und Routern aufbauen möchten. Wir haben die ganze Konfiguration in einer txt-Datei geschrieben und im Anschluss einmal ausgeführt, sodass wir die Konfiguration gespeichert haben und nicht bei jedem Shut-Down erneut einstellen müssen.

Konfigurationsdatei von Server-München:(Routing-Tabelle)

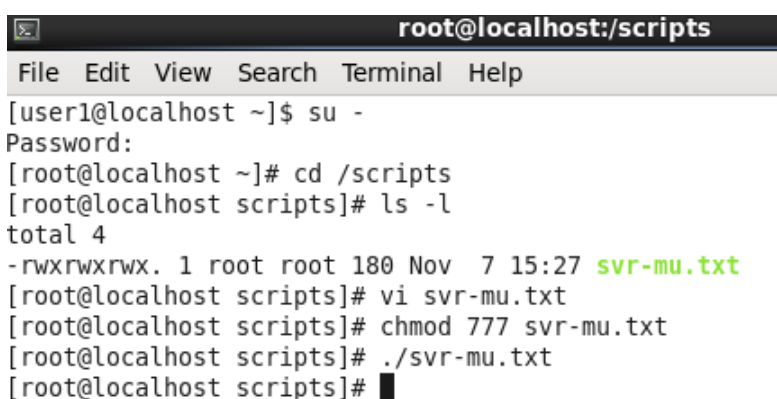
ifconfig eth0 weist der Netzwerkkarte die zugehörige IP-Adresse zu.

Mit dem Befehl route add werden statische Routen zu Rechnern und Netzwerken festgelegt.



```
root@localhost:/scripts
File Edit View Search Terminal Help
ifconfig eth0 10.88.40.129/27
route add -net 10.88.40.96/27 gw 10.88.40.130
route add -net 10.88.40.64/27 gw 10.88.40.130
route add -net 10.88.40.32/27 gw 10.88.40.130
iptables -F
```

Abbildung 22



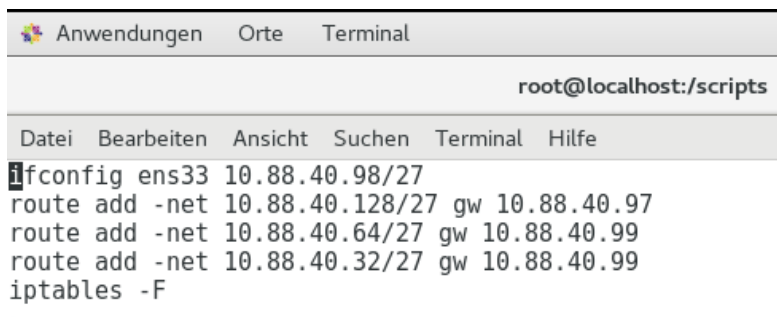
```
root@localhost:/scripts
File Edit View Search Terminal Help
[user1@localhost ~]$ su -
Password:
[root@localhost ~]# cd /scripts
[root@localhost scripts]# ls -l
total 4
-rwxrwxrwx. 1 root root 180 Nov  7 15:27 svr-mu.txt
[root@localhost scripts]# vi svr-mu.txt
[root@localhost scripts]# chmod 777 svr-mu.txt
[root@localhost scripts]# ./svr-mu.txt
[root@localhost scripts]#
```

Abbildung 23

Konfigurationsdatei von WEB-Hannover:(Routing-Tabelle)

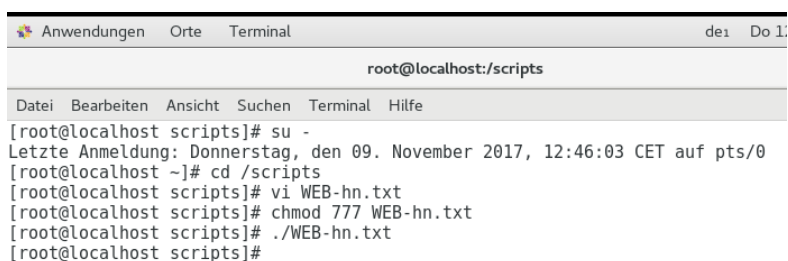
ifconfig ens33 weist der Netzwerkkarte die zugehörige IP-Adresse zu.

Mit dem Befehl `route add` werden statische Routen zu Rechnern und Netzwerken festgelegt.



```
root@localhost:/scripts
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
ifconfig ens33 10.88.40.98/27
route add -net 10.88.40.128/27 gw 10.88.40.97
route add -net 10.88.40.64/27 gw 10.88.40.99
route add -net 10.88.40.32/27 gw 10.88.40.99
iptables -F
```

Abbildung 24



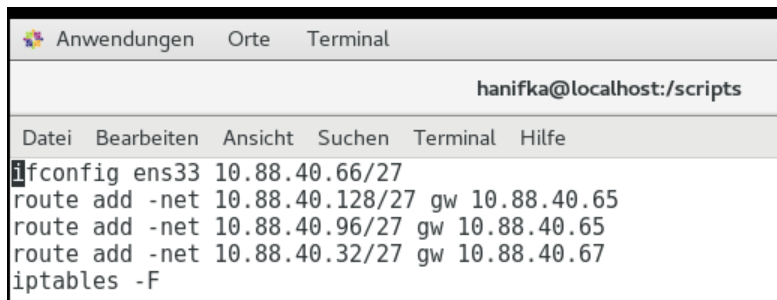
```
root@localhost:/scripts
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[root@localhost scripts]# su -
Letzte Anmeldung: Donnerstag, den 09. November 2017, 12:46:03 CET auf pts/0
[root@localhost ~]# cd /scripts
[root@localhost scripts]# vi WEB-hn.txt
[root@localhost scripts]# chmod 777 WEB-hn.txt
[root@localhost scripts]# ./WEB-hn.txt
[root@localhost scripts]#
```

Abbildung 25

Konfigurationsdatei von Server-Berlin:(Routing-Tabelle)

`ifconfig ens33` weist der Netzwerkkarte die zugehörige IP-Adresse zu.

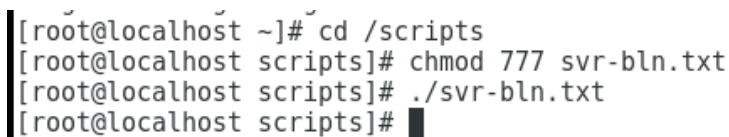
Mit dem Befehl `route add` werden statische Routen zu Rechnern und Netzwerken festgelegt.



A screenshot of a terminal window. The title bar shows 'Anwendungen', 'Orte', and 'Terminal'. The terminal prompt is 'hanifka@localhost:/scripts'. The commands entered are: 'ifconfig ens33 10.88.40.66/27', 'route add -net 10.88.40.128/27 gw 10.88.40.65', 'route add -net 10.88.40.96/27 gw 10.88.40.65', 'route add -net 10.88.40.32/27 gw 10.88.40.67', and 'iptables -F'.

```
hanifka@localhost:/scripts
ifconfig ens33 10.88.40.66/27
route add -net 10.88.40.128/27 gw 10.88.40.65
route add -net 10.88.40.96/27 gw 10.88.40.65
route add -net 10.88.40.32/27 gw 10.88.40.67
iptables -F
```

Abbildung 26



A screenshot of a terminal window. The terminal prompt is '[root@localhost ~]#'. The commands entered are: 'cd /scripts', 'chmod 777 svr-bln.txt', './svr-bln.txt', and the prompt returns to '[root@localhost scripts]#'.

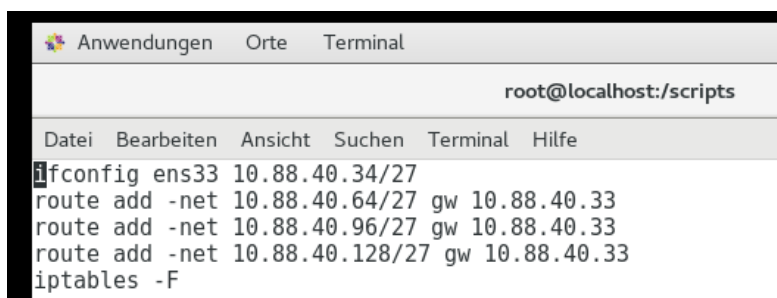
```
[root@localhost ~]# cd /scripts
[root@localhost scripts]# chmod 777 svr-bln.txt
[root@localhost scripts]# ./svr-bln.txt
[root@localhost scripts]#
```

Abbildung 27

Konfigurationsdatei von Server-Hamburg:(Routing-Tabelle)

ifconfig ens33 weist der Netzwerkkarte die zugehörige IP-Adresse zu.

Mit dem Befehl route add werden statische Routen zu Rechnern und Netzwerken festgelegt.



A screenshot of a terminal window. The title bar shows 'Anwendungen', 'Orte', and 'Terminal'. The terminal prompt is 'root@localhost:/scripts'. The commands entered are: 'ifconfig ens33 10.88.40.34/27', 'route add -net 10.88.40.64/27 gw 10.88.40.33', 'route add -net 10.88.40.96/27 gw 10.88.40.33', 'route add -net 10.88.40.128/27 gw 10.88.40.33', and 'iptables -F'.

```
root@localhost:/scripts
ifconfig ens33 10.88.40.34/27
route add -net 10.88.40.64/27 gw 10.88.40.33
route add -net 10.88.40.96/27 gw 10.88.40.33
route add -net 10.88.40.128/27 gw 10.88.40.33
iptables -F
```

Abbildung 28

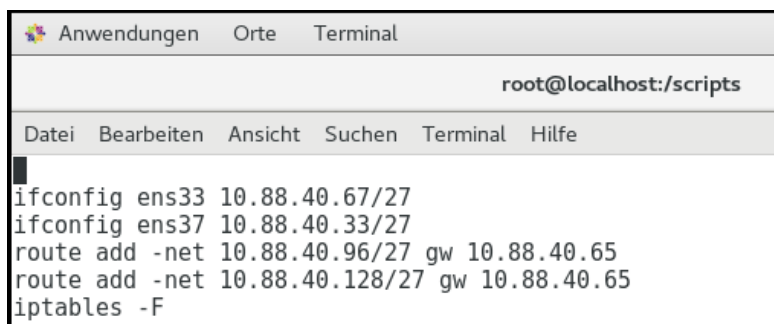
```
[root@localhost ~]# cd /scripts
[root@localhost scripts]# vi svr-hh.txt
[root@localhost scripts]# chmod 777 svr-hh.txt
[root@localhost scripts]# ./svr-hh.txt
[root@localhost scripts]# █
```

Abbildung 29

Konfigurationsdatei von Router 1:(Routing-Tabelle)

Wie in folgenden Ausschnitten zusehen ist haben wir die Router ebenfalls, so wie oben beschrieben, konfiguriert. Allerdings haben wir hier zwei Ethernet-Anbindungen. Denn ein Router hat immer eine Verbindung zwischen mindestens zwei Netzwerken und leitet Datenpakete anhand von Information der IP-Adressen zwischen den Netzwerken weiter. Mit `ifconfig ens33` und `ens37` weist man den Netzwerkkarten die zugehörige IP-Adresse zu.

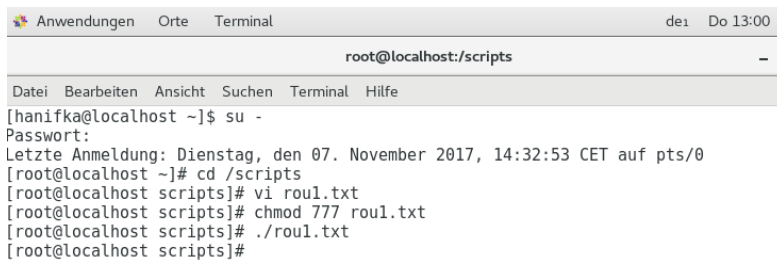
Mit dem Befehl `route add` werden statische Routen zu Rechnern und Netzwerken festgelegt.



The screenshot shows a terminal window with a title bar containing 'Anwendungen', 'Orte', and 'Terminal'. The terminal title is 'root@localhost:/scripts'. The terminal content shows the following commands and their output:

```
ifconfig ens33 10.88.40.67/27
ifconfig ens37 10.88.40.33/27
route add -net 10.88.40.96/27 gw 10.88.40.65
route add -net 10.88.40.128/27 gw 10.88.40.65
iptables -F
```

Abbildung 30



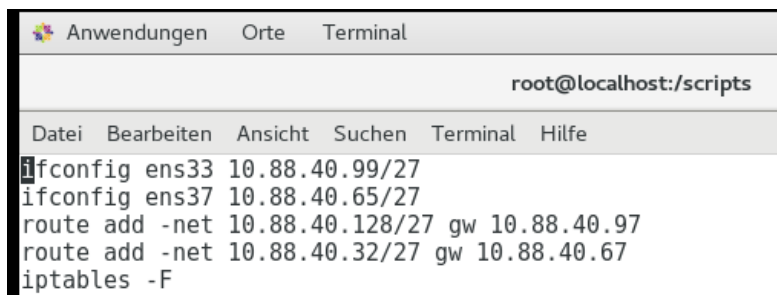
```
hanifka@localhost ~]$ su -
Passwort:
Letzte Anmeldung: Dienstag, den 07. November 2017, 14:32:53 CET auf pts/0
[root@localhost ~]# cd /scripts
[root@localhost scripts]# vi rou1.txt
[root@localhost scripts]# chmod 777 rou1.txt
[root@localhost scripts]# ./rou1.txt
[root@localhost scripts]#
```

Abbildung 31

Konfigurationsdatei von Router 2:(Routing-Tabelle)

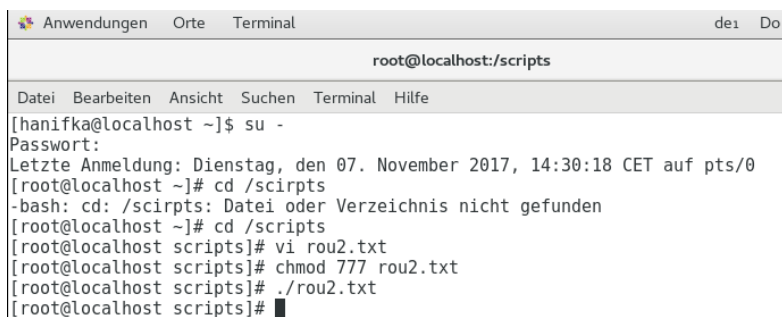
Mit `ifconfig ens33` und `ens37` weist man den Netzwerkkarten die zugehörige IP-Adresse zu.

Mit dem Befehl `route add` werden statische Routen zu Rechnern und Netzwerken festgelegt.



```
ifconfig ens33 10.88.40.99/27
ifconfig ens37 10.88.40.65/27
route add -net 10.88.40.128/27 gw 10.88.40.97
route add -net 10.88.40.32/27 gw 10.88.40.67
iptables -F
```

Abbildung 32



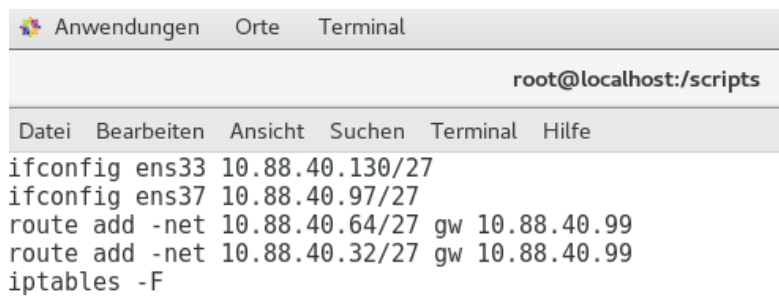
```
hanifka@localhost ~]$ su -
Passwort:
Letzte Anmeldung: Dienstag, den 07. November 2017, 14:30:18 CET auf pts/0
[root@localhost ~]# cd /scripts
-bash: cd: /scripts: Datei oder Verzeichnis nicht gefunden
[root@localhost ~]# cd /scripts
[root@localhost scripts]# vi rou2.txt
[root@localhost scripts]# chmod 777 rou2.txt
[root@localhost scripts]# ./rou2.txt
[root@localhost scripts]#
```

Abbildung 33

Konfigurationsdatei von Router 3:(Routing-Tabelle)

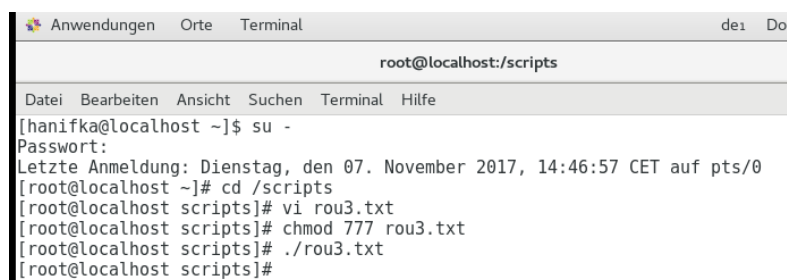
Mit `ifconfig ens33` und `ens37` weist man den Netzwerkkarten die zugehörige IP-Adresse zu.

Mit dem Befehl `route add` werden statische Routen zu Rechnern und Netzwerken festgelegt.



```
root@localhost:/scripts
ifconfig ens33 10.88.40.130/27
ifconfig ens37 10.88.40.97/27
route add -net 10.88.40.64/27 gw 10.88.40.99
route add -net 10.88.40.32/27 gw 10.88.40.99
iptables -F
```

Abbildung 34



```
root@localhost:/scripts
[hanifka@localhost ~]$ su -
Passwort:
Letzte Anmeldung: Dienstag, den 07. November 2017, 14:46:57 CET auf pts/0
[root@localhost ~]# cd /scripts
[root@localhost scripts]# vi rou3.txt
[root@localhost scripts]# chmod 777 rou3.txt
[root@localhost scripts]# ./rou3.txt
[root@localhost scripts]#
```

Abbildung 35

Please use Kali Linux as root: `# zenmap`

Scan the networks:

10.88.X.32/27

10.88.X.64/27

10.88.X.96/27

and 10.88.X.128/27

Zenmap

Zenmap ist eine grafische Ansicht für Nmap, der Ports scannen kann. Wenn man einen

Rechner auf offene Ports checken möchte, dann kommt Nmap zum Einsatz. Der Network Mapper ist dafür da, um alle aktiven Hosts in der Netzwerkumgebung (über Ping) sowie deren Betriebssystem und Versionsnummern installierter Dienste herauszufinden. Infolgedessen konnten wir mit dem Kommando `zenmap` die Verbindungen von Netzwerk 10.88.40.32, 10.88.40.64, 10.88.40.96 und 10.88.40.128 grafisch darstellen und demonstrieren, dass wir die oben aufgeführte Abbildung und somit unser Ziel erreicht haben.

Das Netz 10.88.40.32/27 wird gescannt:

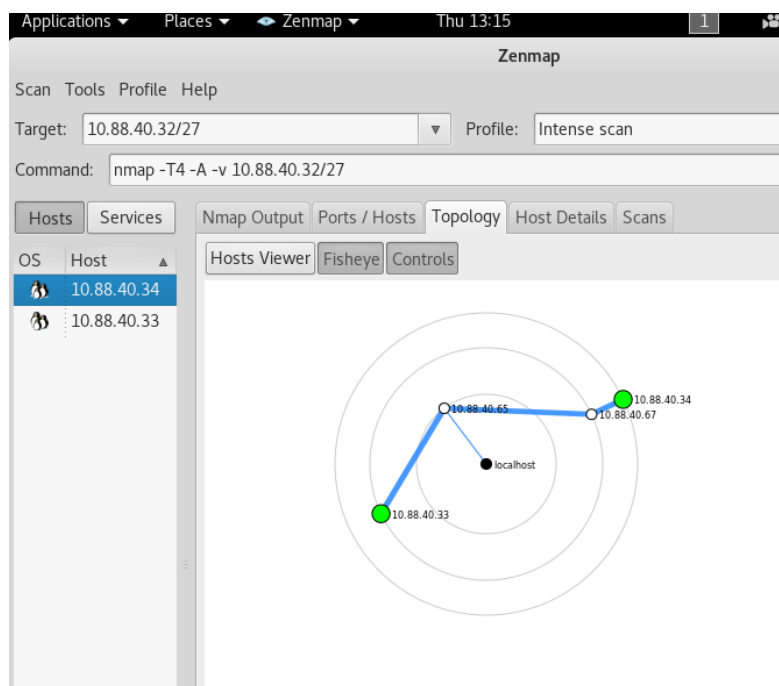


Abbildung 36: Netz: 10.88.40.32/27

Das Netz 10.88.40.64/27 wird gescannt:

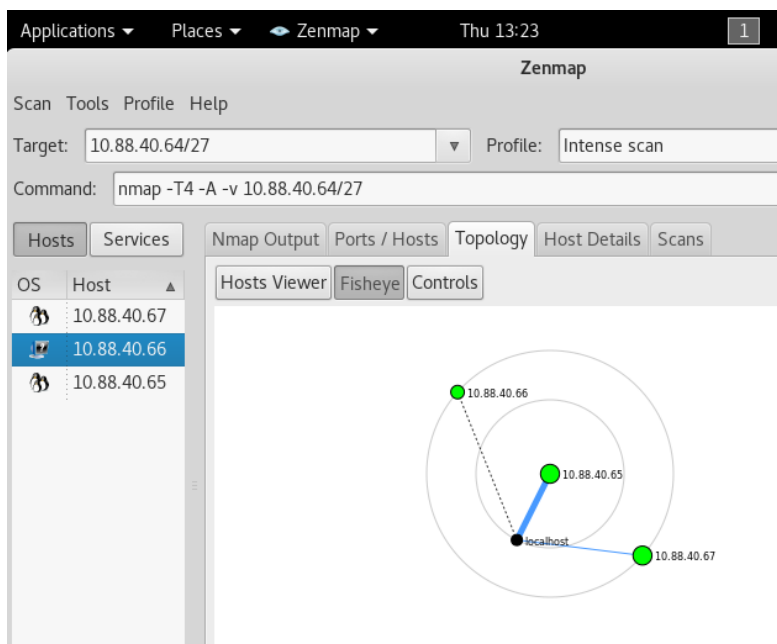


Abbildung 37: Netz: 10.88.40.64/27

Das Netz 10.88.40.96/27 wird gescannt:

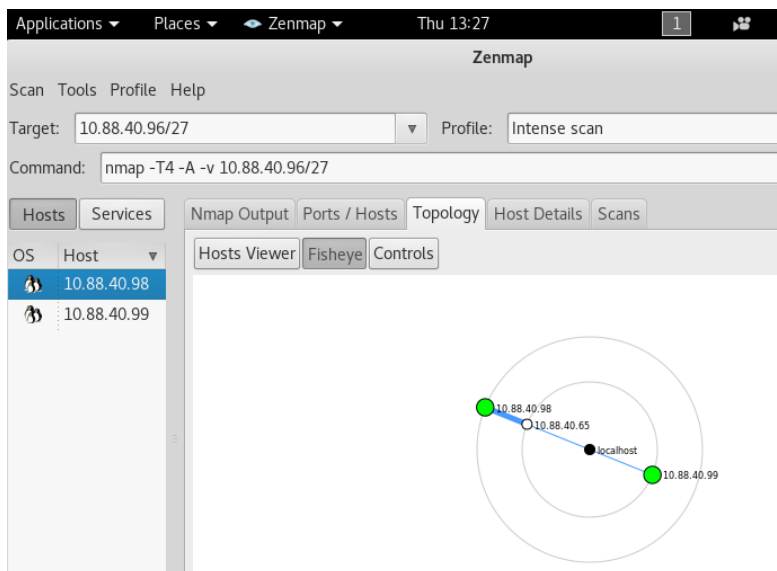


Abbildung 38: Netz: 10.88.40.96/27

Das Netz 10.88.40.128/27 wird gescannt:

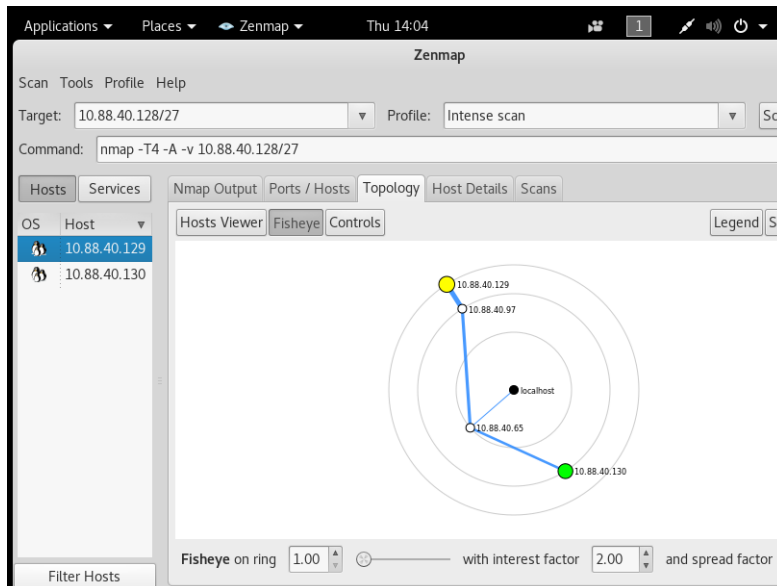


Abbildung 39: Netz: 10.88.40.128/27

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

Figure 1

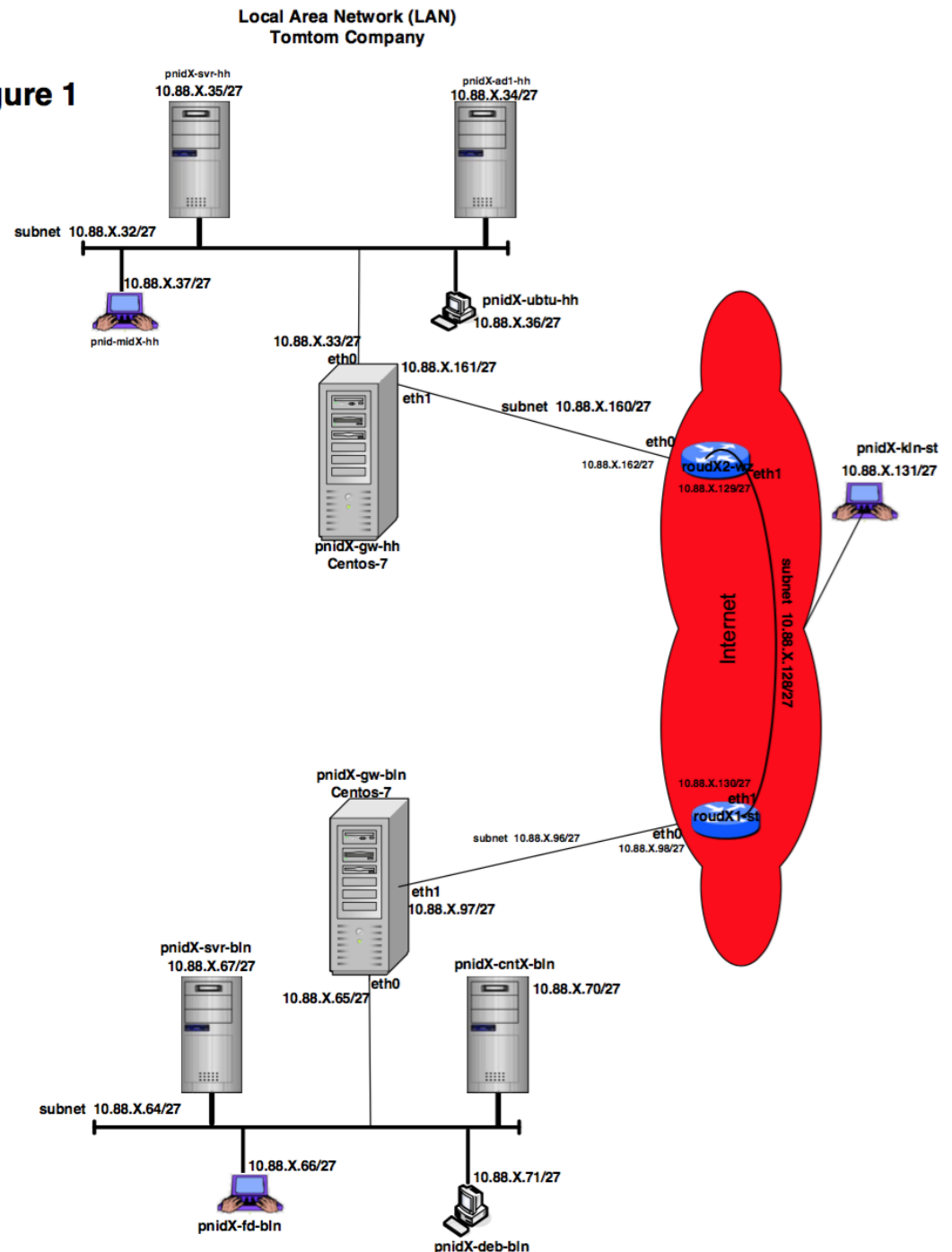


Abbildung 40: Netzwerk mit allen Hosts und Subnetzen

Exercise 1: Configure the networks of figure 1

a) Please copy, configure and set the networks for the following virtual machines provided by your instructor:

vm-Debian-8.5 copy from USB provided ,

vm-Ubuntu-16-10 copy from USB provided.

The password for the virtual machines is hamburg99tkrn for Ubuntu and Debian.

b) Please scan the following networks: 10.88.X.64/27, 10.88.X.96/27, 10.88.X.128/27, 10.88.X.160/27 and 10.88.X.32/27

c) Use Zenmap to scan all the above networks

Zenmap liefert uns alle statisch vergebenen IP Adressen der Hosts. Als Zusatz erhalten wir die Netzwerktopologie, ausgehend von dem aktuellen Host.

Solution of b)

Scan des Subnetzes 10.88.40.64/27



Abbildung 41: Subnetz 10.88.40.64/27

Scan des Subnetzes 10.88.40.96/27

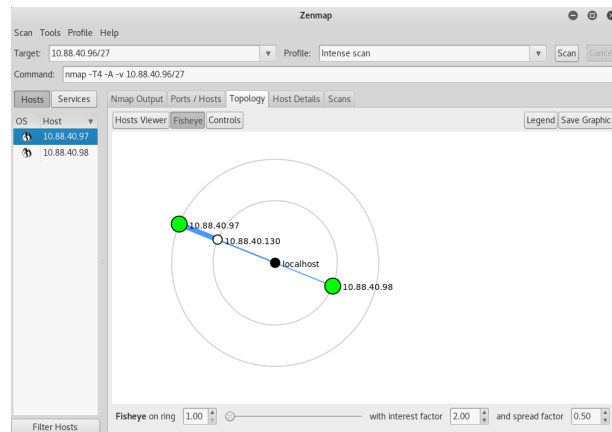


Abbildung 42: Subnetz 10.88.40.96/27

Scan des Subnetztes 10.88.40.128/27

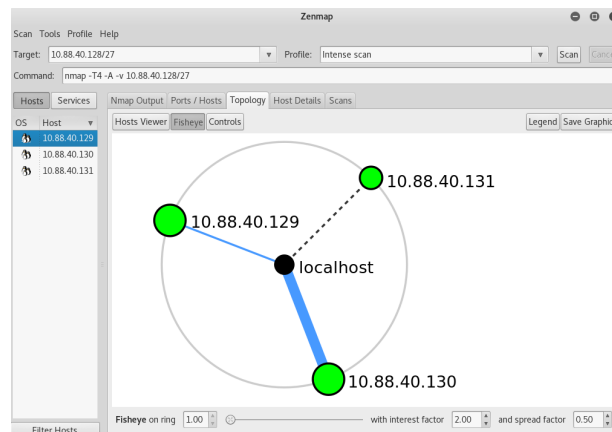


Abbildung 43: Subnetz 10.88.40.128/27

Scan des Subnetztes 10.88.40.160/27



Abbildung 44: Subnetz 10.88.40.160/27

Scan des Subnetztes 10.88.40.32/27

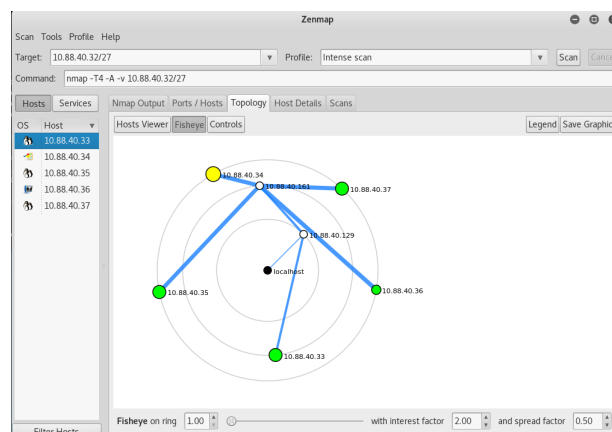


Abbildung 45: Subnetz 10.88.40.32/27

Solution of c) Use Zenmap to scan all the above networks

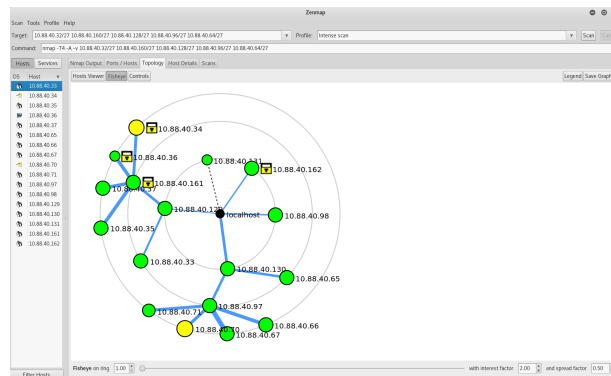


Abbildung 46: Alle Subnetze

Alternativ hätte man als Target auch folgendes in die Eingabemaske einfügen können:
10.88.40.32-160/27

Exercise 2: NMAP

Question 1: Please type and explain the nmap command: `nmap -sS -O 10.88.40.130`

Answer 1: Scannt das Subnetz 10.88.40.128 nach dem Host mit der IP Adresse 10.88.40.130 -sS bedeutet in diesem zusammenhang SYN-Stealth-Scan. Dabei wird keine vollständige TCP/IP Vergindung aufgebaut und ist deshalb unauffälliger als das der Parameter -sT, welcher als einziger ohne root rechte Funktioniert.

```
root@pmd4-klini:~# nmap -sS -O 10.88.40.130
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:13 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid
servers with --dns-servers
Nmap scan report for 10.88.40.130
Host is up (0.00062s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:50:56:39:6E:C2 (VMware)
Warning: OSscan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.2(4.X)
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.8, Linux 3.2 - 4.8
Network Distance: 1 hop
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.32 seconds
root@pmd4-klini:~#
```

Abbildung 47: Wireshark Filter ip.addr == 10.88.40.70

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.

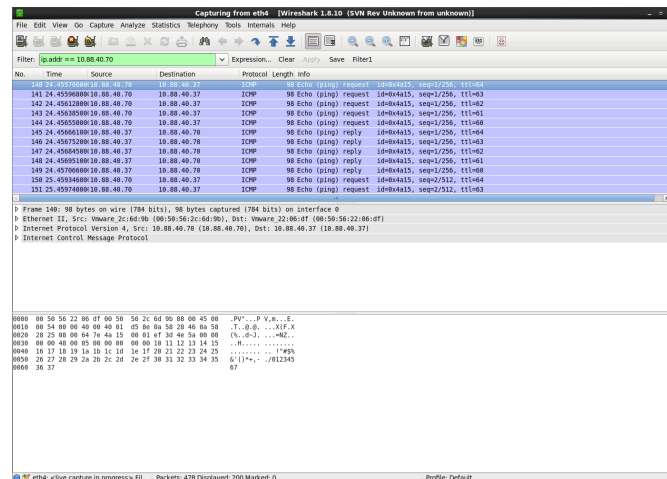


Abbildung 48: Wireshark Filter `ip.addr == 10.88.40.70`

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, $X < 2^{32} \wedge X \in \mathbb{N}$.

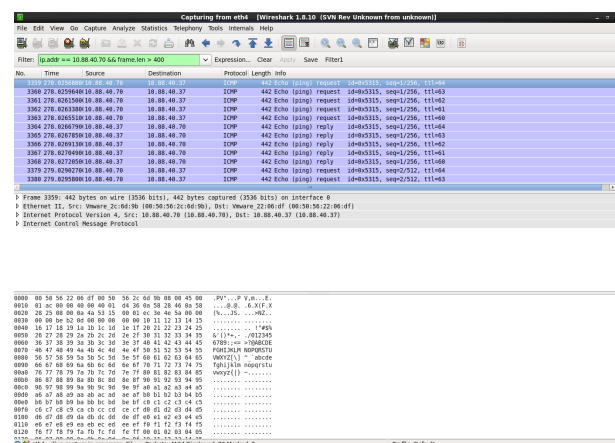


Abbildung 49: Wireshark Filter `ip.addr == 10.88.40.70 && frame.len > 400`

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.

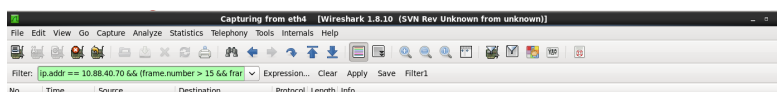


Abbildung 50: Wireshark Filter *ip.addr == 10.88.40.70 && (frame.number > 15 && frame.number < 30)*

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.

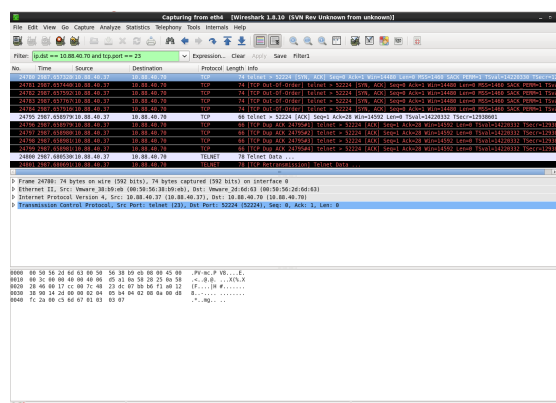


Abbildung 51: Wireshark Filter *ip.dst == 10.88.40.70 and tcp.port == 23*

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.

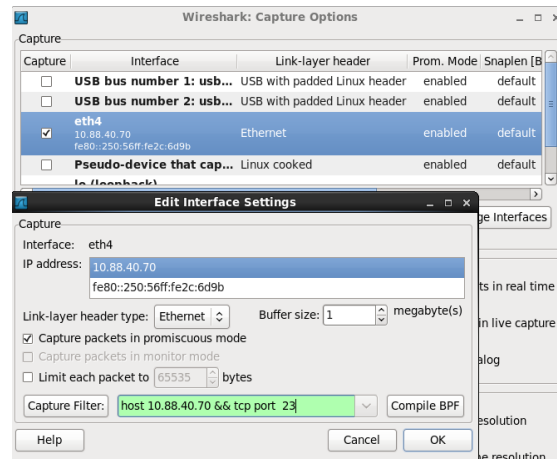


Abbildung 52: Wireshark Filter host 10.88.40.70 && tcp port 23

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.

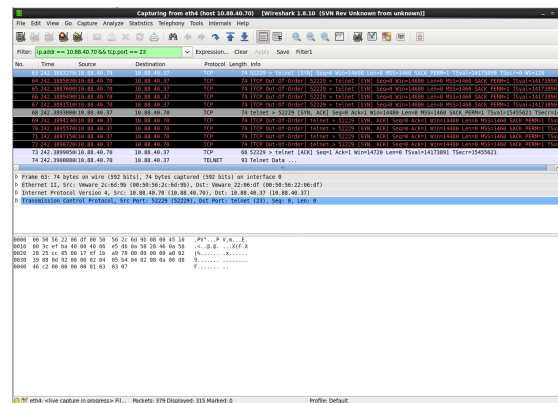


Abbildung 53: Wireshark Filter ip.addr == 10.88.40.70 && tcp.port == 23

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 ftp login Passwort herausfinden, da bei der Übertragung via ftp die Pakete unverschlüsselt übertragen werden. Dazu starten wir zunächst Wireshark auf dem Host pnid4-mid-hh und führen ein ftp login, von cnt-bln nach mid-hh, durch. Zuerst muss der Service ftp auf beiden Host aktiv sein, deshalb überprüfen wir den Status.

```
[root@localhost ~]# service vsftpd status
vsftpd (pid 1768) is running...
[root@localhost ~]#
```

Danach starten wir wireshark auf dem Host mid-hh und melden uns über den Host cnt-bln bei dem Host mid-hh über den ftp service an.

```
[root@localhost ~]# ftp 10.88.40.37
Connected to 10.88.40.37 (10.88.40.37).
220 (vsFTPd 2.2.2)
Name (10.88.40.37:root): trump4
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Über die Ausgabe Login successful sehen wir, dass das anmelden erfolgreich war. Wir öffnen nun Wireshark auf dem Host mid-hh und filtern nach ftp Paketen. Dazu reicht es aus ftp in die Filtermaske einzugeben.

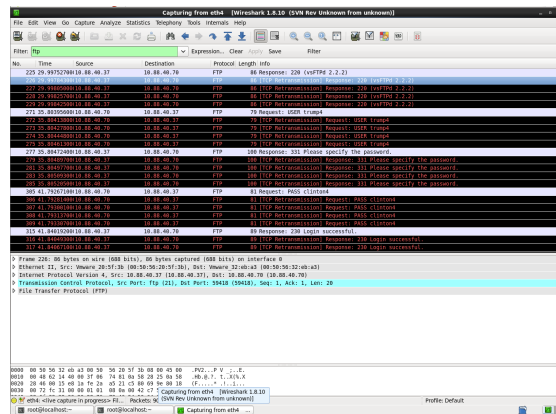


Abbildung 54: Wireshark Filter ftp

Wir schauen uns nun die Pakete genauer an und können die Logininformationen in einem der Pakete anzeigen lassen.

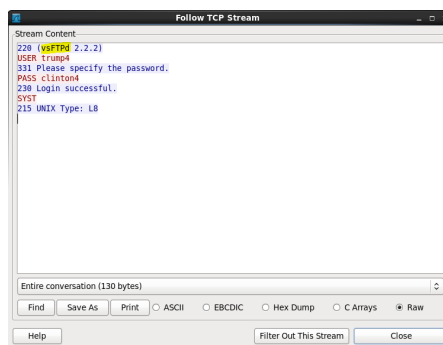


Abbildung 55: Wireshark ftp Login Passwort

Sofort sehen wir den Benutzernamen trump4 und das Passwort clinton4. Dieses Szenario zeigt wie einfach es ist die Logininformationen herauszulesen, wenn die Datenpakete unverschlüsselt übertragen werden.

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 ftp werden bei ssh (Secure Shell) die Pakete verschlüsselt übertragen, sodass es nicht möglich ist das Passwort mitzulesen. Zuerst prüfen wir, ob der ssh service auf beiden Hosts aktiv ist.

```
[root@localhost ~]# service sshd status
openssh-daemon (pid 1749) is running...
[root@localhost ~]#
```

Danach starten wir wireshark auf dem Host mid-hh und melden uns über den Host cnt-bln bei dem Host mid-hh über den ssh servie an.

```
[root@localhost ~]# ssh 10.88.40.37
root@10.88.40.37's password:
Last login: Thu Jan  4 13:55:43 2018 from 10.88.40.70
[root@localhost ~]#
```

Über die Ausgabe Last login..., sehen wir, dass das anmelden erfolgreich war. Wir öffnen nun Wireshark auf dem Host mid-hh und filtern nach ssh Paketen. Dazu reicht es aus ssh in die Filtermaske einzugeben.

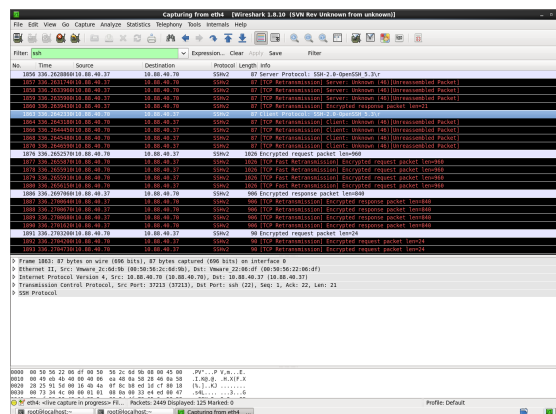


Abbildung 56: Wireshark Filter ssh

Wir schauen uns nun die Pakete genauer an und können keine Informationen über das Login erhalten, da alle Datenfragmente verschlüsselt wurden.

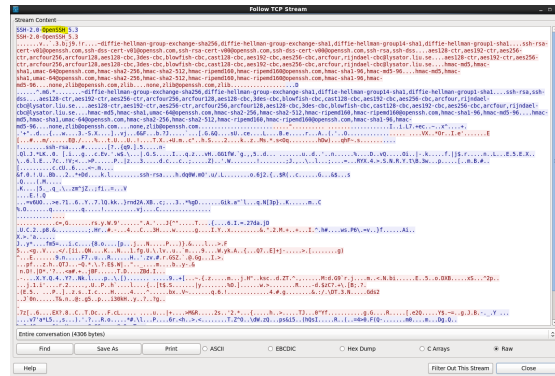


Abbildung 57: Wireshark verschlüsselte Datenfragmente

Exercise 5: Experimenting with network monitoring tools

Exercise: In this exercise you will connect to the webserver pnidX-cnt-blm from pnidX-mid-hh. Let pnidX-cnt-blm 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-blm. Let pnidX-mid-hh determine which services are running on pnidX-cnt-blm.

Solution: Wir führen zunächst nmap auf dem Host pnid4-cnt-blm 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.

```
[root@localhost ~]# nmap -sf -O 10.88.40.37

Starting Nmap 5.51 ( http://nmap.org ) at 2018-01-04 14:33 CET
mass dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify v
alid servers with --dns-servers
Nmap scan report for 10.88.40.37
Host is up (0.0025s latency)
Not shown: 995 closed ports
PORT      STATE      SERVICE
21/tcp    open|filtered  ftp
22/tcp    open|filtered  ssh
23/tcp    open|filtered  telnet
80/tcp    open|filtered  http
113/tcp   open|filtered  rcpd
Too many fingerprints match this host to give specific OS details
Nmap scan report for 10.88.40.37
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.80 seconds
[root@localhost ~]#
```

Abbildung 58: Zeigt uns die offenen Ports an

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.

```
[root@localhost ~]# telnet 10.88.40.37
Trying 10.88.40.37...
Connected to 10.88.40.37.
Escape character is '^]'.
CentOS release 6.5 (Final)
Kernel 2.6.32-431.el6.x86_64 on an x86_64
login: trump4
Password:
Last login: Thu Jan  4 14:04:27 from 10.88.40.70
[trump4@localhost ~]$
```

Abbildung 59: Anmeldung via Telnet

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

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

To create a host-to-host VPN as shown in Figure 1 using preshared keys both systems must have OPENS/WAN properly installed and tested. Next, you must ensure that IP networking is functioning. For this exercise you will capture http packets between the hosts. This capture will allow you to compare and prove that IPSec is functioning after the tunnel is created between the hosts. Make sure Apache and Wireshark are installed. See Figure 1.

Hints: CREATING PRESARED KEY use `ipsec ranbits 256 > filename`

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