

Lab report, 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	7
Exercise 6: Managing Services (Please use pnidX-svr-mu	7
Exercise 7: Configure the following network (figure 1) using ifconfig and route add	15
Exercise 8: Configure the following network (figure 1) using ip and nmcli . .	26
Exercise 9: Configure the following network (figure 1) using GUI	34
Part 4: Network Scanning	48
Exercise 1: Configure the networks of figure 1	49
Exercise 2: NMAP	52
Exercise 3: Nessus network device identification	57
Exercise 4: OpenVAS Network device identification	73
Part 5: Sniffing, Virtual Private Network (VPN)	84
Exercise 1: Configure and set the networks shown below (figure1 and 2) . . .	84
Exercise 2: Getting started with network monitoring tools	84
Exercise 3: TCPDUMP	84
Exercise 4: Wireshark	84
Exercise 5: Experimenting with network monitoring tools	91
Exercise 6: Set up a host-to-host VPN using preshared key	92
Exercise 7: Set up a host-to-host VPN using RSA keys	108
Exercise 8: Set up a network-to-network VPN using preshared key	108
Exercise 9: Set up a network-to-network VPN using RSA secrets keys	108

Abbildungsverzeichnis

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

Abbildung 27:	20
Abbildung 28:	20
Abbildung 29:	21
Abbildung 30:	21
Abbildung 31:	22
Abbildung 32:	22
Abbildung 33:	22
Abbildung 34:	23
Abbildung 35:	23
Abbildung 36: Netz: 10.88.40.32/27	24
Abbildung 37: Netz: 10.88.40.64/27	25
Abbildung 38: Netz: 10.88.40.96/27	25
Abbildung 39: Netz: 10.88.40.128/27	26
Abbildung 39: Konfig. Server München mit ip addr	28
Abbildung 39: Konfig. Router 3 mit ip addr	28
Abbildung 39: Konfig. Router 3 mit nmcli	29
Abbildung 39: Konfig. Router 2 mit ip addr	29
Abbildung 39: Konfig. Router 2 mit nmcli	30
Abbildung 39: Konfig. Router 1 mit ip addr	30
Abbildung 39: Konfig. Router 1 mit nmcli	30
Abbildung 39: Konfig. Webserver Hannover mit ip addr	31
Abbildung 39: Konfig. Server Berlin mit nmcli	31
Abbildung 39: Konfig. Server Hamburg mit ip addr	31
Abbildung P4 figure 1 LAN	48
Abbildung P5 ex. 1 Zenmap Subnetz 64	49
Abbildung P5 ex. 1 Zenmap Subnetz 96	50
Abbildung P5 ex. 1 Zenmap Subnetz 128	50
Abbildung P5 ex. 1 Zenmap Subnetz 160	51
Abbildung P5 ex. 1 Zenmap Subnetz 32	51
Abbildung P5 ex. 1 Zenmap alle Subnetze	52
Abbildung P4 ex. 2 nmap command 1	53
Abbildung P4 ex. 2 nmap command 2	53
Abbildung P4 ex. 2 nmap externes Logfile	54

Abbildung P4 ex. 2 nmap command 3	54
Abbildung P4 ex. 2 nmap command 4	55
Abbildung P4 ex. 2 nmap command 5	55
Abbildung P4 ex. 2 nmap command 6	55
Abbildung P4 ex. 2 nmap command 7	56
Abbildung P4 ex. 2 nmap command 8	56
Abbildung P4 ex. 2 nmap command 9	57
Abbildung P4 ex. 2 nmap command 10	57
Abbildung P4 ex. 3 installation Nessus	60
Abbildung P4 ex. 3 Konfiguration Nessus	61
Abbildung P4 ex. 3 Registrierung Nessus	62
Abbildung P4 ex. 3 Aktivierung Nessus	62
Abbildung P4 ex. 3 License Nessus	63
Abbildung P4 ex. 3 Scan Hosts in Subnetz 10.88.40.32/27	65
Abbildung P4 ex. 3 Ergebnis des Scans im Subnetz 10.88.40.32/27	66
Abbildung P4 ex. 3 Scan Hosts in Subnetz 10.88.40.64/27	66
Abbildung P4 ex. 3 Ergebnis des Scans im Subnetz 10.88.40.64/27	67
Abbildung P4 ex. 3 Scan Hosts in Subnetz 10.88.40.96/27	67
Abbildung P4 ex. 3 Ergebnis des Scans im Subnetz 10.88.40.96/27	68
Abbildung P4 ex. 3 Scan Hosts in Subnetz 10.88.40.128/27	68
Abbildung P4 ex. 3 Ergebnis des Scans im Subnetz 10.88.40.128/27	69
Abbildung P4 ex. 3 Scan Hosts in Subnetz 10.88.40.160/27	69
Abbildung P4 ex. 3 Ergebnis des Scans im Subnetz 10.88.40.160/27	70
Abbildung P4 ex. 3 Ergebnis des Scans aller Subnetze	70
Abbildung P4 ex. 3 Netzwerkkonfiguration über die GUI	72
Abbildung P4 ex. 3 erfolgreiche Internetverbindung	72
Abbildung P4 ex. 4 apt-get update	77
Abbildung P4 ex. 4 apt-get upgrade	77
Abbildung P4 ex. 4 OpenVas Installation	78
Abbildung P4 ex. 4 OpenVas Konfiguration	78
Abbildung P4 ex. 4 OpenVas user anlegen	79
Abbildung P4 ex. 4 OpenVas Login	79
Abbildung P4 ex. 4 OpenVas Scan des Subnetzes 10.88.40.32/27	80

Abbildung P4 ex. 4 OpenVas Scan des Subnetzes 10.88.40.64/27	80
Abbildung P4 ex. 4 OpenVas Scan des Subnetzes 10.88.40.96/27	81
Abbildung P4 ex. 4 OpenVas Scan des Subnetzes 10.88.40.128/27	81
Abbildung P4 ex. 4 OpenVas Scan des Subnetzes 10.88.40.160/27	82
Abbildung P4 ex. 4 OpenVas Scan alle Subnetze	83
Abbildung P5 ex. 4 Wireshark Filter 1	85
Abbildung P5 ex. 4 Wireshark Filter 2	85
Abbildung P5 ex. 4 Wireshark Filter 3	86
Abbildung P5 ex. 4 Wireshark Filter 4	86
Abbildung P5 ex. 4 Wireshark Filter 5	87
Abbildung P5 ex. 4 Wireshark Filter 7	88
Abbildung P5 ex. 4 Wireshark Filter 8	89
Abbildung P5 ex. 4 Wireshark ftp Login Passwort	89
Abbildung P5 ex. 4 Wireshark Filter 9	90
Abbildung P5 ex. 4 Wireshark ssh Datenpaket	91
Abbildung P5 ex. 5 nmap offene Ports anzeigen	91
Abbildung P5 ex. 5 Telnet login	92
Abbildung P5 Installation Openswan	93
Abbildung P5 Figure 1: HOST-TO-HOST-VPN	94
Abbildung P5 dst 10.88.40.130 && tcp && port 80	95
Abbildung P5 http://10.88.40.130	96
Abbildung P5 Filter: dst 10.88.40.130 && tcp && port 80	96
Abbildung P5 ipsec ranbits 256 >	97
Abbildung P5 psk.secrets	97
Abbildung P5: psk.conf	98
Abbildung P5 ipsec setup reload	99
Abbildung P5 ipsec auto –add	100
Abbildung P5 ipsec auto –up	100

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
abrtd 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:on 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
nflock 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
ntpdate 0:off 1:off 2:off 3:off 4:off 5:off 6:off
oddijobd 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
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running Transaction Check
--> Package xinetd.x86_64 2:2.3.14-39.el6_4 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

Transaction Summary
Install 1 Package(s)

Total download size: 121 k
Installed size: 259 k
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Running 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 "diable = 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` anzeigenlassen, 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
kinetd      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

kinetd 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: 151 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
1/1
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`" 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 aktivierte 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@root@localhost:~]
[  OK  ]
```

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-bln 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.

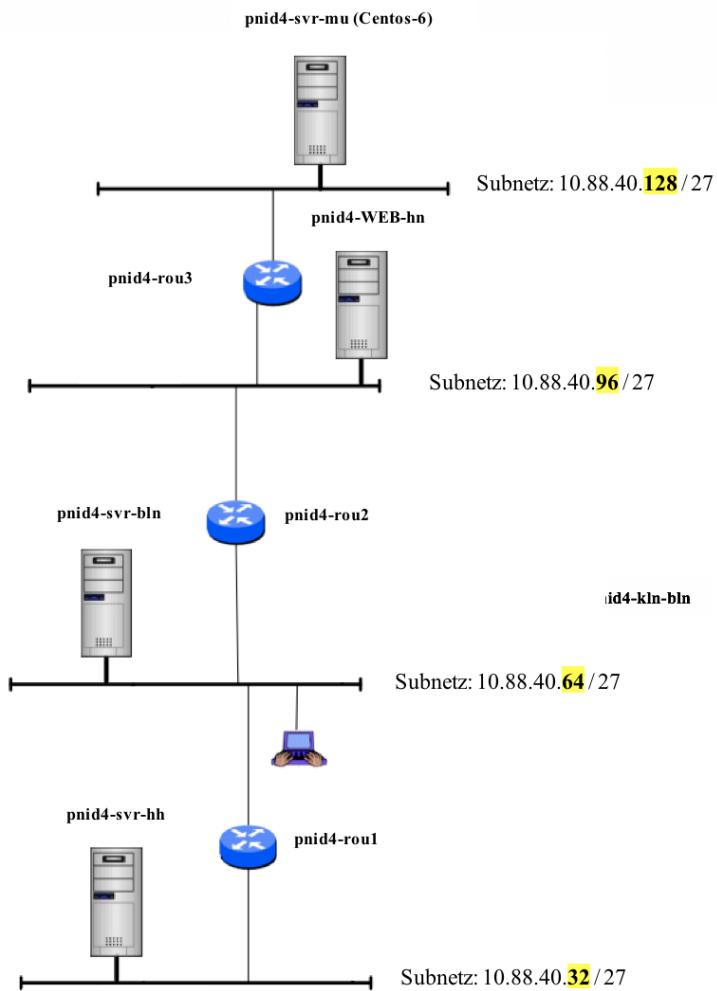


Abbildung 21

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.

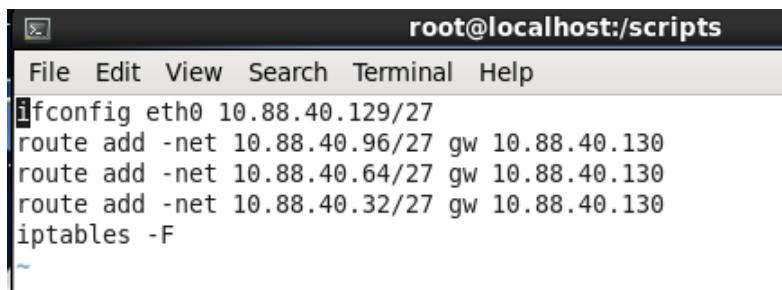
Anschließend haben wir die Routing tables über dem Kommando route add erstellt. Hier haben wir die Netze 10.88.40.96, 10.88.40.64 und 10.88.40.32 mit dem Kommando versehen, da wir über alle drei Netze eine Verbindung herstellen möchten.

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)

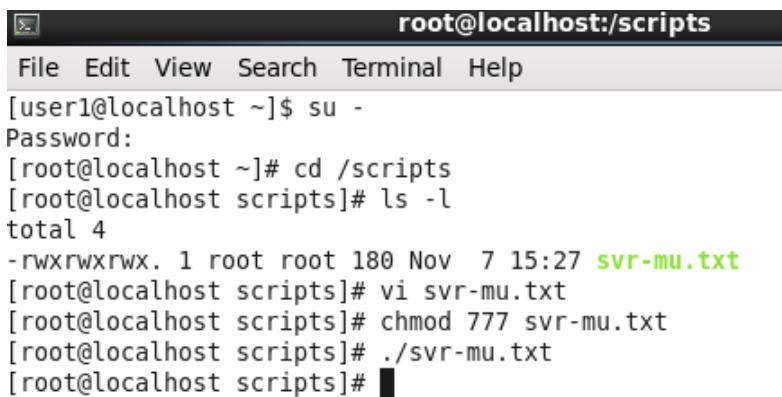
ifconfig eth0 weißt der Netzwerkkarte die zugehörige IP-Adresse zu.

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



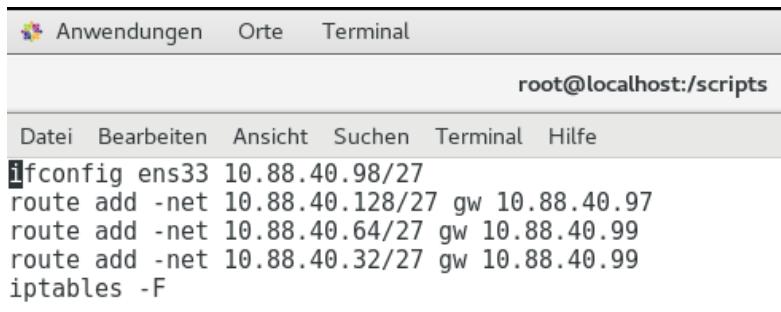
```
root@localhost:/scripts
File Edit View Search Terminal Help
[user1@localhost ~]$ su -
Password:
[root@localhost ~]# cd /scripts
[root@localhost scripts]# ls -l
total 4
-rwxrwxrwx. 1 root root 180 Nov  7 15:27 svr-mu.txt
[root@localhost scripts]# vi svr-mu.txt
[root@localhost scripts]# chmod 777 svr-mu.txt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

Abbildung 23

Konfigurationsdatei von WEB-Hannover:(Routing-Tabelle)

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

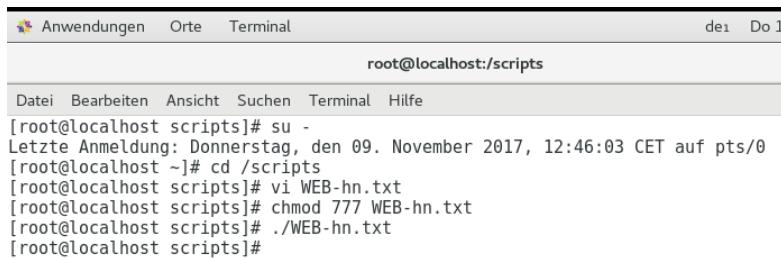
Mit dem Befehl route add werden statische Routen zu Rechnern und Netzwerken festgeleget.



The screenshot shows a terminal window with a menu bar at the top. The menu bar includes "Anwendungen", "Orte", "Terminal", and "root@localhost:/scripts". Below the menu bar is a toolbar with "Datei", "Bearbeiten", "Ansicht", "Suchen", "Terminal", and "Hilfe". The main area of the terminal displays the following command output:

```
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



The screenshot shows a terminal window with a menu bar at the top. The menu bar includes "Anwendungen", "Orte", "Terminal", and "root@localhost:/scripts". Below the menu bar is a toolbar with "Datei", "Bearbeiten", "Ansicht", "Suchen", "Terminal", and "Hilfe". The main area of the terminal displays the following command output:

```
[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)

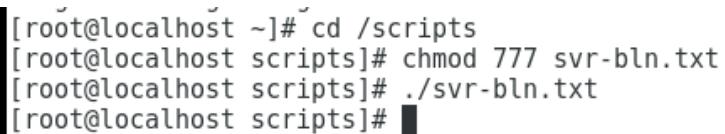
ifconfig ens33 weißt der Netzwerkkarte die zugehörige IP-Adresse zu.

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



```
Anwendungen  Orte  Terminal  
hanifka@localhost:/scripts  
Datei  Bearbeiten  Ansicht  Suchen  Terminal  Hilfe  
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



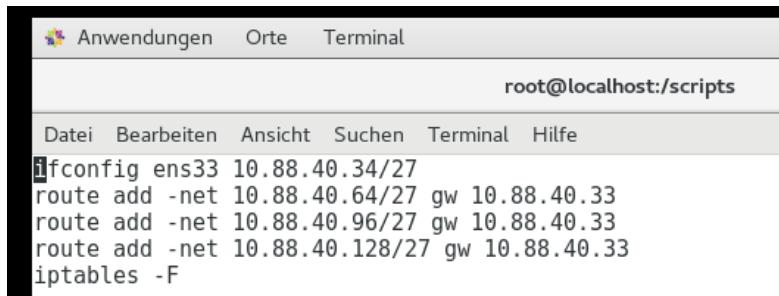
```
[root@localhost ~]# cd /scripts  
[root@localhost scripts]# chmod 777 svr-bln.txt  
[root@localhost scripts]# ./svr-bln.txt  
[root@localhost scripts]# █
```

Abbildung 27

Konfigurationsdatei von Server-Hamburg:(Routing-Tabelle)

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

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



```
Anwendungen  Orte  Terminal  
root@localhost:/scripts  
Datei  Bearbeiten  Ansicht  Suchen  Terminal  Hilfe  
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 weißt man den Netzwerkarten die zugehörige IP-Adresse zu.

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

```
Anwendungen Orte Terminal
root@localhost:/scripts
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
█
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

```

Anwendungen Orte Terminal de1 Do 13:00
root@localhost:/scripts -
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[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.

```

Anwendungen Orte Terminal de1 Do
root@localhost:/scripts -
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
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

```

Anwendungen Orte Terminal de1 Do
root@localhost:/scripts -
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[hanifka@localhost ~]$ su -
Passwort:
Letzte Anmeldung: Dienstag, den 07. November 2017, 14:30:18 CET auf pts/0
[root@localhost ~]# cd /scirpts
-bash: cd: /scirpts: 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]#

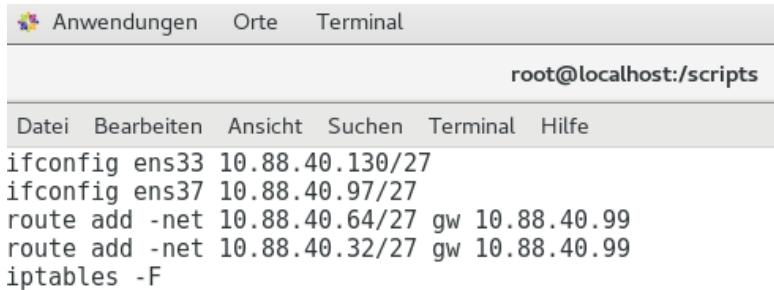
```

Abbildung 33

Konfigurationsdatei von Router 3:(Routing-Tabelle)

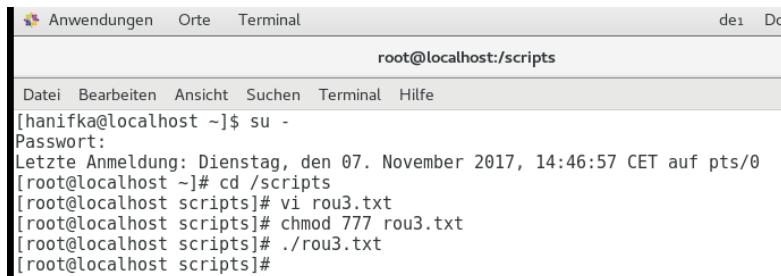
Mit ifconfig ens33 und ens37 weißt man den Netzwerkkarten die zugehörige IP-Adresse zu.

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



```
Anwendungen Orte Terminal
root@localhost:/scripts
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
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



```
Anwendungen Orte Terminal de1 Do
root@localhost:/scripts
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[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:

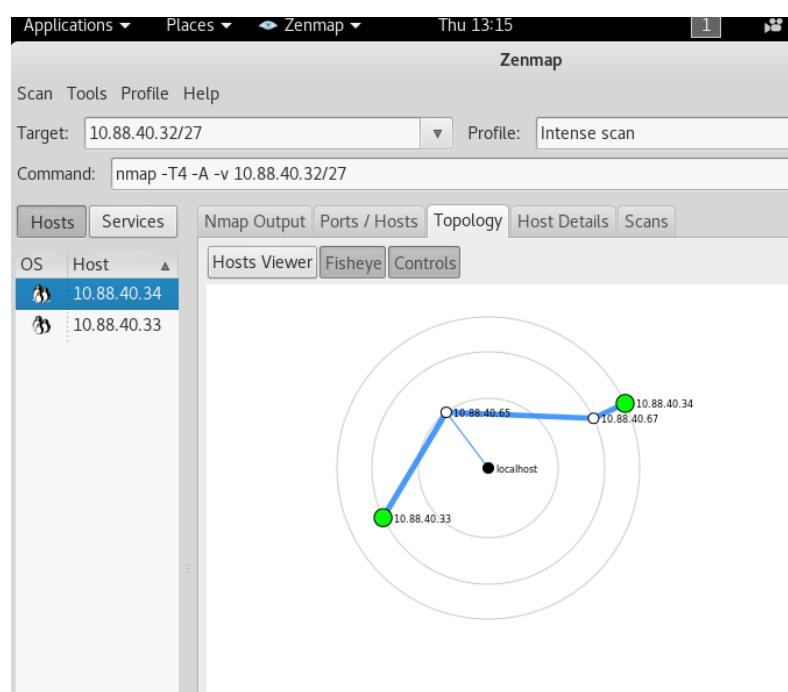


Abbildung 36: Netz: 10.88.40.32/27

Das Netz 10.88.40.64/27 wird gescannt:

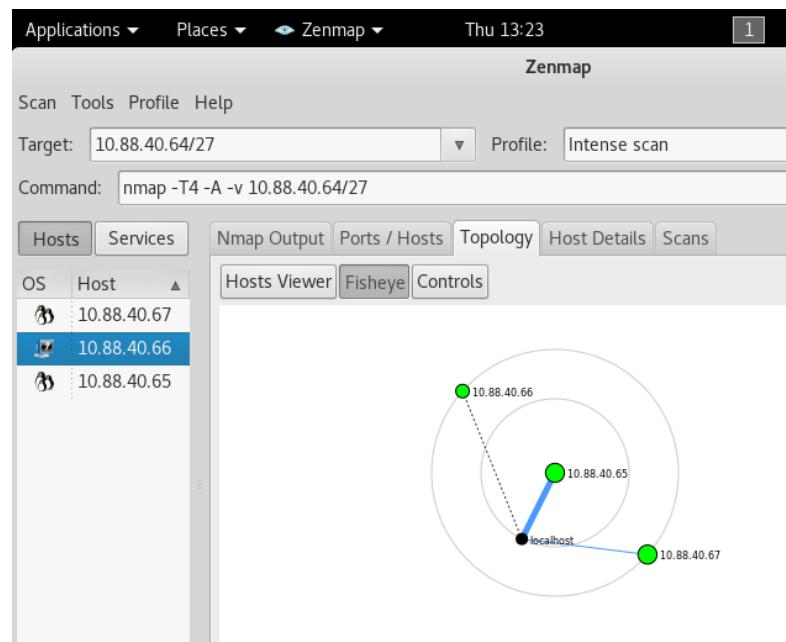


Abbildung 37: Netz: 10.88.40.64/27

Das Netz 10.88.40.96/27 wird gescannt:

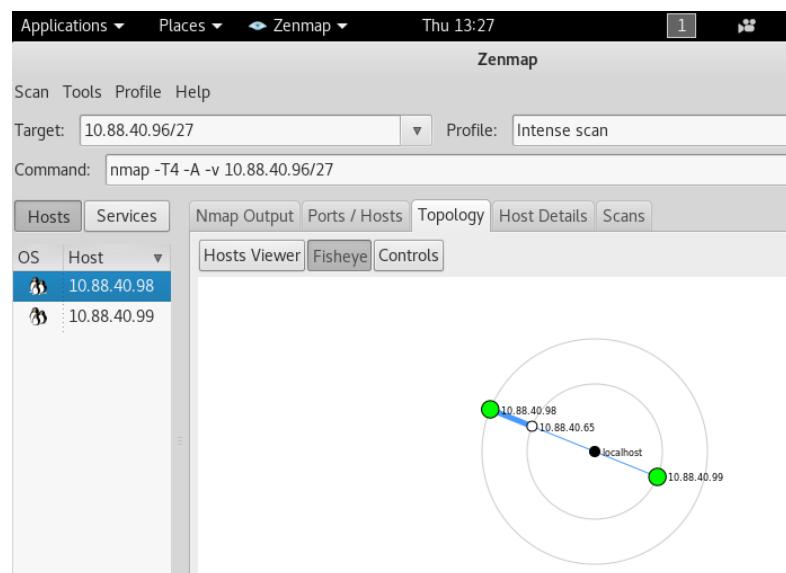


Abbildung 38: Netz: 10.88.40.96/27

Das Netz 10.88.40.128/27 wird gescannt:

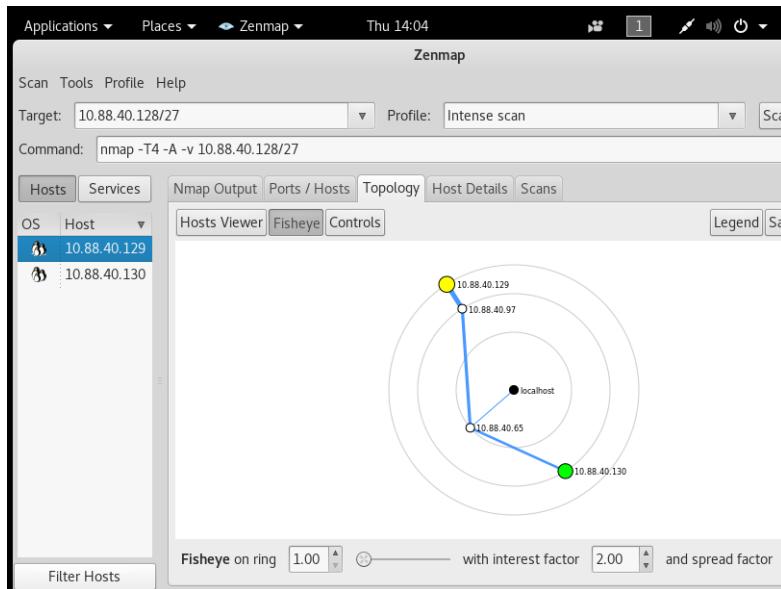


Abbildung 39: Netz: 10.88.40.128/27

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

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

10.88.X.128/27

Einleitung:

In dieser Aufgabe geht es darum das Netzwerk mittels ip und nmcli zu konfigurieren. Ip ist die neuere Version des Kommandozeilenprogramms von ifconfig. Demzufolge ist Sie leistungsfähiger und wird irgendwann ifconfig ersetzen. Man muss sich nur mit der Syntax vertraut machen. In den nächsten Schritten wird gezeigt, wie man mittels ip die Netze konfiguriert. In dieser Aufgabe gehen wir vom Aufbau her wie in Exercise 7 vor. Deswegen betrachten wir der übersichtshalber nur die vi-Datei, die entsprechend angepasst werden muss.

Vergleich von Nmcli und IP:

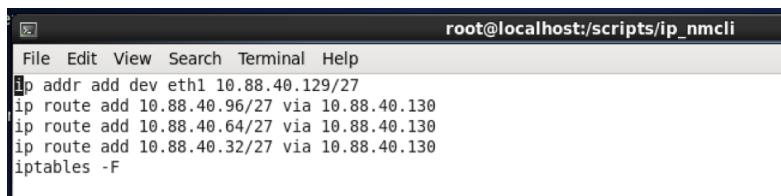
Nmcli auch bekannt als Network Manager hat viele Vorteile gegenüber anderen Verfahren, insbesondere bei einer WLAN-Verbindung. Denn Nmcli regelt die Verbindung zum Netzwerk nicht ausschließlich beim Hochfahren, sondern überwacht auch im Laufe eines Betriebes die Verbindung und wenn nötig stellt nmcli die Verbindung nach Unterbrechungen wieder her. Deswegen ist der Network Manager auch für Server-Installationen interessant, quasi als Ersatz für die Netzwerkkonfiguration. Auch bei der Fehlersuche spielt nmcli eine bedeutende Rolle. Die Möglichkeiten von nmcli und die dabei zu verwendende Kommandosprache sind versionsabhängig. Zunächst betrachten wir die für CentOS 7 verwendete Kommandosprache.

Ähnlich wie bei ip wird zunächst die IP-Adresse festgelegt. Dies geschieht mit dem Befehl nmcli con add con-name mit dem jeweiligen Namen der Netzwerkkarte. In diesem Fall ist es ens33 mit dem Typen type ethernet ifname, wie man der Abbildung 48

entnehmen kann. Anschließend folgt die IP-Adresse in Version 4. Deswegen schreibt man ip4 und direkt danach die dazugehörige IP-Adresse mit der entsprechenden Subnetzmaske. In diesem Fall wieder 255.255.255.224 und vereinfacht dargestellt als /27 direkt hinter der IP-Adresse.

Die nächsten Zeilen legen fest, durch welches Gateway ein Datenpaket weitergeleitet werden soll, wenn es nicht im entsprechenden Subnetz ist, sprich wenn die Netzadresse nicht dieselbe ist. Der Befehl nmcli connection modify dient dazu die entsprechenden Routen zu bestimmen. Im Anschluss steht der Name der Netzwerkkarte. Der Befehl +ipv4.routes sorgt dafür, dass das entsprechende Subnetz angegeben wird, in das geroutet werden soll. Dafür muss man noch das jeweilige Gateway angeben, durch welches man in das entsprechende Subnetz gelangt. Dies geschieht mittels ipv4.gateway. Am Ende wird noch mit dem Befehl iptables -F die Firewall ausgeschaltet.

Konfigurationsdatei von Server München mittels ip addr (Routing-Tabelle)



```
root@localhost:/scripts/ip_nmcli
File Edit View Search Terminal Help
ip addr add dev eth1 10.88.40.129/27
ip route add 10.88.40.96/27 via 10.88.40.130
ip route add 10.88.40.64/27 via 10.88.40.130
ip route add 10.88.40.32/27 via 10.88.40.130
iptables -F
```

Abbildung 40: Konfig. Server München ip addr

Konfigurationsdatei von Router 3 mittels ip addr (Routing-Tabelle)

```
ip addr add dev ens33 10.88.40.130/27
ip addr add dev ens37 10.88.40.97/27
ip route add 10.88.40.64/27 via 10.88.40.99
ip route add 10.88.40.32/27 via 10.88.40.99
iptables -F
```

Abbildung 41: Konfig. Router 3 ip addr

Gleiche Konfigurationsdatei von Router 3 mittels nmcli (Routing-Tabelle)

```

Anwendungen Orte Terminal de1 Sa 0
root@localhost:/scripts/nmcli

Datei Bearbeiten Ansicht Suchen Terminal Hilfe
nmcli con add con-name ens33 type ethernet ifname ens33 ip4 10.88.40.130/27
nmcli con add con-name ens37 type ethernet ifname ens37 ip4 10.88.40.97/27
nmcli con mod ens33 +ipv4.routes 10.88.40.64/27 ipv4.gateway 10.88.40.99
nmcli con mod ens33 +ipv4.routes 10.88.40.32/27 ipv4.gateway 10.88.40.99
iptables -F

```

Abbildung 42: Konfig. Router 3 nmcli

Nach dem Ausführen des Skriptes wurden die Verbindungen ens33 und ens37 erfolgreich hinzugefügt. Mit nmcli con up wird die Verbindung aktiviert und anschließend wird über netstat -nr die Routing-Tabelle angezeigt.

```

[root@localhost nmcli]# chmod 777 rou-3.txt
[root@localhost nmcli]# ./rou-3.txt
Verbindung »ens33« (318e4197-c623-45c3-baac-d1234d67f7c9) erfolgreich hinzugefügt.
Verbindung »ens37« (cb236514-d214-49ab-95fc-03fc165ca9f0) erfolgreich hinzugefügt.
[root@localhost nmcli]# nmcli con up ens37
Verbindung wurde erfolgreich aktiviert (aktiver D-Bus-Pfad: /org/freedesktop/NetworkManager/ActiveConnection/20)
[root@localhost nmcli]# nmcli con up ens33
Verbindung wurde erfolgreich aktiviert (aktiver D-Bus-Pfad: /org/freedesktop/NetworkManager/ActiveConnection/21)
[root@localhost nmcli]# netstat -nr
Kernel IP Routentabelle
Ziel      Router    Genmask     Flags   MSS Fenster irtt Iface
0.0.0.0    10.88.40.99  0.0.0.0    UG        0 0       0 ens33
10.88.40.32  0.0.0.0    255.255.255.224 U        0 0       0 ens33
10.88.40.64  0.0.0.0    255.255.255.224 U        0 0       0 ens33
10.88.40.96  0.0.0.0    255.255.255.224 U        0 0       0 ens37
10.88.40.99  0.0.0.0    255.255.255.255 UH       0 0       0 ens33
10.88.40.128 0.0.0.0    255.255.255.224 U        0 0       0 ens33
192.168.122.0 0.0.0.0    255.255.255.0   U        0 0       0 virbr0
[root@localhost nmcli]#

```

Konfigurationsdatei von Router 2 mittels ip addr (Routing-Tabelle)

```

Anwendungen Orte Terminal root@localhost:/scripts/ip_nmcli

Datei Bearbeiten Ansicht Suchen Terminal Hilfe
ip addr add dev ens33 10.88.40.99/27
ip addr add dev ens37 10.88.40.65/27
ip route add 10.88.40.128/27 via 10.88.40.97
ip route add 10.88.40.32/27 via 10.88.40.67
iptables -F

```

Abbildung 43: Konfig. Router 2 ip addr

Gleiche Konfigurationsdatei von Router 2 mittels nmcli (Routing-Tabelle)

```

Anwendungen Orte Terminal de1 Sa 00:28
root@localhost:/scripts/test
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
nmcli con add con-name ens33 type ethernet ifname ens33 ip4 10.88.40.99/27
nmcli con add con-name ens37 type ethernet ifname ens37 ip4 10.88.40.65/27
nmcli connection modify ens33 +ipv4.routes 10.88.40.32/27 ipv4.gateway 10.88.40.67
nmcli connection modify ens37 +ipv4.routes 10.88.40.128/27 ipv4.gateway 10.88.40.97
iptables -F
~
```

Abbildung 44: Konfig. Router 2 nmcli

Konfigurationsdatei von Router 1 mittels ip addr (Routing-Tabelle)

```

Anwendungen Orte Terminal
root@localhost:/scripts/ip_nmcli
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
ip addr add dev ens33 10.88.40.33/27
ip addr add dev ens37 10.88.40.67/27
ip route add 10.88.40.128/27 via 10.88.40.65
ip route add 10.88.40.96/27 via 10.88.40.65
iptables -F
```

Abbildung 45: Konfig. Router 1 ip addr

Gleiche Konfigurationsdatei von Router 1 mittels nmcli (Routing-Tabelle)

```

Datei Bearbeiten Ansicht Suchen Terminal Hilfe
nmcli con add con-name ens33 type ethernet ifname ens33 ip4 10.88.40.33/27
nmcli con add con-name ens37 type ethernet ifname ens37 ip4 10.88.40.67/27
nmcli connection modify ens33 +ipv4.routes 10.88.40.128/27 ipv4.gateway 10.88.40.65
nmcli connection modify ens33 +ipv4.routes 10.88.40.96/27 ipv4.gateway 10.88.40.65
nmcli connection modify ens37 +ipv4.routes 10.88.40.128/27 ipv4.gateway 10.88.40.65
nmcli connection modify ens37 +ipv4.routes 10.88.40.96/27 ipv4.gateway 10.88.40.65
iptables -F
```

Abbildung 46: Konfig. Router 1 nmcli

Konfigurationsdatei von Webserver Hannover mittels ip addr (Routing-Tabelle)

```

Anwendungen Orte Terminal
root@localhost:/scripts/ip_nmcli
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
ip addr add dev ens33 10.88.40.98/27
ip route add 10.88.40.128/27 via 10.88.40.97
ip route add 10.88.40.64/27 via 10.88.40.99
ip route add 10.88.40.32/27 via 10.88.40.99
iptables -F

```

Abbildung 47: Konfig. Webserver Hannover ip addr

Konfigurationsdatei von Server Berlin mittels nmcli (Routing-Tabelle)

```

Anwendungen Orte Terminal de1 So 16:36
root@localhost:/scripts/nmcli
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
nmcli con add con-name ens33 type ethernet ifname ens33 ip4 10.88.40.66/27
nmcli connection modify ens33 +ipv4.routes 10.88.40.96/27 ipv4.gateway 10.88.40.65
nmcli connection modify ens33 +ipv4.routes 10.88.40.128/27 ipv4.gateway 10.88.40.65
nmcli connection modify ens33 +ipv4.routes 10.88.40.32/27 ipv4.gateway 10.88.40.67
iptables -F

```

Abbildung 48: Konfig. Server Berlin nmcli

```

[root@localhost nmcli]# ./svr-bln.txt
[Verbindung >ens33< (fee9cc86-f0aa-48ab-b377-c76b71ceb9bb) erfolgreich hinzugefügt.
[root@localhost nmcli]#
[root@localhost nmcli]# nmcli con up ens33
[Verbindung wurde erfolgreich aktiviert (aktiver D-Bus-Pfad: /org/freedesktop/NetworkManager/ActiveConnection/9)
[root@localhost nmcli]# netstat -nr
Kernel IP Routentabelle
Ziel      Router      Genmask      Flags   MSS Fenster irtt Iface
0.0.0.0    10.88.40.67  0.0.0.0      UG     0 0        0 ens33
10.88.40.32 0.0.0.0    255.255.255.224 U       0 0        0 ens33
10.88.40.64 0.0.0.0    255.255.255.224 U       0 0        0 ens33
10.88.40.96 0.0.0.0    255.255.255.224 U       0 0        0 ens33
10.88.40.128 0.0.0.0   255.255.255.224 U       0 0        0 ens33
192.168.122.0 0.0.0.0   255.255.255.0   U       0 0        0 virbr0
[root@localhost nmcli]#

```

Konfigurationsdatei von Server Hamburg mittels ip addr (Routing-Tabelle)

```

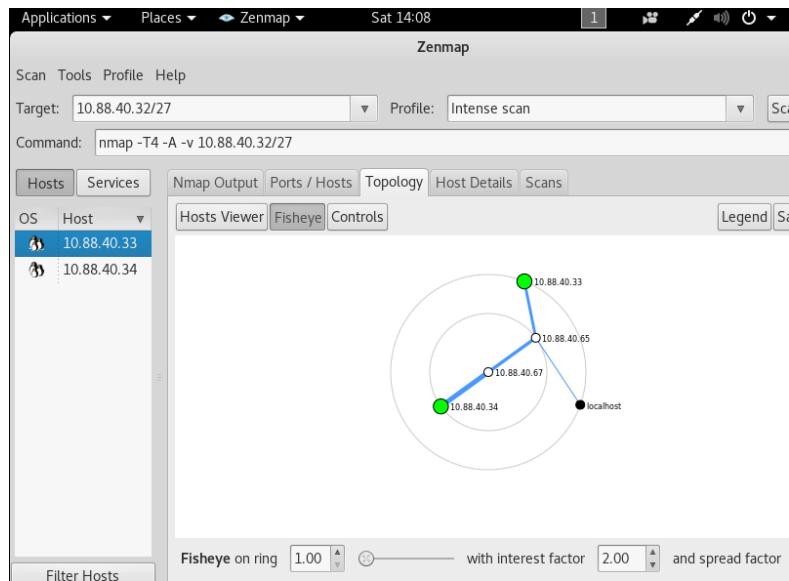
root@localhost:/scripts/ip_nmcli
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
ip addr add dev ens33 10.88.40.34/27
ip route add 10.88.40.64/27 via 10.88.40.33
ip route add 10.88.40.96/27 via 10.88.40.33
ip route add 10.88.40.128/27 via 10.88.40.33
iptables -F

```

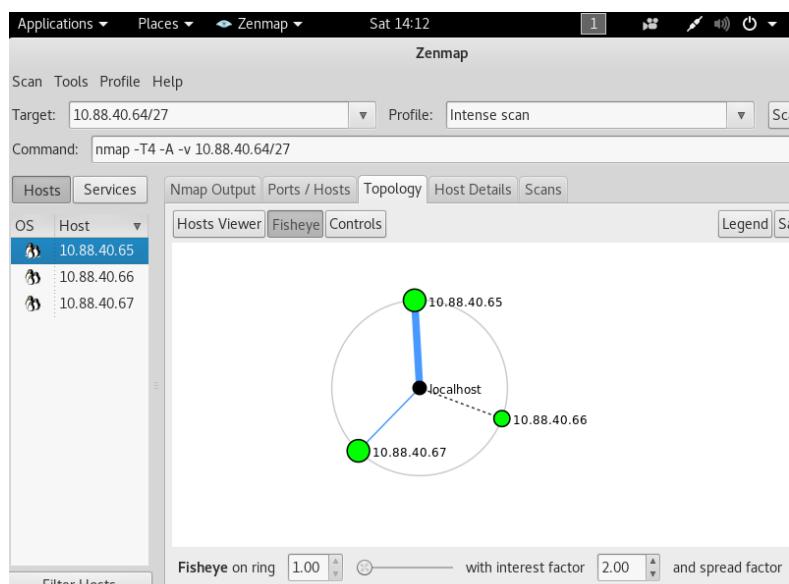
Abbildung 49: Konfig. Server Hamburg ip addr

Zenmap Ergebnisse der gescannten Subnetze:

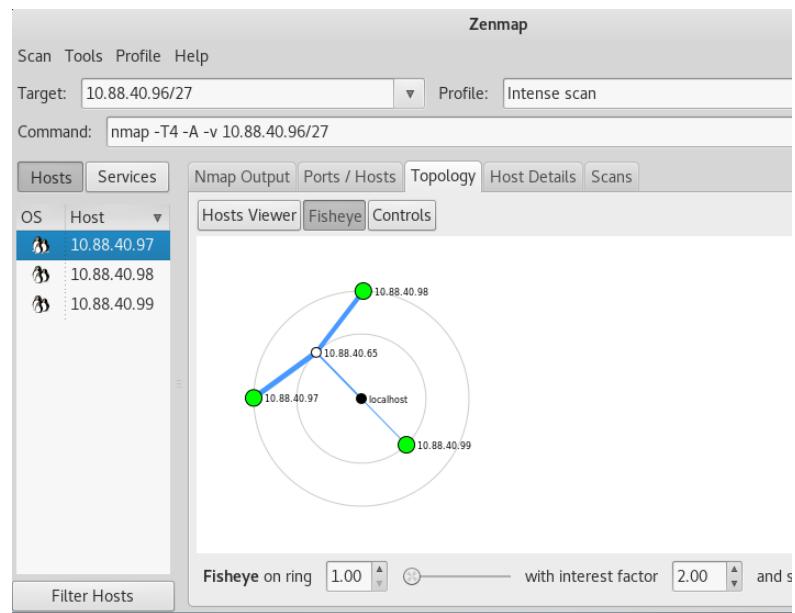
Scan des Subnetzes 10.88.40.32/27



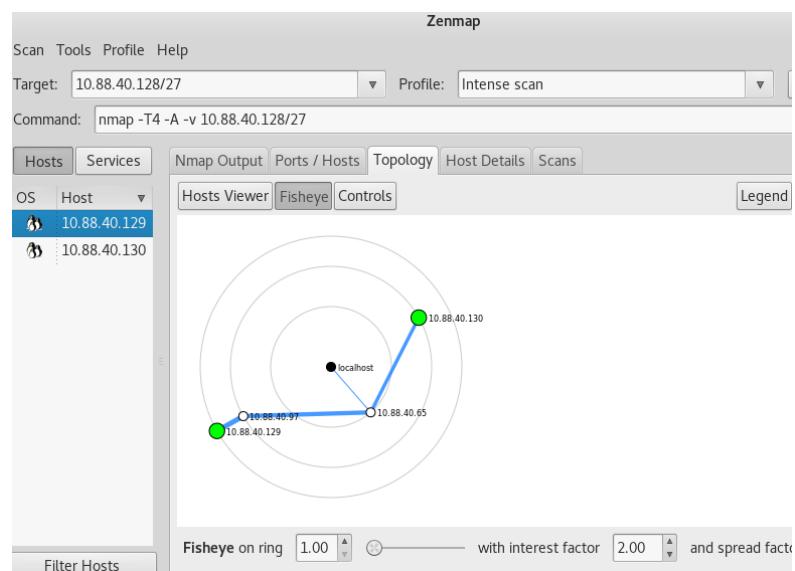
Scan des Subnetzes 10.88.40.64/27



Scan des Subnetzes 10.88.40.96/27



Scan des Subnetzes 10.88.40.128/27

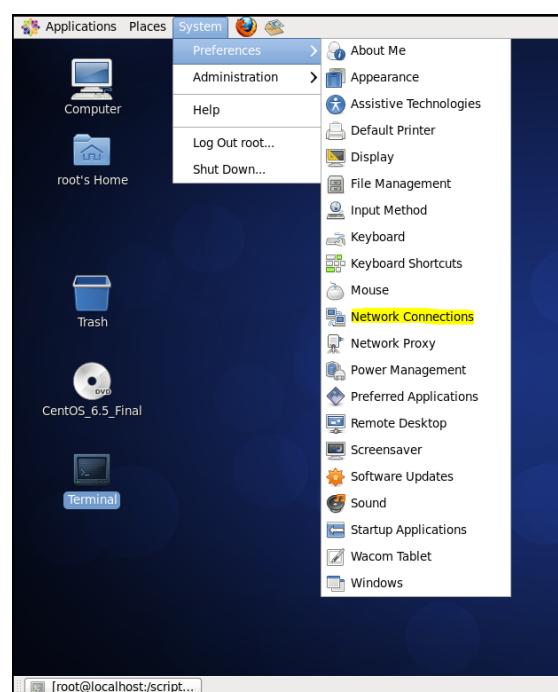


Exercise 9: Configure the following network (figure 1) using GUI

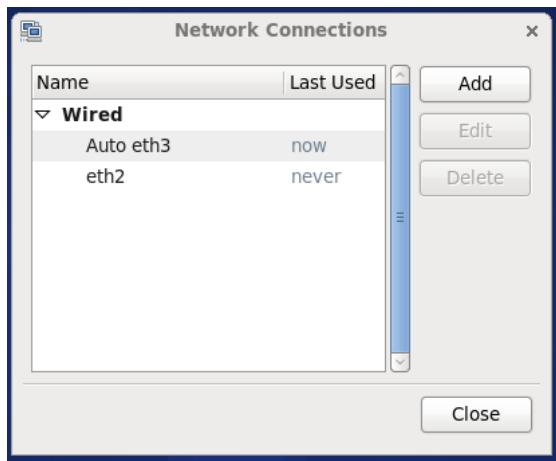
Damit die Konfiguration problemlos gelingen kann, sollte auf jedem Host, im Terminal, mit dem Kommando `iptables -F` die Firewall deaktiviert werden.

Konfiguration von Server München über die GUI:

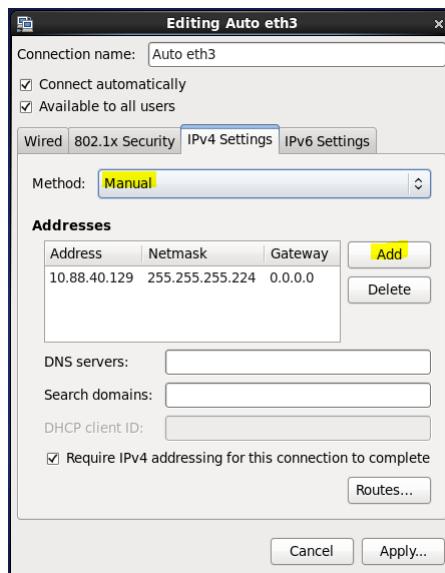
Wir starten die Netzwerkkonfiguration auf dem Host `pnid4-svr-mu` mit dem Betriebssystem CentOS 6. Zunächst gehen wir im Auswahlmenü auf System -> Preferences und wählen Network Connections aus.



Es öffnet sich ein Fenster in der wir die Bezeichner alle verfügbaren Netzwerk Interfaces sehen. Wir Wählen Auto eth3 aus und bestätigen mit Edit. Mit Add lässt sich ein neues Interface einrichten.

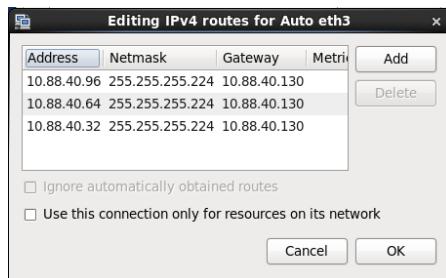


Wir können nun unser bereits vorhandenes Netzwerk Interface Auto eth3 bearbeiten. Dazu Wählen wir in der Navigation IPv4 Settings aus und erhalten folgende Ansicht. Falls Auto ausgewählt ist, ändern wie dieses auf Manual, damit wir eine statische IP Adresse eintragen können. Wir können im Bereich Addresses mit Add eine statische IPv4 Adresse für den Host eintragen. Da wir auf pnid4-svr-mu, vergeben wir die Adresse 10.88.40.129 mit der Subnetzmaske 255.255.255.224. Zuletzt müssen wir noch die jeweiligen Subnetze Eintragen und das Gateway über das wir die anderen Hosts erreichen. Dazu bestätigen wir den Button Routes.

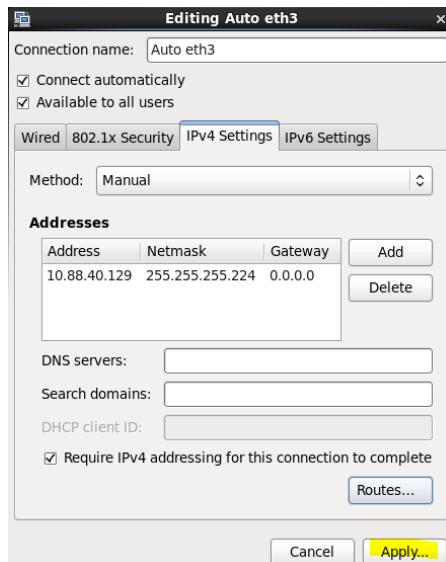


Der Button Routes, öffnet ein neues Fenster. Mit Add können wir nun die Subnetze, dessen Subnetzmaske und das Gateway unseres Edge Routers (Router 3) eintragen.

Da wir von pnid4-svr-mu über Router 3 alle anderen Subnetze erreichen können ist das Gateway für alle Subnetze gleich und lautet 10.88.40.130. Abschließend bestätigen wir mit OK.



Damit alle Änderungen für diesen Host übernommen werden bestätigen wir noch mal mit Apply.



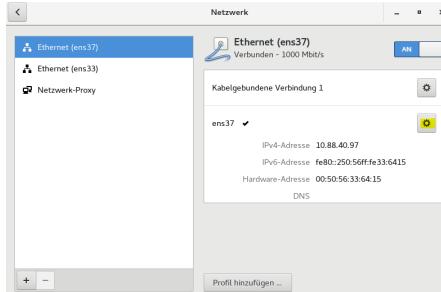
Das Netzwerk ist für den Host pnid4-svr-mu nun erfolgreich eingerichtet.

Konfiguration von Router 3 über die GUI:

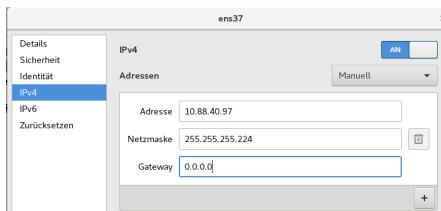
Router 1, 2 und 3 laufen mit dem Betriebssystem Linux Centos 7. Dementsprechend ist die Netzwerkkonfiguration auf allen Routern die gleiche. Wir erläutern die Konfiguration von Router 3 ausführlich und zeigen dann, welche IP Adressen Router 2 und Router 1 besitzen.

Wir Wählen zuerst unter dem Menüpunkt Anwendungen -> Systemsteuerung die Option Einstellungen aus. Im Auswahlmenü der Systemsteuerung wählen wir Netzwerk

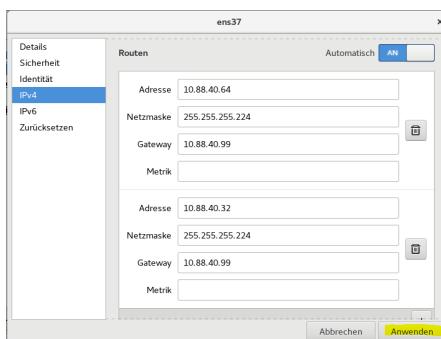
aus. Wir erhalten nun eine Netzwerkansicht bei der wir alle Netzwerk Interfaces ansehen können. Zurzeit existieren die Interfaces ens37 und ens33, da wir sie bereits beim Kopieren des Betriebssystems in der VMware Workstation festgelegt haben.



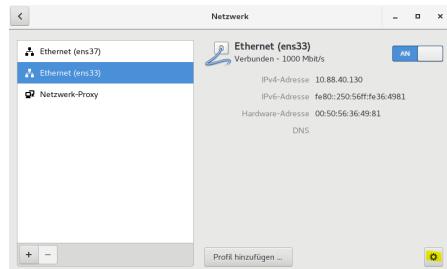
Mit dem Einstellungssymbol können wir die Schnittstellen konfigurieren. Wir wählen also ens37 aus und bestätigen mit Einstellungssymbol. Danach Navigieren wir an der linken Seite zu dem Namen IPv4. Falls noch nicht ausgewählt, stellen wir Adressen auf Manuell, damit wir eine statische IP Adresse vergeben können. Nun tragen wir die IP Adresse für die erste Schnittstelle des Routers ein. Als Gateway wählen wir 0.0.0.0 aus.



In der gleichen Ansicht wie zuvor scrollen wir etwas runter bis zum Abschnitt Routen und tragen nun die Adressen der Subnetze, die Subnetzmaske und das Gateway, über welches wir die Subnetze erreichen können, ein. Zuletzt bestätigen wir die Konfiguration mit Anwenden.



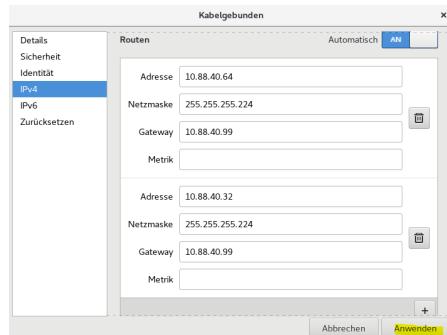
Damit haben wir das Interface mit dem Namen ens37 konfiguriert und müssen uns der zweiten Schnittstelle ens33 zuwenden. Wie eben können wir das Interface über den Button mit dem Einstellungssymbol konfigurieren.



Dieses Mal tragen wir die zweite statische IP Adresse für die zweite Schnittstelle ens33 ein.



Analog zum vorherigen Schritt scrollen wir runter und tragen die IP Adressen für die Subnetze, Subnetzmasken und das Gateway ein. Die Subnetze, Subnetzmasken und das Gateway sind bei ens33 und ens37 absolut identisch.

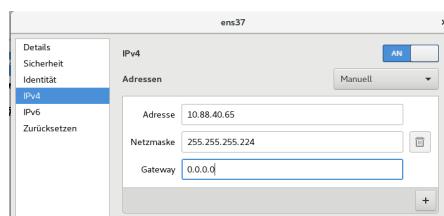


Zuletzt bestätigen wir die Konfiguration mit Anwenden. Der Router mit dem Namen pnid4-rou3 ist jetzt vollständig konfiguriert. Es folgen noch pnid4-rou2 (Router 2) und pnid4-rou1 (Router 1).

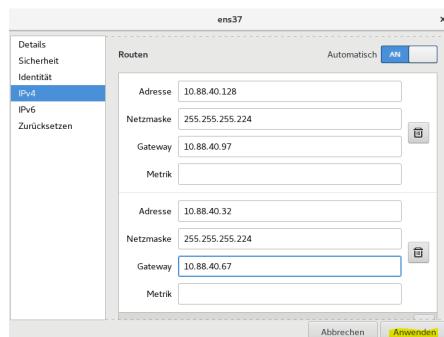
Konfiguration von Router 2 über die GUI:

Die Konfiguration auf allen Routern ist identisch. Lediglich die IP Adressen und das Routen unterscheidet sich, deshalb zeigen wir von nun an jeweils die Vergabe der statischen IP Adressen.

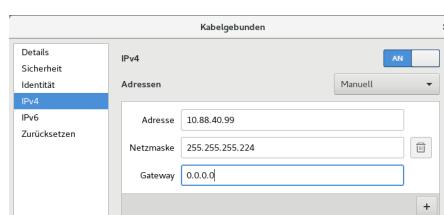
IP Adresse für das Interface ens37 festlegen:



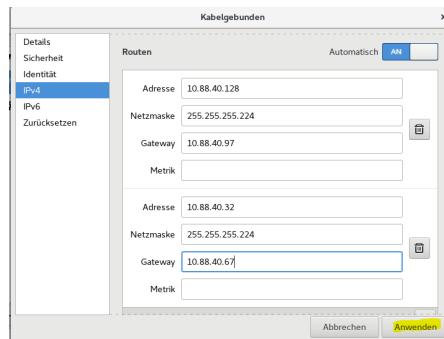
Adressen für die Subnetze festlegen und das jeweilige Gateway einstellen. Zuletzt mit Anwenden bestätigen.



IP Adresse für das Interface ens33 festlegen:



Adressen für die Subnetze festlegen und das jeweilige Gateway einstellen. Zuletzt mit Anwenden bestätigen.

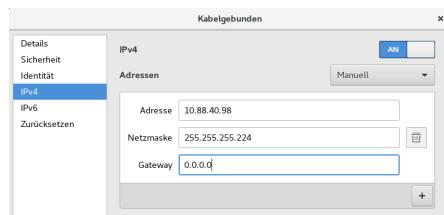


Damit ist die Konfiguration von Router 1 abgeschlossen.

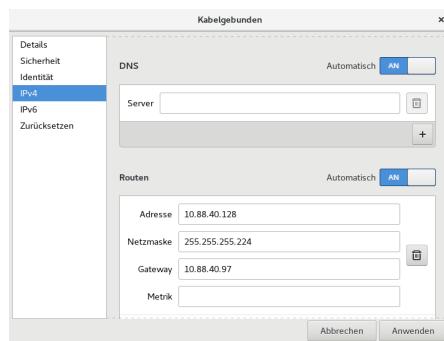
Konfiguration von Webserver Hannover über die GUI:

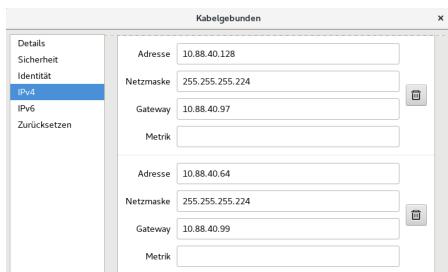
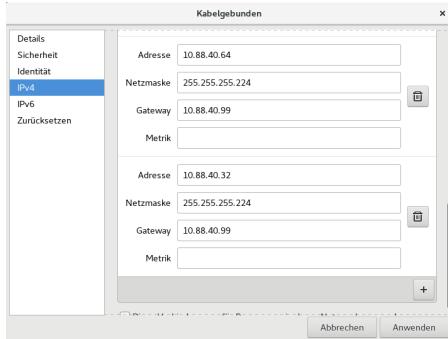
Dieser Host hat ebenfalls Linux Centos 7 als Betriebssystem, deshalb sind die Konfigurationsschritte dieselben wie bei den Routern.

IP Adresse für das Interface ens33 festlegen:



Adressen für die Subnetze festlegen, das jeweilige Gateway einstellen und die Routen eintragen.





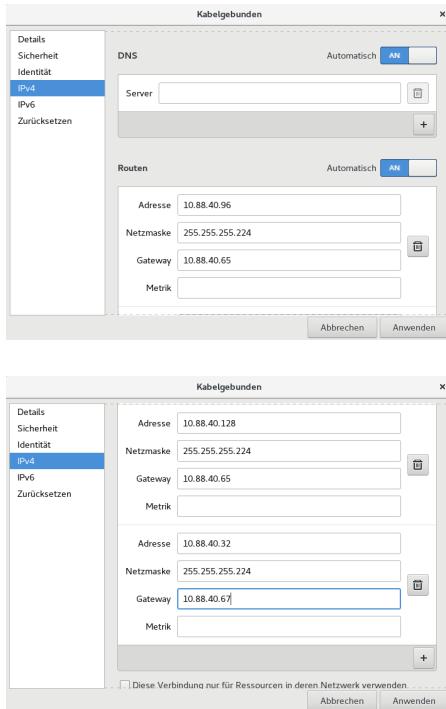
Zuletzt mit Anwenden bestätigen. Damit ist auch der Host pnid4-WEB-hn vollständig für die Subnetze konfiguriert.

Konfiguration von Server Berlin über die GUI:

IP Adresse für das Interface ens33 festlegen:



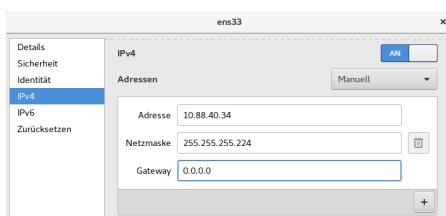
Adressen für die Subnetze festlegen, das jeweilige Gateway einstellen und die Routen eintragen.



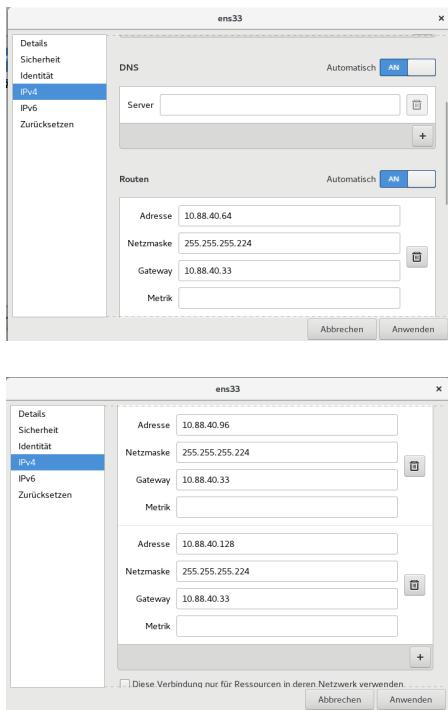
Zuletzt mit Anwenden bestätigen. Damit ist auch der Host pnid4-svr-bln vollständig für die Subnetze konfiguriert.

Konfiguration von Server Hamburg über die GUI:

IP Adresse für das Interface ens33 festlegen:



Adressen für die Subnetze festlegen, das jeweilige Gateway einstellen und die Routen eintragen.



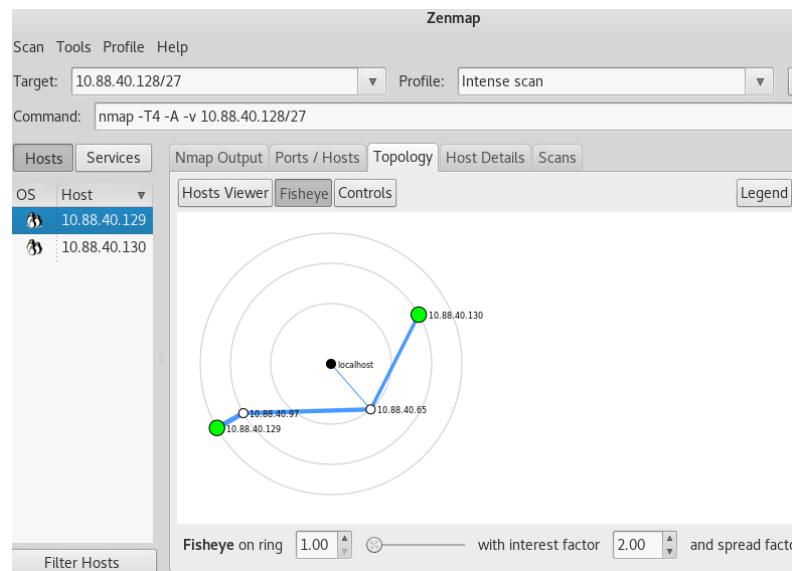
Zuletzt mit Anwenden bestätigen. Damit ist auch der Host pnid4-svr-hh vollständig für die Subnetze konfiguriert.

Zenmap Ergebnisse der gescannten Subnetze über die GUI:

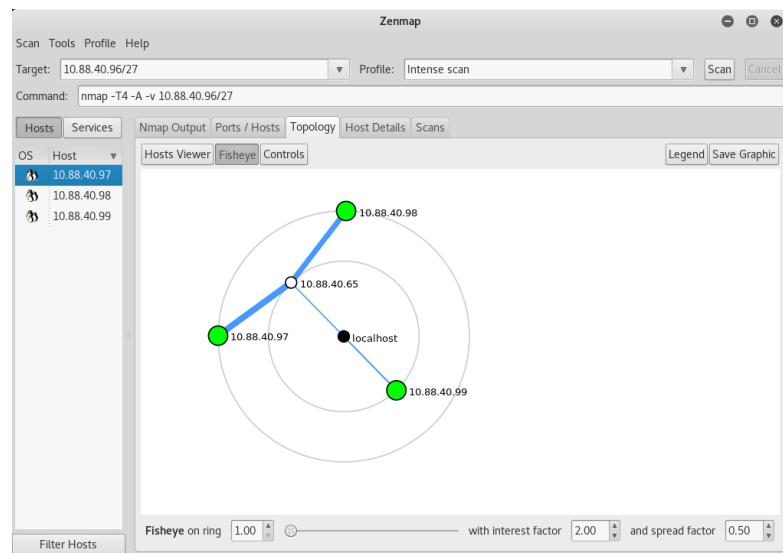
Alle Hosts sind jetzt in ihren jeweiligen Subnetzen konfiguriert. Um einen Visuellen Überblick über die Netzwerktopologie der einzelnen Subnetze zu erhalten, können wir mithilfe der Zenmap die einzelnen Netzwerke scannen. Dazu melden wir uns auf dem Host mit dem Betriebssystem Kali Linux an, welcher die IP Adresse 10.88.40.69/27 hält und geben in das Terminal den Befehl # zenmap ein.

Scan des Subnetzes 10.88.40.64/27

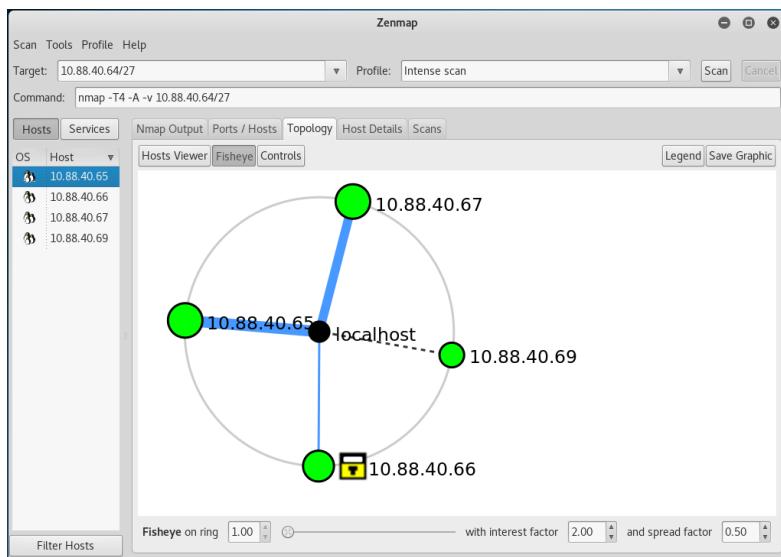
In der Suchleiste Target können wir nun die IP Adresse des Subnetzes eingeben, welchen wir scannen möchten. Wir starten mit dem Subnetz 10.88.40.128 und bestätigen den Vorgang mit Scan.



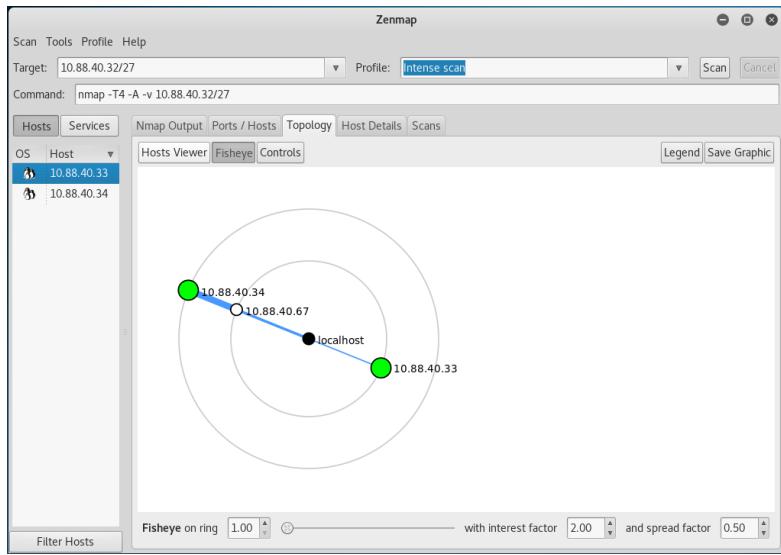
Scan des Subnetzes 10.88.40.96/27



Scan des Subnetzes 10.88.40.64/27



Scan des Subnetzes 10.88.40.32/27



Part 4: Network Scanning

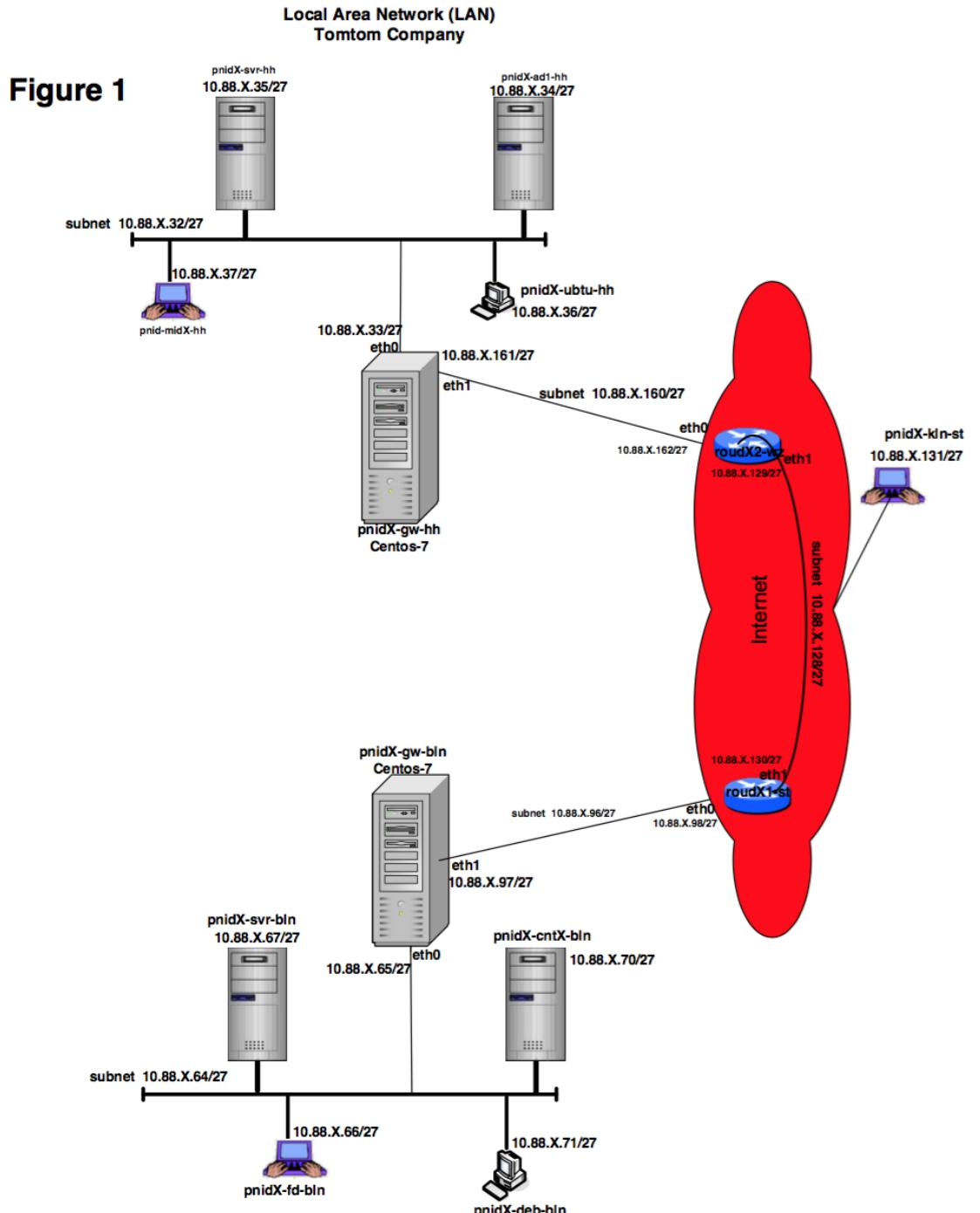


Abbildung 50: Netzwerk mit allen Hosts und Subnetzen

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 51: Subnetz 10.88.40.64/27

Scan des Subnetzes 10.88.40.96/27

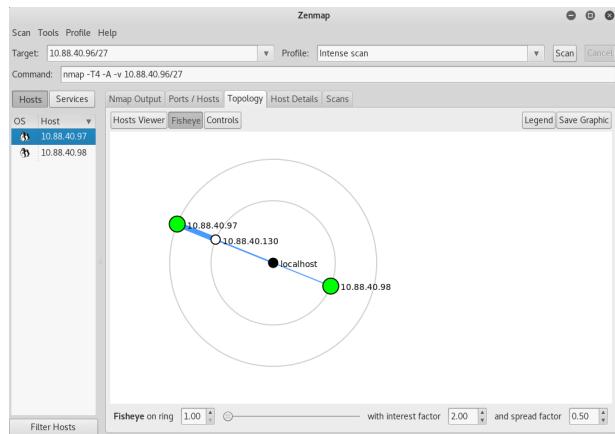


Abbildung 52: Subnetz 10.88.40.96/27

Scan des Subnetzes 10.88.40.128/27

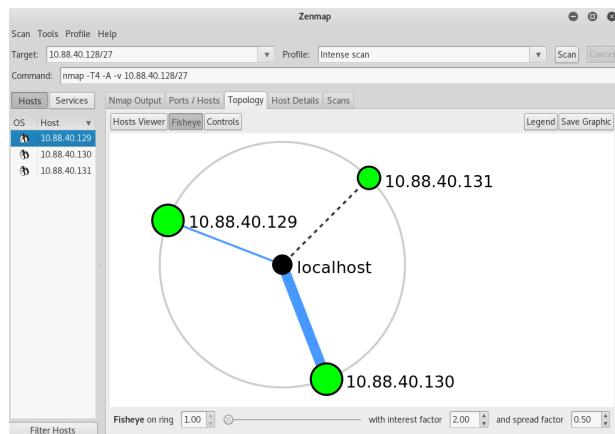


Abbildung 53: Subnetz 10.88.40.128/27

Scan des Subnetzes 10.88.40.160/27

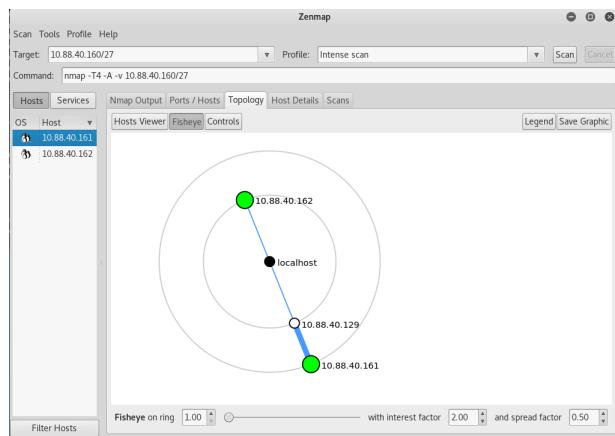


Abbildung 54: Subnetz 10.88.40.160/27

Scan des Subnetzes 10.88.40.32/27

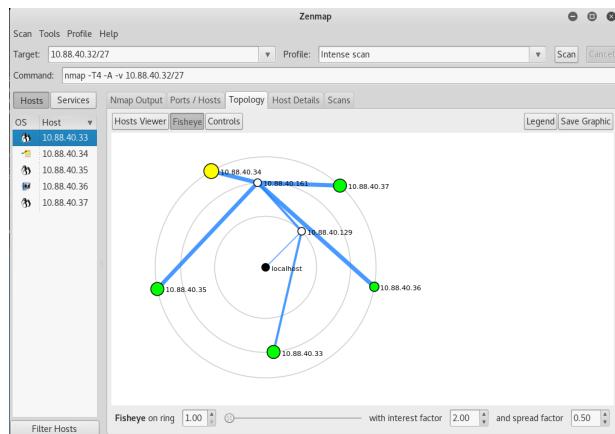


Abbildung 55: Subnetz 10.88.40.32/27

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

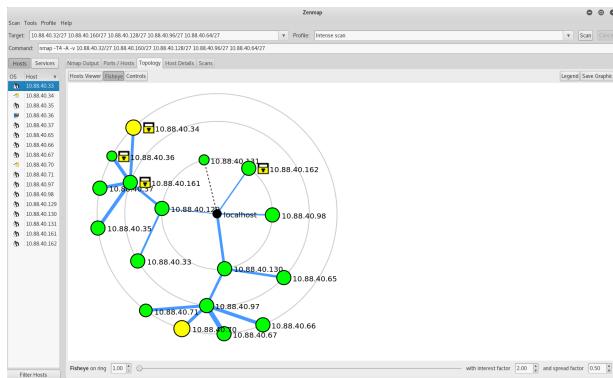


Abbildung 56: Alle Subnetze

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

Exercise 2: NMAP

Analyze your host system and your virtual machines with Nmap. Please test the following commands before explaining the meaning.

- Q1: Please type and explain the nmap command: nmap -sS -O 10.88.X.?
- Q2: Please type and explain the nmap command: nmap -sF 10.88.X.?-oN outfile
- Q3: Please type and explain the nmap command: nmap -sS 10.88.X.?-D 10.100.X.P
- Q4: Please type and explain the nmap command: nmap -sS -O 10.88.X.Y/Z
- Q5: Please type and explain the nmap command: nmap -sP -PS 10.88.X.?
- Q6: Please type and explain the nmap command: nmap -sP -PS25 10.88.X.?
- Q7: Please type and explain the nmap command: nmap -sP -PS80 10.88.X.?/Z
- Q8: Please type and explain the nmap command: nmap -sP -PS53 10.88.X.?/27
- Q9: Please type and explain the nmap command: nmap -sS -v 10.88.X.?
- Q10: Please type and explain the nmap command: nmap -sP -v 10.88.X.?

Question 1: Please type and explain the nmap command: nmap -sS -O 10.88.X.?

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 Verbindung aufgebaut und ist deshalb unauffälliger als das der Parameter -sT, welcher als einziger ohne root rechte Funktioniert. -O steht für OS-Detection. Es wird versucht, an besonderen Eigenarten der Netzwerkimplementierungen des Betriebssystems des Ziels zu identifizieren. Im Gegensatz zu Zenmap, wird nmap im Terminal durchgeführt und besitzt keine grafische Benutzeroberfläche um die Netzwerktopologie zu visualisieren.

```

root@pmid4-klnx1:~# nmap -sS -O 10.88.40.130
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:13 CET
nmap: 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
Device type: general purpose
Running: Linux 3.X|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@pmid4-klnx1:~#

```

Abbildung 57: nmap -sS -O 10.88.40.130

Question 2: Please type and explain the nmap command: nmap -sF 10.88.X.? -oN outfile

Answer 2: Bei diesem Scan wird das Subnetz 10.88.40.128/27 mit dem Argument -sF gescannt. -sF bezeichnet die Art des Scans bei der nur Pakete mit FIN-Flags zum Ziel Host gesendet werden. Durch das Argument -oN outfile erstellen wir ein externes Logfile, mit dem Namen outfile.

```

root@pmid4-klnx1:~# nmap -sF 10.88.40.128/27 -oN outfile
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 12:19 CET
nmap: 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 10.88.40.129
Host is up (0.00046s latency).
All 1000 scanned ports on 10.88.40.129 are open|filtered
MAC Address: 00:50:56:29:79:5E (VMware)

Nmap scan report for 10.88.40.130
Host is up (0.0016s latency).
All 1000 scanned ports on 10.88.40.130 are open|filtered
MAC Address: 00:50:56:39:6E:C2 (VMware)

Nmap scan report for 10.88.40.131
Host is up (0.000010s latency).
All 1000 scanned ports on 10.88.40.131 are closed

Nmap done: 32 IP addresses (3 hosts up) scanned in 43.88 seconds
root@pmid4-klnx1:~#

```

Abbildung 58: nmap -sF 10.88.40.128 -oN outfile

```

# Nmap 7.60 scan initiated Thu Nov 23 12:19:19 2017 as: nmap -sF -oN outfile 10.88.40.128/27
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 10.88.40.129
Host is up (0.00046s latency).
All 10000 scanned ports on 10.88.40.129 are open|filtered
MAC Address: 00:50:56:29:79:5E (VMware)

Nmap scan report for 10.88.40.130
Host is up (0.0016s latency).
All 10000 scanned ports on 10.88.40.130 are open|filtered
MAC Address: 00:50:56:39:6E:C2 (VMware)

Nmap scan report for 10.88.40.131
Host is up (0.000010s latency).
All 10000 scanned ports on 10.88.40.131 are closed

# Nmap done at Thu Nov 23 12:20:03 2017 -- 32 IP addresses (3 hosts up) scanned in 43.88 seconds

```

Abbildung 59: externes Logfile

Mit dem Befehl `-sF` werden die Ports, die gescannt werden manipuliert, in dem verfälschte TCP-Pakete versendet werden. Dadurch erhält man die Information, ob ein Port offen oder von einer Firewall geschützt ist. Die Ausgabe wird im Terminal angezeigt, sowie durch das Argument `-oN outfile`, in einer separaten `outfile.txt` Datei.

Question 3: Please type and explain the nmap command: Please type and explain the nmap command: `nmap -sS 10.88.X.? -D 10.100.X.P`

Answer 3: Dieser Befehl führt einen Decoy-Scan durch. Mit dem Argument `-D 10.88.40.36` legen wir einen Köder aus, mit dem wir den Ziel Host / Subnetz scannen. Diese Methode wird verwendet um die eigene IP Adresse zu verbergen, jedoch sollte der Host, welcher als Köder benutzt wird eingeschaltet sein.

```

root@pnid4-klnx1:~# nmap -sS 10.88.40.130 -D 10.88.40.36
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:02 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 (0.00050s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:50:56:39:6E:C2 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 4.73 seconds
root@pnid4-klnx1:~#

```

Abbildung 60: `nmap -sS 10.88.40.130 -D 10.88.40.36`

Question 4: Please type and explain the nmap command: `nmap -sS -O 10.88.X.Y/Z`

Answer 4: Der Befehl versucht alle erreichbaren Hosts im Netzwerk X, mit ihrer IP Adresse anzuzuzeigen. Zusätzlich wird `-sS` (SYN-Stealth-Scan) und `-O` (OS-Detection) verwendet.

```

root@pnid4-klnx1:~# nmap -sS -O 10.88.40.160/27
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 12:26 CET
Nmap scan report for 10.88.40.160
Host is up (0.00075s latency).
All 1080 scanned ports on 10.88.40.160 are filtered
Too many fingerprints match this host to give specific OS details

Nmap scan report for 10.88.40.162
Host is up (0.00075s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
Warning: OSscan results may be unreliable because we could not find at least 1 open and 1 closed port
Detailed info not available for this host
Running: Linux 3.X|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

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 32 IP addresses (2 hosts up) scanned in 26.72 seconds

```

Abbildung 61: nmap -sS -O 10.88.40.160/27

Question 5: Please type and explain the nmap command: nmap -sP -PS 10.88.X.?

Answer 5: Das Argument -sP ist der sogenannte Ping-Scan. Es werden alle Hosts ausgegeben, welche auf den Scan geantwortet haben. So kann die Verfügbarkeit eines Rechners im Netzwerk gezählt werden, sowie die Server-Verfügbarkeit überwacht werden. Das Argument -PS sendet ein leeres TCP-Paket mit gesetzten SYN-Flag. Ein SYN-Flag ist ein Synchronisations-Flag, bestehend aus einem Bit. Ist dieser Flag gesetzt, will der Sender eine Verbindung zum Empfänger aufbauen.

```

root@pnid4-klnx1:~# nmap -sP -PS 10.88.40.130
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:17 CET
Nmap scan report for 10.88.40.130
Host is up (0.00069s latency).
MAC Address: 00:50:56:39:6E:C2 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
root@pnid4-klnx1:~#

```

Abbildung 62: nmap -sP -PS 10.88.40.130

Question 6: Please type and explain the nmap command: nmap -sP -PS25 10.88.X.?

Answer 6: Mit dem Argument -sP wird die Ping-Scan Methode ausgewählt. -PS wird verwendet um SYN-Pakete, mit gesetztem SYN-flag über den Port 25 (SMTP) zu senden.

```

root@pnid4-klnx1:~# nmap -sP -PS25 10.88.40.130
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:49 CET
Nmap scan report for 10.88.40.130
Host is up (0.00076s latency).
MAC Address: 00:50:56:39:6E:C2 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
root@pnid4-klnx1:~#

```

Abbildung 63: nmap -sP -PS25 10.88.40.130

Question 7: Please type and explain the nmap command: nmap -sP -PS80 10.88.X.?/Z

Answer 7: Mit dem Argument -sP wird die Ping-Scan Methode ausgewählt. -PS wird verwendet um SYN-Pakete, mit gesetztem SYN-flag über den Port 80 zu senden. Port 80 ist zuständig für den Hypertext Transfer Protocol (HTTP). HTTP verwendet das TCP-Protokoll und benutzt diesen am Port 80.

```
root@pnid4-klnx1:~# nmap -sP -PS80 10.88.40.130/27
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:50 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid
servers with -sS or -sM.
Nmap scan report for 10.88.40.129
Host is up (0.0017s latency).
MAC Address: 00:50:56:29:79:5E (VMware)
Nmap scan report for 10.88.40.130
Host is up (0.0011s latency).
MAC Address: 00:50:56:39:6E:C2 (VMware)
Nmap scan report for 10.88.40.131
Host is up.
Nmap done: 32 IP addresses (3 hosts up) scanned in 0.79 seconds
root@pnid4-klnx1:~#
```

Abbildung 64: nmap -sP -PS80 10.88.40.130/27

Question 8: Please type and explain the nmap command: nmap -sP -PS53 10.88.X.?/27

Answer 8: Hier wird ebenso ein TCP SYN-Scan auf dem Port 53 durchgeführt. Port 53 wird verwendet für das Domain Name System (DNS) und wird meist über UDP verwendet.

```
root@pnid4-klnx1:~# nmap -sP -PS53 10.88.40.130/27
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:51 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid
servers with -sS or -sM.
Nmap scan report for 10.88.40.129
Host is up (0.0017s latency).
MAC Address: 00:50:56:29:79:5E (VMware)
Nmap scan report for 10.88.40.130
Host is up (0.0013s latency).
MAC Address: 00:50:56:39:6E:C2 (VMware)
Nmap scan report for 10.88.40.131
Host is up.
Nmap done: 32 IP addresses (3 hosts up) scanned in 0.79 seconds
root@pnid4-klnx1:~#
```

Abbildung 65: nmap -sP -PS53 10.88.40.130/27

Question 9: Please type and explain the nmap command: nmap -sS -v 10.88.X.?/27

Answer 9: -sS-Der SYN-Scan ist eine Methode fürs schnelle Scannen und scannt dabei Tausende von Ports pro Sekunde, wenn es nicht von einer Firewall gestört wird. Der Syn-Scan schließt die TCP-Verbindungen nicht ab. Außerdem kann zwischen den Zuständen offen, geschlossen und gefiltert unterschieden werden. Da keine vollständigen TCP-Verbindungen hergestellt werden, wird dies auch als halboffenes Scannen

bezeichnet. Ein SYN-Paket wird gesendet. Dann wird auf eine Antwort gewartet. Ein SYN/ACK gibt an, dass jemand auf dem Port lauscht. Dies ist an einem offenen Port erkennbar. RST jedoch bedeutet, dass der Port geschlossen ist. -v bedeutet, dass die Ausführlichkeitsstufe erhöht wird(verbosity-Level), um mehr Wirkung zu erzielen.

```
root@pnid4-klnx1:~# nmap -sS -v 10.88.40.130
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:52 CET
Initiating ARP Ping Scan at 13:52
Scanning 10.88.40.130 [1 port]
Completed ARP Ping Scan at 13:52, 0.22s elapsed (1 total hosts)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid
servers with --dns-servers
Initiating SYN Stealth Scan at 13:52
Scanning 10.88.40.130 [1000 ports]
Discovered open port 22/tcp on 10.88.40.130
Completed SYN Stealth Scan at 13:52, 14.92s elapsed (1000 total ports)
Nmap scan report for 10.88.40.130
Host is up (0.00044s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:50:56:39:6E:C2 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 15.28 seconds
          Raw packets sent: 2981 (131.148KB) | Rcvd: 23 (1.556KB)
root@pnid4-klnx1:~#
```

Abbildung 66: nmap -sS -v 10.88.40.130

Question 10: Please type and explain the nmap command: nmap -sP -v 10.88.X.?

Answer 10: Ein Ping-Scan wird ausgeführt mit erhöhtem Verbosity Level.

```
root@pnid4-klnx1:~# nmap -sP -v 10.88.40.130
Warning: The -sP option is deprecated. Please use -sn
Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-23 13:53 CET
Initiating ARP Ping Scan at 13:53
Scanning 10.88.40.130 [1 port]
Completed ARP Ping Scan at 13:53, 0.23s elapsed (1 total hosts)
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 (0.00066s latency).
MAC Address: 00:50:56:39:6E:C2 (VMware)
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
          Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
root@pnid4-klnx1:~#
```

Abbildung 67: nmap -sP -v 10.88.40.130

Exercise 3: Nessus network device identification

Nessus is a scanner program.

It includes following features:

- a) It is a vulnerability scanner
- b) It is a port scanner

- c) It is a host/device detection program
- d) It can be used to scan Netbios Servers e.g. Windows Servers and Samba Servers
- e) Nessus can be used as a penetrating testing tool
- f) It is a client-server-system. The server performs the actual scan but it is controlled through the client. Both client and server can be run on the same system

In this exercise you will download, install and configure Nessus-6.11.2-es7.x86_64.rpm (use the RPM from your instructor) Do the following steps to install and run nessus:

- g. # Go to the nessus website and register your product for home feed means free cost
<https://www.tenable.com/products/nessus/select-your-operating-system#tos> h. # rpm -Uvh Nessus-6.11.2-es7.x86_64.rpm on Centos 7
- i. # After registration open your email and copy the activation code and type your activation key
- j. Now open a web browser and type the following: <https://10.88.X.P:8834/>
 Enter your login name and password that you enter while installation. Note: download the user manual to obtain more information or refer to the internet. Then you will perform a scan on all subnet as below and analyze the result. Exercise:

- 1.) Download Nessus from www.nessus.org (Linux Version Nessus-6.11.2 - es7.x86_64.rpm)
- 2.) Install the Nessus-6.11.2-es7.x86_64.rpm Binary on your Centos7 Virtual Machine
- 3.) Register Nessus to obtain the plugins (note: choose the offline method)
- 4.) Install the plugins
- 5.) Perform a host identification of the localhost
- 6.) 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 using nessus
- 7.) Perform a network device identification on your subnet 10.88.X.P/27, see figure 1

Einleitung: Nessus ist ein Netzwerk- und Vulnerability Scanner. Unter einem Vulnerability Scanner versteht man Computerprogramme. Diese sind dafür zuständig Zielsysteme auf Sicherheitslücken bzw. Schwachstellen zu untersuchen. Der Scanner hat somit Zugriff auf entsprechende Datenbanken, um Informationen zu Sicherheitsproblemen zu bekommen. Dazu gehören der Einsatz bzw. das Vorhandensein von unsicheren oder nicht benötigten Diensten, Fehler in der Konfiguration bzw. Anwendung von Passwort-

und Benutzerrichtlinien sowie offene Ports.

Nessus beruht auf einem Client-Server-Prinzip. Hierbei wird auf einem Rechner der Nessusserver gestartet und im Anschluss wird eine Verbindung zu anderen Rechnern hergestellt. Sobald der Server gestartet wird, werden die Plug-ins geladen. Diese sind notwendig, da man Sicherheitslücken des Betriebssystems finden kann während des Scans eines Hostes. Nessus kann als Penetrationstest verwendet werden. Darunter versteht man Sicherheitstests eines Rechners oder von Netzwerken. Dabei wird die Sicherheit der Systembestandteile und Anwendungen eines Netzwerks überprüft. Die Mittel, die dafür verwendet werden sind Methoden, die ein Angreifer verwenden würden, um unautorisiert in das System einzudringen, weshalb dies Penetration bezeichnet wird. Man möchte somit vor Angriffen schützen.

Solution of 1) Download Nessus from www.nessus.org (Linux Version Nessus-6.11.2-es7.x86_64.rpm)

Zunächst geht man zur folgenden Webseite: <https://www.tenable.com/products/nessus/select-your-operating-system>.

Anschließend wird die entsprechende Datei für CentOS 7 runtergeladen. Diese ist: Nessus-6.11.3-es7.x86_64.rpm. Nach dem Runterladen muss man sich noch registrieren, damit man einen Aktivierungs Code bekommt. Diesen erhält man per Email. Nach dem Herunterladen wird die Binary, wie man unten im Bild sieht, in die virtuelle Maschine reinkopiert.

Solution of 2) Install the Nessus-6.11.2-es7.x86_64.rpm Binary on your Centos7 Virtual Machine

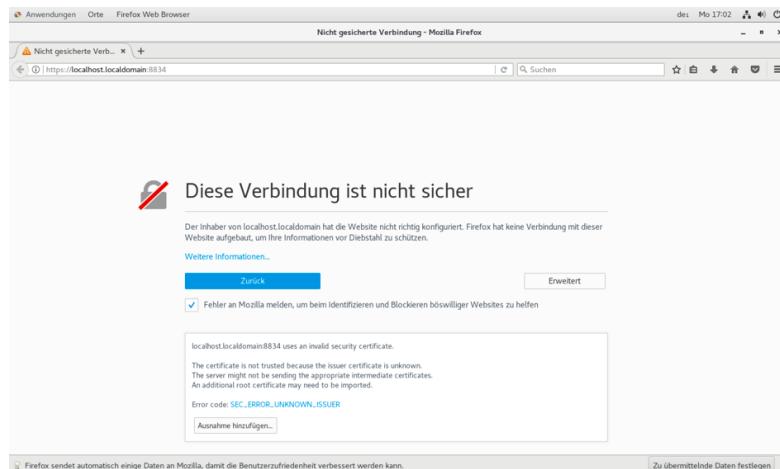
Um die Binary zu installieren wird der Befehl: rpm -Uvh /home/hanifka/Schreibtisch/Nessus-6.11.2-es7.x86_64.rpm im Terminal ausgeführt.

```
[root@localhost ~]# rpm -Uvh /home/hanifka/Schreibtisch/Nessus-6.11.2-es7.x86_64.rpm
Warning: /home/hanifka/Schreibtisch/Nessus-6.11.2-es7.x86_64.rpm: Header V4 RSA/SHA1 Signature, Schlüssel-ID 1c0cc4a5d: NOKEY
Vorbereiten...
Aktualisierung/ Installation...
1:Nessus-6.11.2-es7
Unpacking Nessus Core Components...
nessusd (Nessus) 6.11.2 [build M20102] for Linux
Copyright (C) 1990 - 2017 Tenable Network Security, Inc

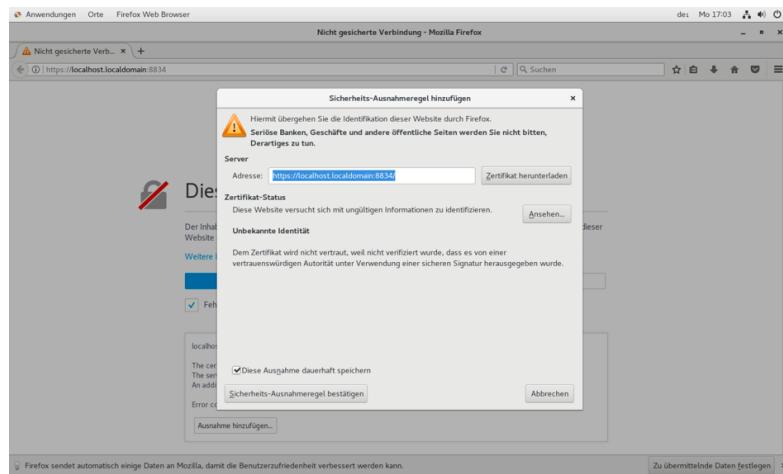
Processing the Nessus plugins...
[########################################] [100%]
All plugins loaded (1sec)
- You can start Nessus by typing /bin/systemctl start nessusd.service
- Then go to https://localhost.localdomain:8834/ to configure your scanner
[root@localhost ~]#
```

Abbildung 68: Installation von Nessus im Terminal

Nach erfolgreicher Installation kann Nessus mit folgendem Befehl gestartet werden: /bin/systemctl start nessusd.service. Anschließend wird der Browser geöffnet und man gibt folgenden Link ein: <https://localhost.localdomain:8834/>, um den Scanner konfigurieren zu können.



Damit dies funktioniert muss man auf den Button, Ausnahme hinzufügen, klicken. Danach muss man die Sicherheits-Ausnahmeregel bestätigen. Auch auf diesen Button wird geklickt.



Jetzt kann mit der Konfiguration gestartet werden.

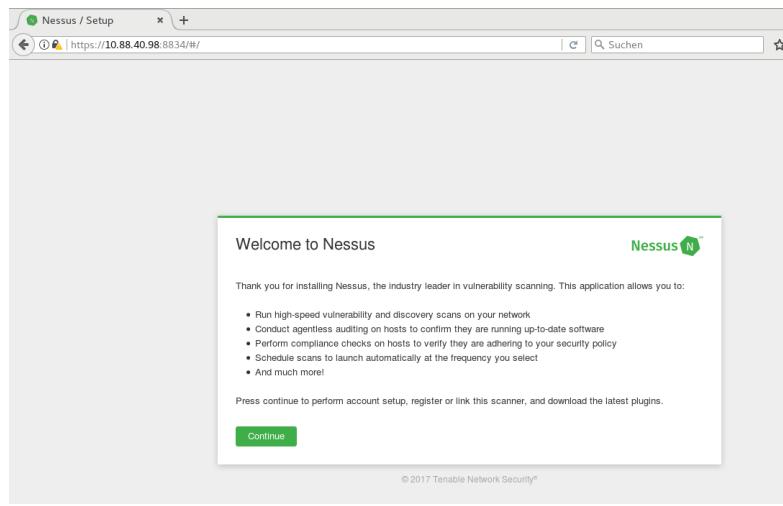


Abbildung 69: Nessus Konfiguration Startseite

Solution of 3) Register Nessus to obtain the plugins (note: choose the offline method)

Damit man sich erfolgreich registrieren kann, wählt man die offline Methode. Dann erhält man einen challenge code, den man fürs weitere Vorgehen benötigen wird.

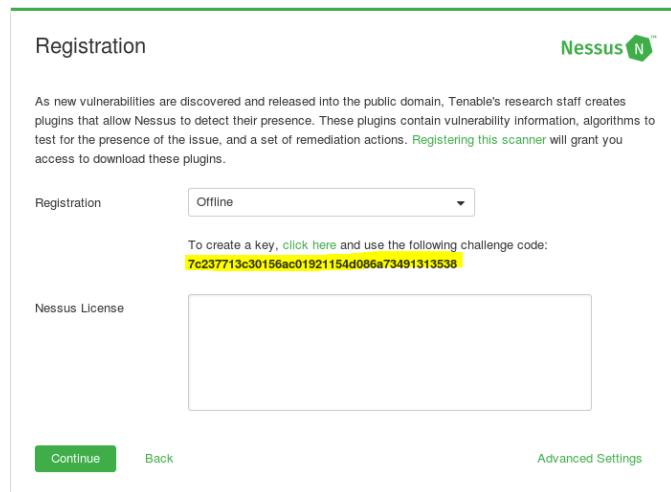


Abbildung 70: Nessus Registrierung

Im Browser wird nun folgender Tab geöffnet. Hier wird zuerst unser zu eben erzeugter challenge code eingegeben und unten wird der der Aktivierungs Code eingegeben, den man per Email nach der Registrierung erhalten hat.

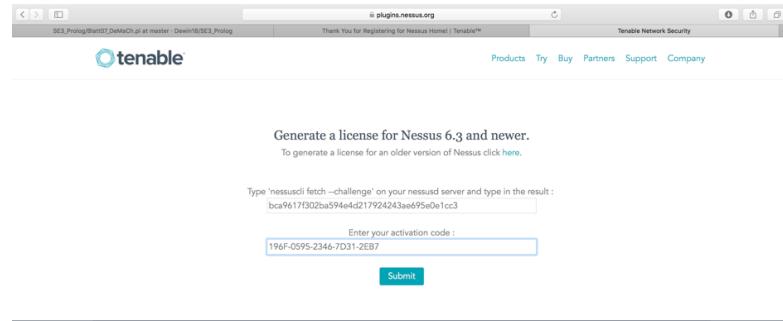


Abbildung 71: Nessus Aktivierungskey

Nach dem Klicken auf Submit erhalten wir unsere Nessus License.

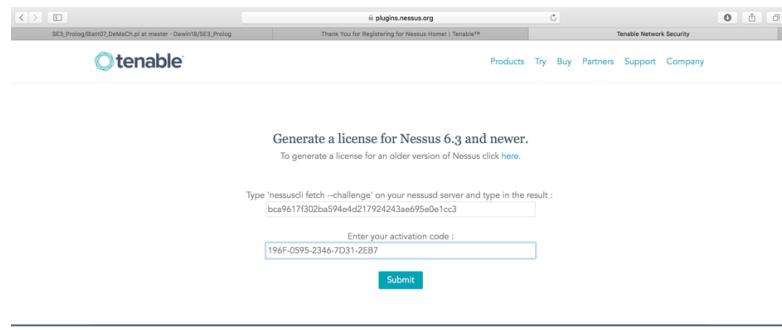
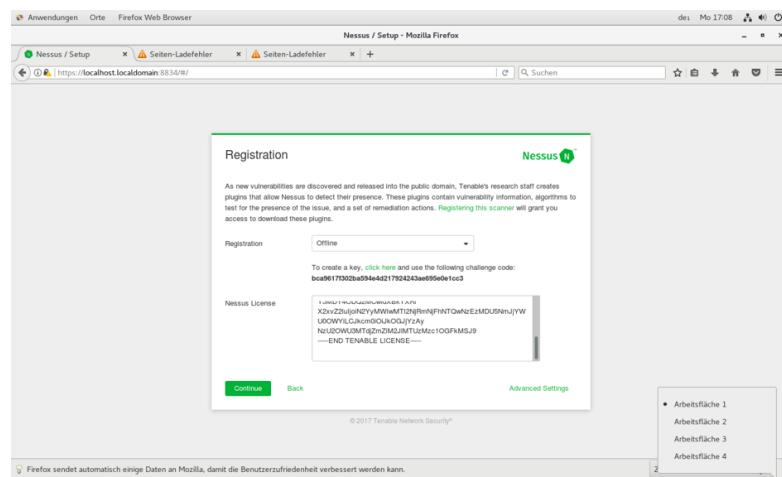
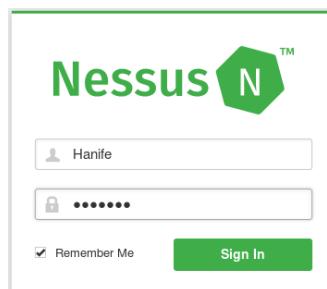


Abbildung 72: Nessus License

Die Nessus License kopieren wir und fügen diese ein, da sie für die erfolgreiche Registrierung notwendig ist.



Nach dem alles erfolgreich konfiguriert wurde und man sich erfolgreich registriert hat, kann man sich anmelden.

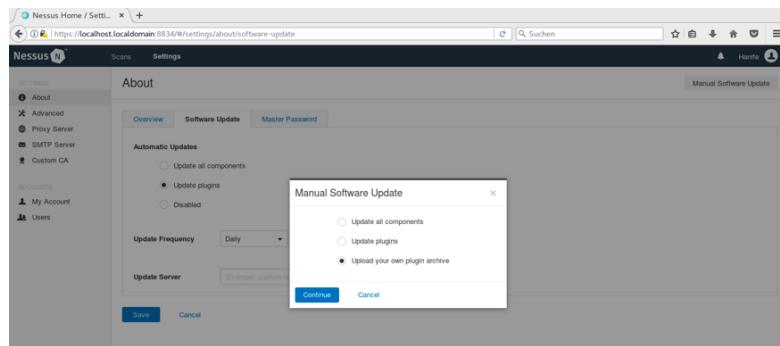


Solution of 4) Install the plugins

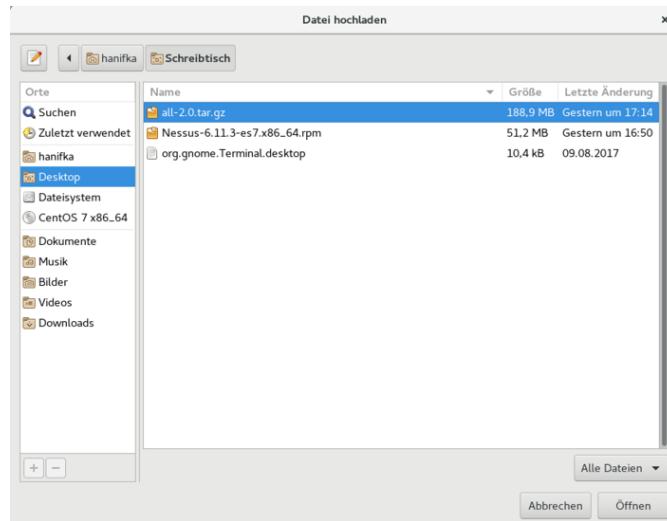
Um die Plugins zu installieren, laden wir die notwendige Datei all-2.0.tar.gz runter und kopieren diese anschließend in die virtuelle Machine. Dort kann sie wie folgt installiert werden übers Terminal oder manuell über die GUI.

```
[root@localhost sbin]# /opt/nessus/sbin/nessuscli update /home/hanifka/Schreibtisch/all-2.0.tar.gz
* Update successful. The changes will be automatically processed by Nessus.
[root@localhost sbin]#
```

Zunächst melden wir uns bei Nessus an. Unter dem Reiter Software Update können wir über den Button Manual Software Update unsere Datei all-2.0.tar.gz mit den ganzen Plugins reinladen.



Hier sieht man, dass die Datei all-2.0.tar.gz ausgewählt wird und anschließend geladen werden kann.



Solution of 5) Perform a host identification of the localhost and **Solution of 6)** Please scan the following networks: 10.88.40.64/27, 10.88.40.96/27, 10.88.40.128/27, 10.88.40.160/27 and 10.88.40.32/27 using nessus

Das Netz 10.88.40.32/27 wird gescannt. Wir führen eine Host Discovery durch. Die Scans verlaufen alle sehr schnell. Unter einer Host-Discovery versteht man Methoden oder Maßnahmen, um die erreichbaren Rechner in einem Netzwerk zu identifizieren. Im Allgemeinen versucht man die Rechner zu ermitteln, die über eine IP-Adresse erreichbar sind. Zuerst versucht man alle Hosts in einem Netzwerk zu identifizieren und danach erfolgt eine differenzierte Untersuchung des entsprechenden Hosts. Dadurch erhält man Informationen über die Hardware, das Betriebssystem, laufende Dienste und offene Ports, welches man als Host-Scan bezeichnet.

Scan des Subnetzes 10.88.40.32/27:

Wir geben bei Targets das zugehörige Netz an, welches gescannt werden soll. Diesem können wir einen Namen und eine Beschreibung hinzufügen

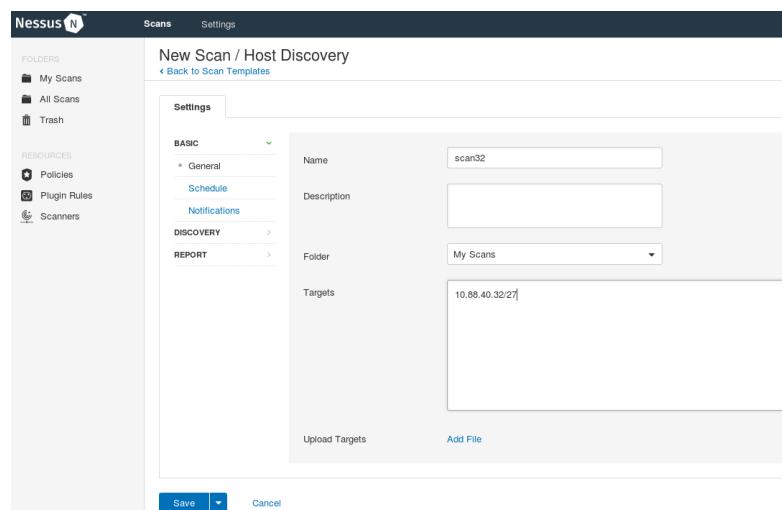


Abbildung 73: Scan des Subnetzes 10.88.40.32/27

Hier sieht man alle zugehörigen Netze, die zum Subnetz 10.88.40.32/27 gehören, die durch Host-Discovery ermittelt wurden.

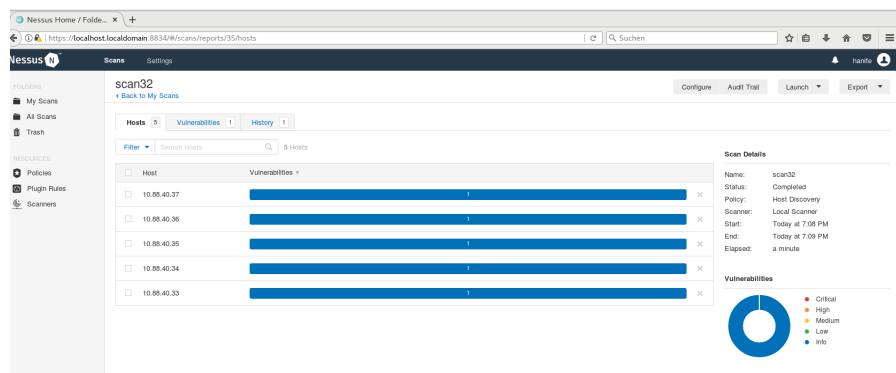


Abbildung 74: Alle Hosts im Subnetz 10.88.40.32/27

Scan des Subnetzes 10.88.40.64/27:

The screenshot shows the Nessus configuration interface for a new scan named '64'. The 'BASIC' tab is selected, showing fields for 'Name' (64), 'Description' (empty), 'Folder' (My Scans), and 'Targets' (10.88.40.64/27). There are also 'Upload Targets' and 'Add File' buttons at the bottom.

Abbildung 75: Scan des Subnetzes 10.88.40.64/27

Hier sieht man alle zugehörigen Netze, die zum Subnetz 10.88.40.64/27 gehören, die durch Host-Discovery ermittelt wurden.

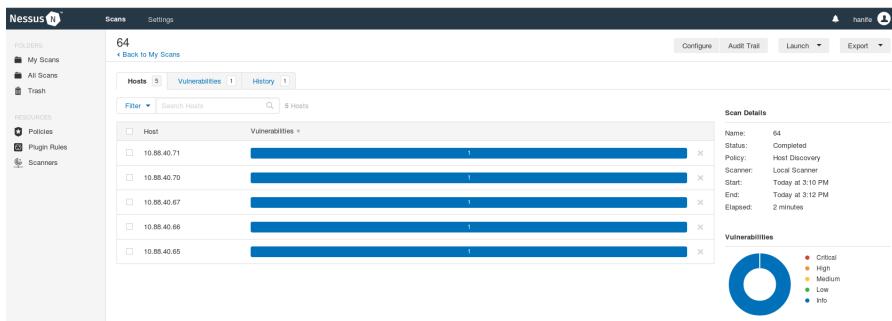


Abbildung 76: Alle Hosts im Subnetz 10.88.40.64/27

Scan des Subnetzes 10.88.40.96/27:

New Scan / Host Discovery
Back to Scan Templates

Settings

BASIC

- General
- Schedule
- Notifications

DISCOVERY

REPORT

Name: 96

Description:

Folder: My Scans

Targets: 10.88.40.96/27

Upload Targets Add File

Save Cancel

Abbildung 77: Scan des Subnetzes 10.88.40.96/27

Hier sieht man alle zugehörigen Netze, die zum Subnetz 10.88.40.96/27 gehören, die durch Host-Discovery ermittelt wurden.

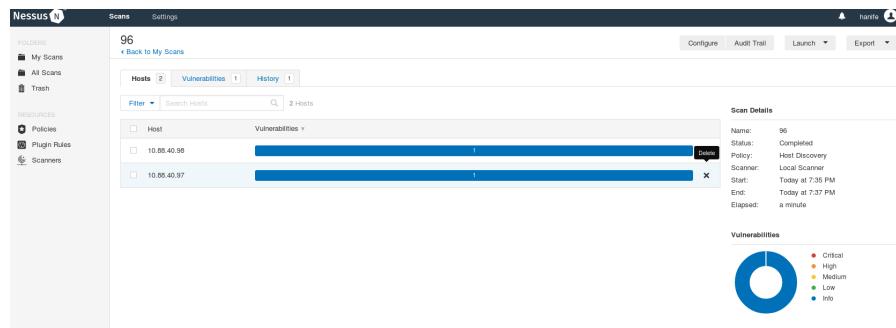


Abbildung 78: Alle Hosts im Subnetz 10.88.40.96/27

Scan des Subnetzes 10.88.40.128/27:

The screenshot shows the Nessus N configuration page for a new scan named '128'. The 'Settings' tab is selected. The 'BASIC' section includes fields for 'Name' (set to '128'), 'Description' (empty), 'Folder' (set to 'My Scans'), and 'Targets' (set to '10.88.40.128/27'). Below these fields are buttons for 'Upload Targets' and 'Add File'. At the bottom of the form are 'Save' and 'Cancel' buttons.

Abbildung 79: Scan des Subnetzes 10.88.40.128/27

Hier sieht man alle zugehörigen Netze, die zum Subnetz 10.88.40.128/27 gehören, die durch Host-Discovery ermittelt wurden.

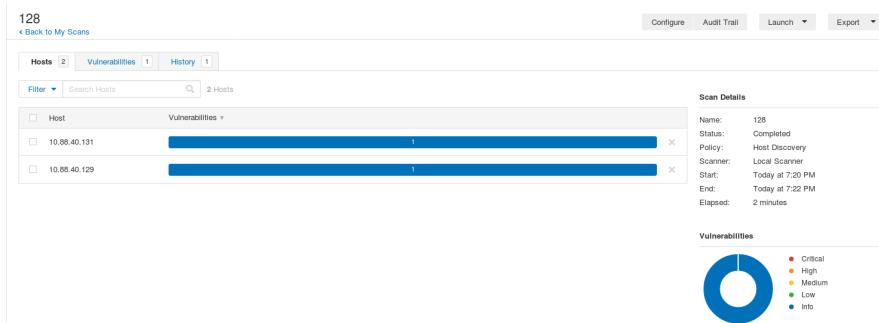


Abbildung 80: Alle Hosts im Subnetz 10.88.40.128/27

Scan des Subnetzes 10.88.40.160/27:

The screenshot shows the configuration for a new scan:

- Settings:**
 - BASIC:**
 - General: Name 160, Description (empty)
 - Schedule (disabled)
 - Notifications (disabled)
 - DISCOVERY:** Targets: 10.88.40.160/27
 - REPORT:** Folder: My Scans
- Buttons:** Upload Targets, Add File

Abbildung 81: Scan des Subnetzes 10.88.40.160/27

Hier sieht man alle zugehörigen Netze, die zum Subnetz 10.88.40.160/27 gehören, die durch Host-Discovery ermittelt wurden.

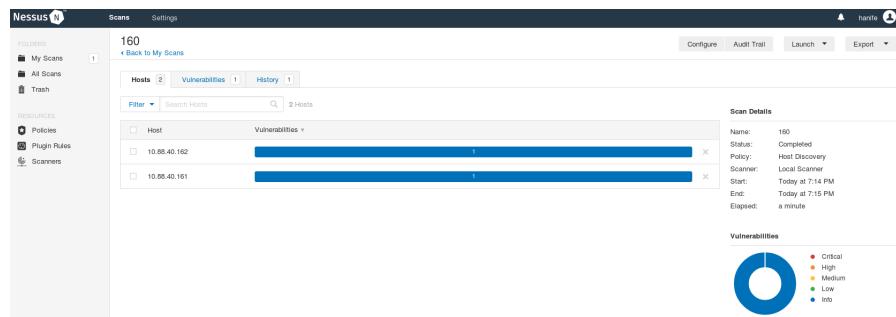


Abbildung 82: Alle Hosts im Subnetz 10.88.40.160/27

Alle Subnetze:

Im folgenden Bild wurden alle Subnetze gescannt, nämlich 10.88.40.64/27, 10.88.40.96/27, 10.88.40.128/27, 10.88.40.160/27 und 10.88.40.32/27.

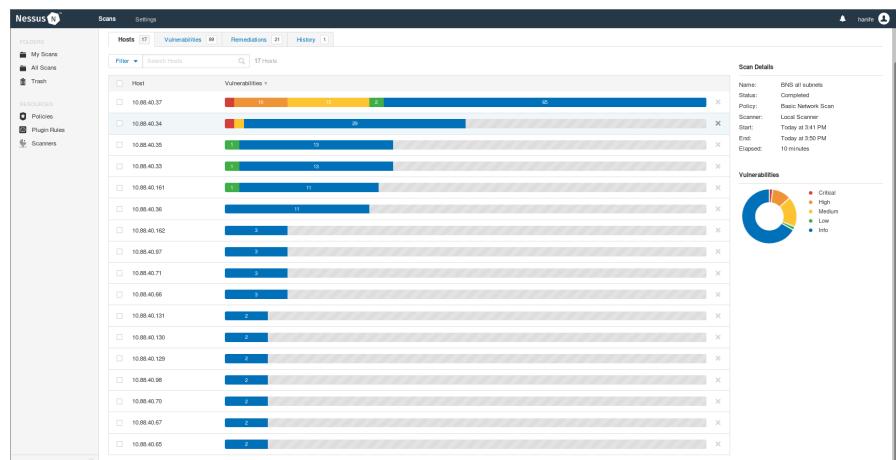


Abbildung 83: Alle Subnetze

Ergebnis aller Scans: Die Scans wurden als PDF-Datei extrahiert. Darin stehen alle relevanten Informationen, die über das erfolgreiche Scannen der Host Discovery ermittelt wurden. Die gesamte Datei befindet sich im Anhang der Email.

Solution of 7) Perform a network device identification on your subnet 10.88.X.P/27, see figure 1

set the ip for this machine to 172.16.15.P, where P stands for the ip address for your reserved ip, see below and change the dns to 134.100.9.61

group 1 will be 172.16.15.13
group 2 will be 172.16.15.23
group 3 will be 172.16.15.33
group 4 will be 172.16.15.43
group 5 will be 172.16.15.53
group 6 will be 172.16.15.63
group 7 will be 172.16.15.73
group 8 will be 172.16.15.83
group 9 will be 172.16.15.93
group 10 will be 172.16.15.103
group 11 will be 172.16.15.113
group 12 will be 172.16.15.123

Please download the installation guide and the user guide and read the Linux section carefully to solve the exercise.

Solution: Damit wir eine Internetverbindung von einem Hostrechner herstellen können wählen wir zuerst einen Host aus. Wir entschieden uns für pnid4-svr-hh mit dem Betriebssystem Centos 7. Als nächstes öffnen wir auf diesem Host die Netzwerkeinstellungen und fügen ein neue Schnittstelle (hier Profil 1) hinzu.

Da wir die Gruppe 4 sind, vergeben wir die IP Adresse 172.16.15.43 und tragen als Vorgaberooute die IP Adresse 172.16.15.249 ein. Der öffentliche DNS server der Uni-Hamburg läuft unter der IP Adresse 134.100.9.61, welche wir ebenfalls eintragen. Die Konfiguration über die GUI sieht dann folgendermaßen aus:

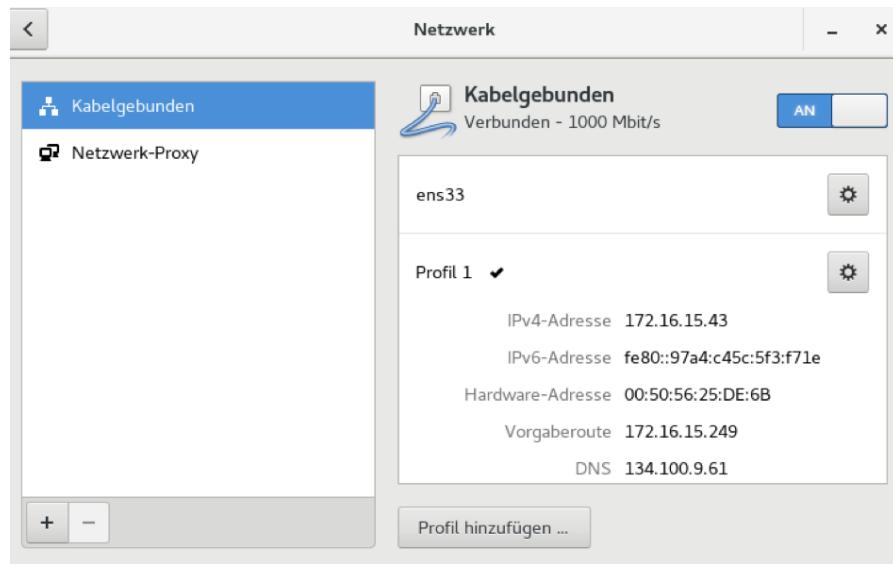


Abbildung 84: Netzwerkkonfiguration über die GUI

Die Internetverbindung ist nun erfolgreich hergestellt, sodass wir beliebige Webseiten, wie zum Beispiel www.google.de, auf unserer virtuellen Maschine (pnid4-svr-hh -> Server Hamburg) aufrufen können.

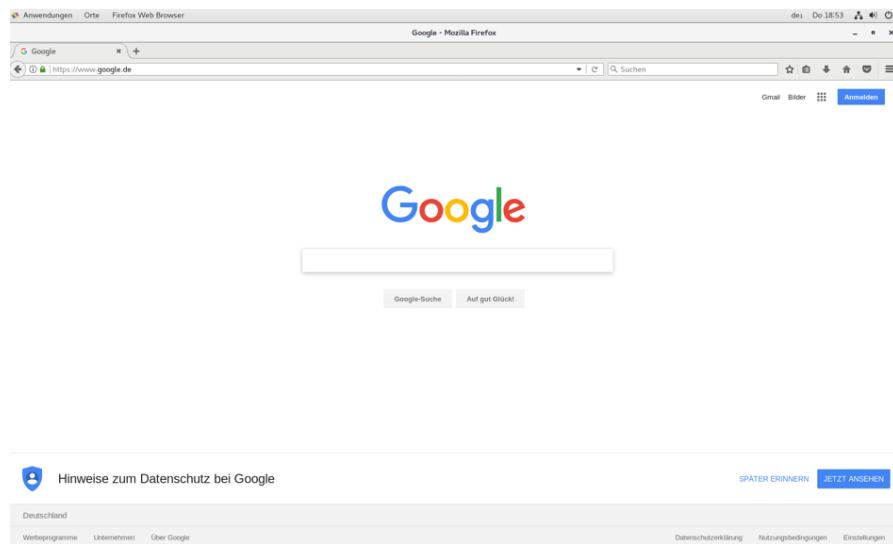


Abbildung 85: erfolgreiche Internetverbindung

Exercise 4: OpenVAS Network device identification

OpenVAS is a free scanner program. Look at the website https://www.bsi.bund.de/DE/Themen/ProdukteTools/OpenVAS/OpenVAS_node.html#doc2450430bodyText2 for more information or Google

OpenVAS includes following features:

- a) It is a vulnerability scanner
- b) It is a port scanner
- c) It is a host/device detection program
- d) It can be used to scan Netbios Servers e.g. Windows Servers and Samba Servers e)
OpenVAS can be used as a penetrating testing tool
- f) It is a client-server-system. The server performs the actual scan but it is controlled through the client. Both client and server can be run on the same system

Do the following steps to install and run OpenVAS:

1. Create a new Nessus virtual machine as shown in figure 1.
2. Login with your user name if not existent, please create it.
3. #su- <cr> 4. # vi /etc/selinux/config Set SELINUX=permissive 5. # set the ip for this machine to 172.16.15.P, where P stands for the ip address for your reserved ip, see below and change the dns to 134.100.9.61

group 1 will be 172.16.15.13
group 2 will be 172.16.15.23
group 3 will be 172.16.15.33
group 4 will be 172.16.15.43
group 5 will be 172.16.15.53
group 6 will be 172.16.15.63
group 7 will be 172.16.15.73
group 8 will be 172.16.15.83
group 9 will be 172.16.15.93

```
group 10 will be 172.16.15.102  
group 11 will be 172.16.15.113  
group 12 will be 172.16.15.123
```

```
# apt-get update && apt-get install -y openvas  
# openvas-setup  
# openvasmd --user=admin --new-password=admin  
# openvas-start
```

6. Use your webbrowser <https://10.88.X.P:9392/>

7. Now you must use openVAS to identify all the devices running on the following network as depicted on figure 1:

```
10.88.X.32/27  
10.88.X.64/27  
10.88.X.96/27  
10.88.X.128/27  
10.88.X.160/27
```

Einleitung OpenVas

OpenVAS (Open Vulnerability Assessment System[1]) ist eine Open Source Software (OSS), die Sicherheitslücken im Netzwerk sammelt und grafisch darstellt. Dabei kann der Benutzer dieses Dienstes die Prüftiefen der Sicherheitsuntersuchungen individuell konfigurieren. Sobald ein Scan durchgeführt ist, stellt OpenVAS eine Vielzahl von informationsreichen Abbildungen dar. Ebenso hat der Benutzer die Möglichkeit in den Prüfberichten zu sehen, um noch genauere und detailliertere Auskünfte einzusehen. Die, durch den Scan, gefundenen Probleme werden in Sicherheitsklassen wie Low und Medium eingestuft, sodass eine Priorität der zu behebenden Schwachstellen sich automatisch erschließen lässt. Es gibt eine Menge Funktionalitäten die ein OpenVAS ermöglicht, einige dieser Funktionalitäten werden in folgendem Absatz stichpunktartig ausgelegt:

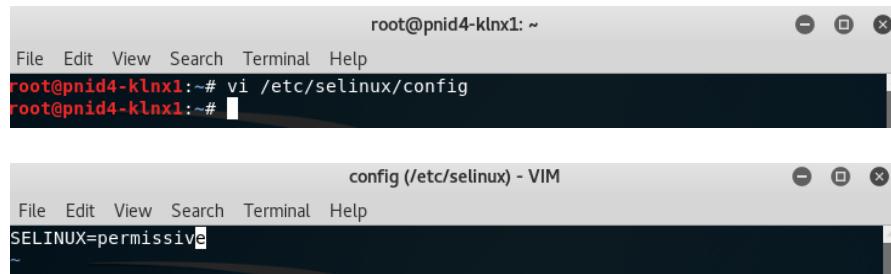
- Security-Scanning von ganzen Netzwerken

- Einbindung verschiedener Sicherheits-Tools dank entsprechender Schnittstellen und Steuerprotokolle
- Zentrale Sammlung und Auswertung der Prüfergebnisse
- Prüfung von Web Anwendungen möglich
- Management von Scan-Aufgaben
- Vergleich von Berichten
- Benutzeroberfläche -, GUI- und Web-basiert
- Alarmierung bei Richtlinien-Verstoß oder Gefahr

Installation: Um OpenVAS zu installieren, sollten wir nach Angaben im Skript folgende Aufgaben lösen:

1. Create a new Nessus virtual machine as shown in figure 1.
 2. Login with your user name if not existent, please create it.
 3. # su - <cr>
 4. # vi /etc/selinux/config
- Set SELINUX=permissive

So wie die Anleitung beschreibt haben wir die Schritte durchgeführt und SELINUX = permissive gesetzt.



```
root@pnid4-klnx1: ~
File Edit View Search Terminal Help
root@pnid4-klnx1:~# vi /etc/selinux/config
root@pnid4-klnx1:~#
```



```
config (/etc/selinux) - VIM
File Edit View Search Terminal Help
SELINUX=permissive
~
```

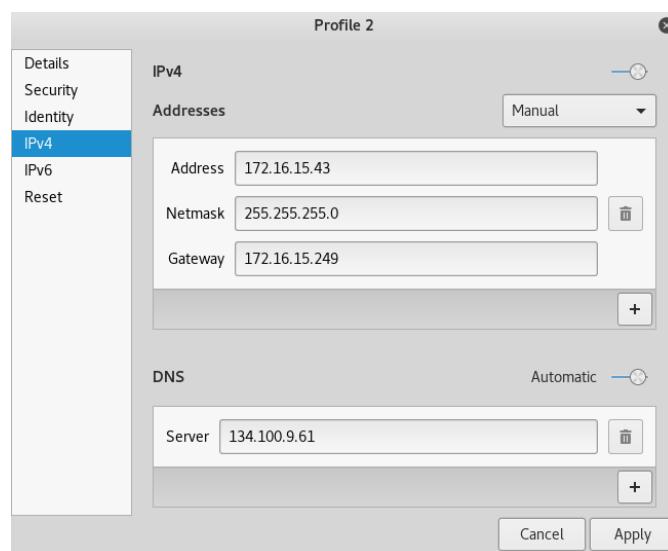
SELinux (Security-Enhanced Linux) ist in allgemeinen eine Erweiterung des Linux Kernels. Es verwaltet die Berechtigungen von unterschiedlichen Unix-Systemen. Die Berechtigung kann von dem Benutzer selbst gesetzt werden.

Es ist möglich SELinux im Enforcing oder Permissive Modus zu setzen. Enforcing, wie der Name selbst sagt, ist der Modus wo die SELinux Policy erzwungen wird. Im

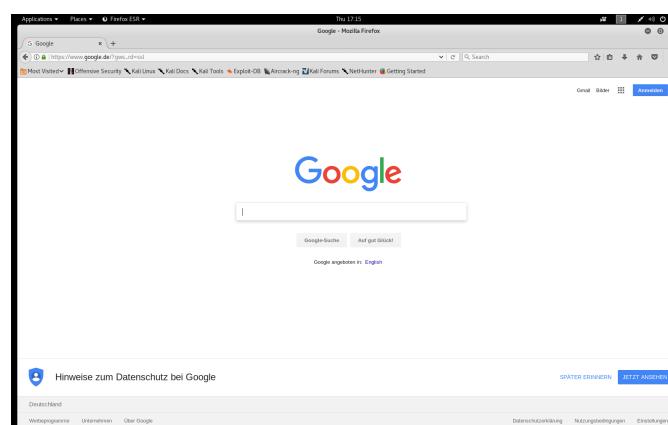
Gegenteil dazu werden im Permissive Modus alle Verletzungen protokolliert, aber nicht verhindert.

5. # set the ip for this machine to 172.16.15.P, where P stands for the ip address for your reserved ip, see below and change the dns to 134.100.9.61, group 4 will be 172.16.15.43

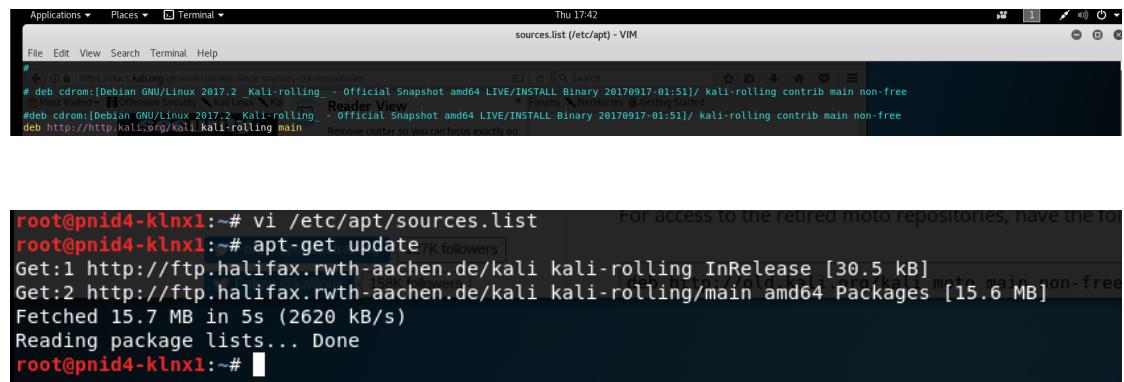
Wie die Aufgabe appelliert, haben wir die IP-Adresse 172.16.15.43 über den DNS 134.100.9.61 für pnid4-klnx1 konfiguriert.



Infolgedessen konnten wir über diese VM eine Verbindung zum Internet aufbauen, als Beispiel hier zu Google.



Nachdem wir IPv4 für diese VM eingerichtet haben, lassen wir die Paketlisten (sources.list) durch den Kommando apt-get update neu einlesen. Es wird auf die Signature dieser Paketlisten gepürft.



```

root@pnid4-klx1:~# vi /etc/apt/sources.list
root@pnid4-klx1:~# apt-get update
Get:1 http://ftp.halifax.rwth-aachen.de/kali kali-rolling InRelease [30.5 kB]
Get:2 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 Packages [15.6 MB]
Fetched 15.7 MB in 5s (2620 kB/s)
Reading package lists... Done
root@pnid4-klx1:~#

```

Abbildung 86: apt-get update

Nachdem wir das update durchgeführt haben, installieren wir Pakete wenn es welche gibt auf eine aktuelle Version durch den Kommando apt-get upgrade.



```

root@pnid4-klx1:~# apt-get upgrade
Reading package lists... Done
Building dependency tree...
Reading state information... Done
Retired Kali sana (2.0) Repositories
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  dconf-editor dconf-tools gir1.2-nm-1.0 libblas-common libglib2.0-common libqcustomplot0.3 python3.5 python3.5-minimal tcpd
Use 'apt autoremove' to remove them.
The following packages have been kept back:
  bdfproxy bind9-host commix debconf-ilon dnutils e2fslibs e2fsprogs mdai-bin nzb nirc1.2-ndkpxbuf-2.0 nirc1.2-javascriptcorekit-4.0 nirc1.2-totem-1.0 nirc1.2-webrtc2-4.0 nis

```

Abbildung 87: apt-get upgrade

Und um openVas vollständig zu installieren, haben wir den Kommendo apt -get install -y openvas genutzt. Das Installieren hat sehr viel Zeit in Anspruch genommen, denn über diesen Kommando werden nicht nur die Pakete installiert, sondern darüber hinaus noch nicht installierte Abhängigkeiten werden ebenfalls installiert. Man sieht dann auch im Terminal welche Packete installisiert wurden.

```

root@pnid4-klnx1:~# apt-get install -y openvas
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required; they are probably no longer needed since the release of Kali 2016.1. Kali Rolling users are encouraged to remove them.
  dconf-editor dconf-tools gir1.2-mm-1.0 libblas-common libgomp-1.0-common libqcustomplot1.3 python3.5 python3.5-minimal tcpd
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  doc-base fonts-texgyre gnutls-bin greenbone-security-assistant greenbone-security-assistant-common libgnutls-dane0 libnihredis0.13 libmicrohttpd12 libopenvas9 libunbound2
  libunbound2 perl openvas-client openvas-manager openvas-scanner preview-latex-style redis-server redis-tools tex-gyre texlive-fonts-recommended
  texlive-fonts-recommended-doc texlive-latex-extra texlive-latex-recommended texlive-latex-recommended-doc texlive-pictures texlive-pictures-doc
  texlive-plain-generic tipa
Suggested packages:
  rarian-compat openvas-client pncan strobe ruby-redis libcurl-profiles libfile-which-perl libspreadsheet-parseexcel-perl texlive-pstricks dot2tex preTeXt ruby-tcltk | libtcltk-ruby
The following NEW packages will be installed:
  doc-base fonts-texgyre gnutls-bin greenbone-security-assistant greenbone-security-assistant-common libgnutls-dane0 libnihredis0.13 libmicrohttpd12 libopenvas9 libunbound2
  libunbound2 perl openvas-client openvas-manager openvas-scanner preview-latex-style redis-server redis-tools tex-gyre texlive-fonts-recommended
  texlive-fonts-recommended-doc texlive-latex-extra texlive-latex-recommended texlive-latex-recommended-doc texlive-pictures texlive-pictures-doc
  texlive-plain-generic tipa
0 upgraded, 31 newly installed, 0 to remove and 127 not upgraded.
Need to get 625 MB of archives.
After this operation, 945 MB of additional disk space will be used.
Get:1 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 libubuid-perl amd64 0.27-1+b2 [18.4 kB]
Get:2 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 libunbound-tiny-perl all 1.70-1 [32.0 kB]
Get:3 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 doc-base all 0.10.7 [100 kB]
Get:4 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 fonts-texgyre-all 20160520-1 [8761 kB]

```

Abbildung 88: OpenVas Installation

Anschließend haben wir OpenVas eingerichtet, indem wir folgenden Kommando in den Terminal eingegeben haben: openvas-setup. Wie der Name bereits verrät ist dieser Kommando für die Einrichtung für OpenVAS da.



```

Applications ▾ Places ▾ Terminal ▾ Thu 19:50
root@pnid4-klnx1:~#
File Edit View Search Terminal Help
root@pnid4-klnx1:~# systemctl list-unit-files |grep rsync
rsync.service                                         enabled
root@pnid4-klnx1:~# systemctl disable rsync.service
Synchronizing state of rsync.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable rsync
root@pnid4-klnx1:~# systemctl list-unit-files |grep rsync
rsync.service                                         disabled
root@pnid4-klnx1:~# systemctl enable rsync.service
Synchronizing state of rsync.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable rsync
root@pnid4-klnx1:~# systemctl list-unit-files |grep rsync
rsync.service                                         enabled
root@pnid4-klnx1:~# systemctl start rsync.service
root@pnid4-klnx1:~# systemctl [grep rsync
root@pnid4-klnx1:~# openvas-setup
OK: Directory for keys (/var/lib/openvas/private/CA) exists.
OK: Directory for certificates (/var/lib/openvas/CA) exists.
OK: CA key found in /var/lib/openvas/private/CA/cakey.pem
OK: CA certificate found in /var/lib/openvas/CA/cacert.pem
OK: CA certificate verified.
OK: Certificate /var/lib/openvas/CA/servercert.pem verified.
OK: Certificate /var/lib/openvas/CA/clientcert.pem verified.

OK: Your OpenVAS certificate infrastructure passed validation.
OpenVAS community feed server - http://www.openvas.org/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the OpenVAS mailing lists
or the OpenVAS IRC chat. See http://www.openvas.org/ for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be blocked.

receiving incremental file list
plugin_feed_info.inc
      1,131 100%   1.08MB/s   0:00:00 (xfr#1, to-chk=0/1)

sent 43 bytes  received 1,235 bytes  852.00 bytes/sec
total size is 1,131 speedup is 0.88
OpenVAS community Feed server - http://www.openvas.org/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the OpenVAS mailing lists
or the OpenVAS IRC chat. See http://www.openvas.org/ for details.

By using this service you agree to our terms and conditions.

```

Abbildung 89: OpenVas Konfiguration

Im nächsten Schritt haben wir uns einen Benutzer und das dazugehörige Passwort für den Manager daemon über den Kommandaro `openvasmd --user=admin --new-password=admin` angelegt. Im Anschluss dessen haben wir die Services anhand Eingabe des `openvas-start` Kommando gestartet.

```
root@pnid4-klxn1:~# openvasmd --user=admin --new-password=admin
root@pnid4-klxn1:~# openvas-start
Starting OpenVas Services
root@pnid4-klxn1:~#
```

Abbildung 90: user anlegen

6. Use your webbrowser <https://127.0.0.1:9392/>

Im nächsten Schritt haben wir `https://127.0.0.1:9392/` in die Adresszeile unseres Browsers eingegeben und haben unsere Login-Daten eingetippt.

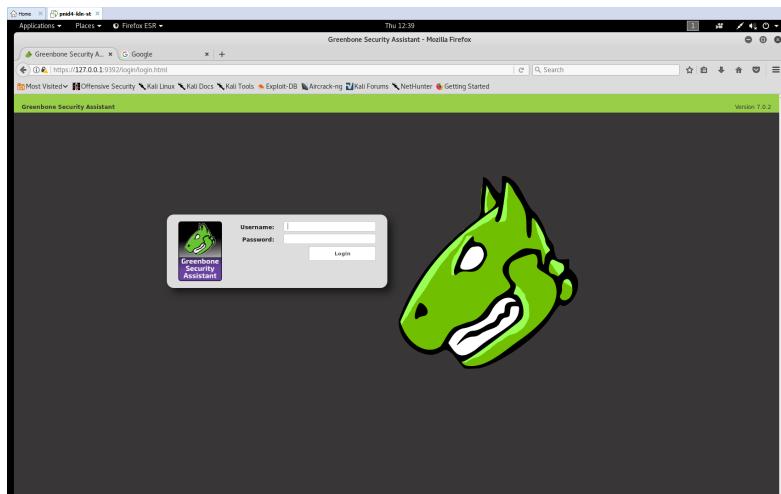


Abbildung 91: Login OpenVas

7. Now you must use openVAS to identify all the devices running on the following network as depicted on figure 1:

Daraufhin haben wir OpenVAS genutzt um alle VM in unserem Netz zu scannen. In Folgenden Bildausschnitten ist zu erkennen, welche VM (mit Angabe der IP-Adresse)

gescannt wurde.

Scan des Subnetzes 10.88.40.32/27:

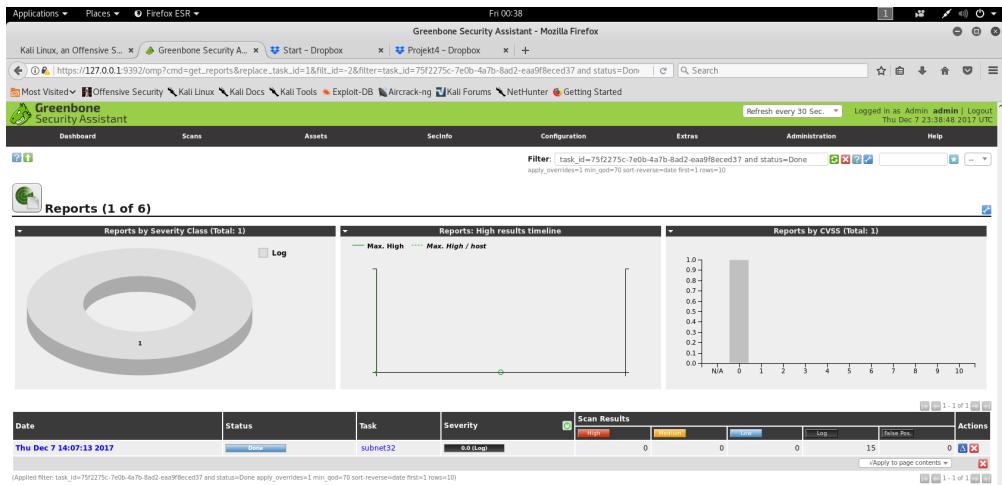


Abbildung 92: Scan des Subnetzes 10.88.40.32/27

Scan des Subnetzes 10.88.40.64/27:

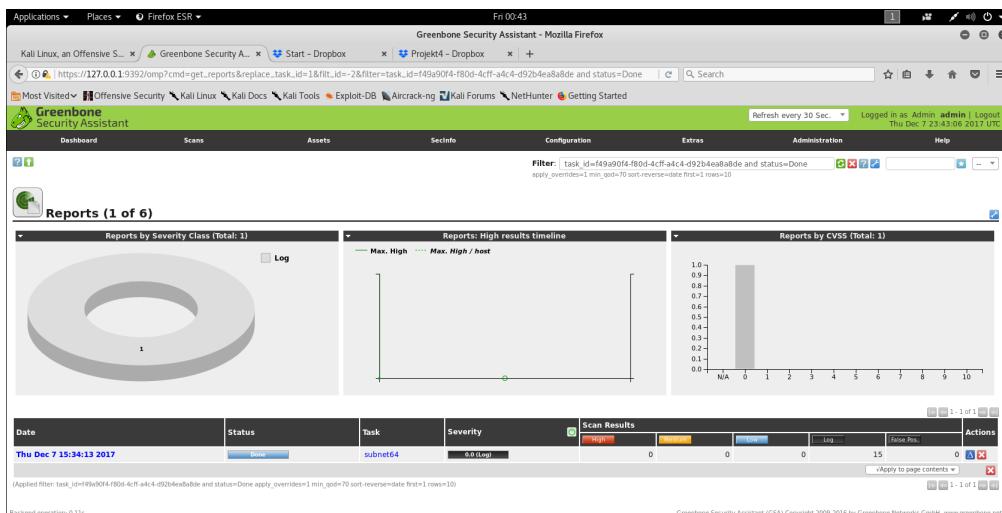


Abbildung 93: Scan des Subnetzes 10.88.40.64/27

Scan des Subnetzes 10.88.40.96/27:

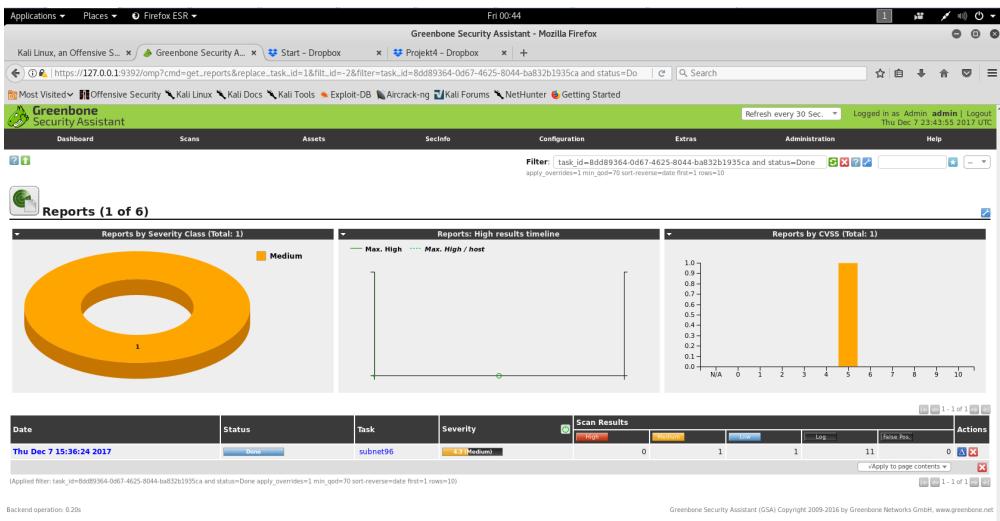
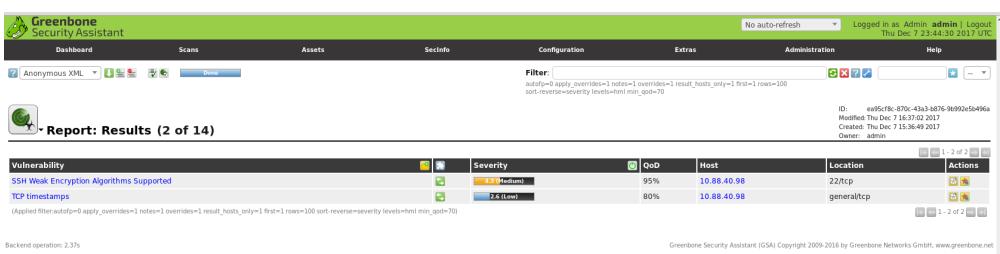


Abbildung 94: Scan des Subnetzes 10.88.40.96/27



Scan des Subnetzes 10.88.40.128/27:

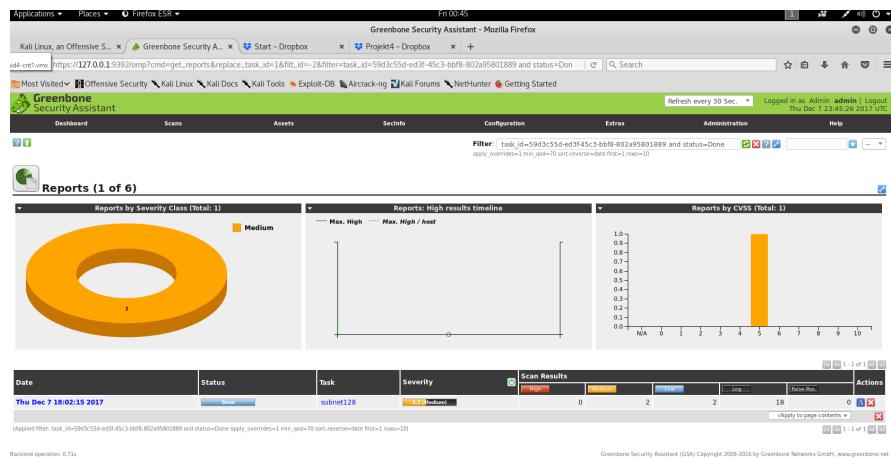


Abbildung 95: Scan des Subnetzes 10.88.40.128/27

Scan des Subnetzes 10.88.40.160/27:

Abbildung 96: Scan des Subnetzes 10.88.40.160/27

Ebenfalls ist hier eine Übersicht aller (10.88.40.32/27 ,10.88.40.64/27 ,10.88.40.96/27 10.88.40.128/27 und 10.88.40.160/27) gescannten Netze. Wir haben 5 Berichte durch den Scan erhalten, welche wir als PDF im Anhang beifügen werden.

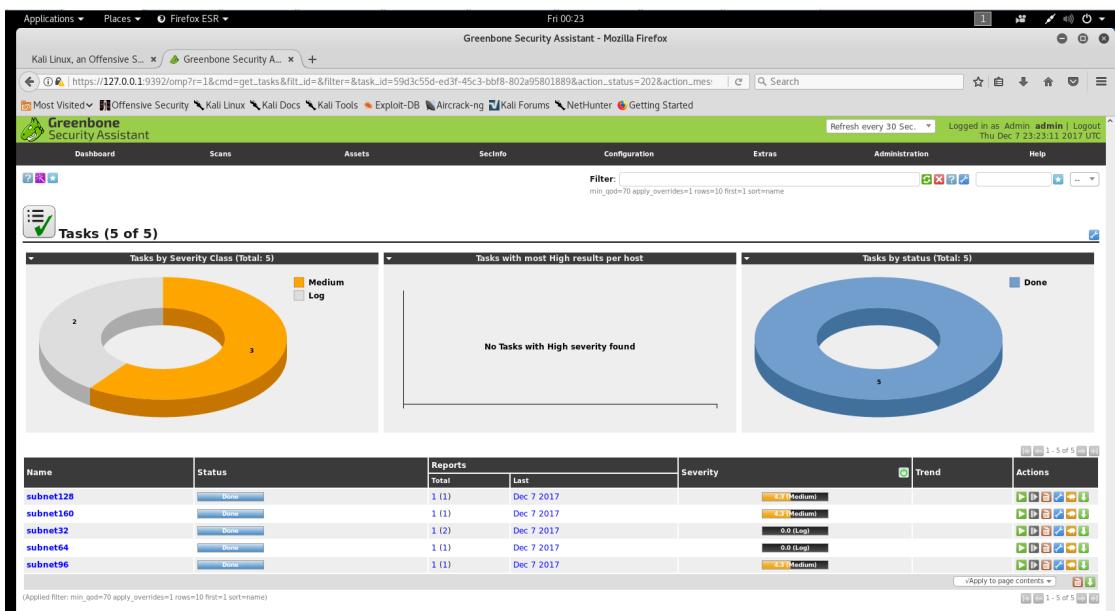


Abbildung 97: Scan aller Subnetze

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 oder empfangen werden, abfangen und anzeigen lassen.

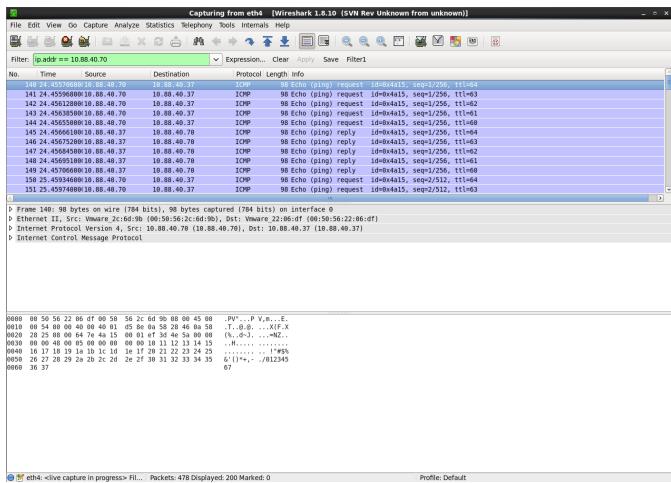


Abbildung 98: Wireshark Filter ip.addr == 10.88.40.70

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

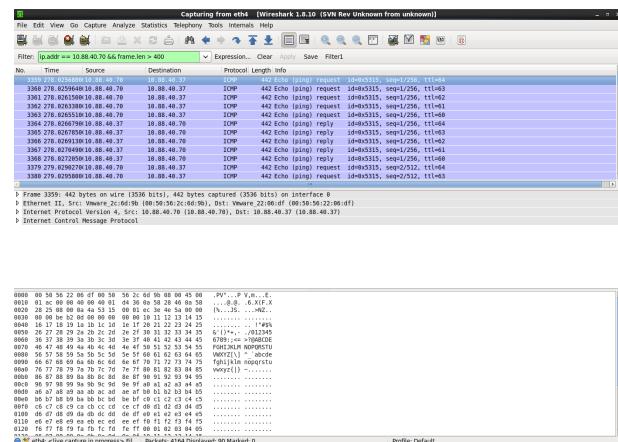


Abbildung 99: Wireshark Filter ip.addr == 10.88.40.70 && frame.len > 400

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 100: 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: $ip.dst == 10.88.40.70 \text{ and } tcp.port == 23$. Bei TCP handelt es sich um ein Übertragungsprotokoll (Transmission Control Protocol) aus der Familie der Internetprotokolle. Port 23 ist standardisiert für den Service Telnet.

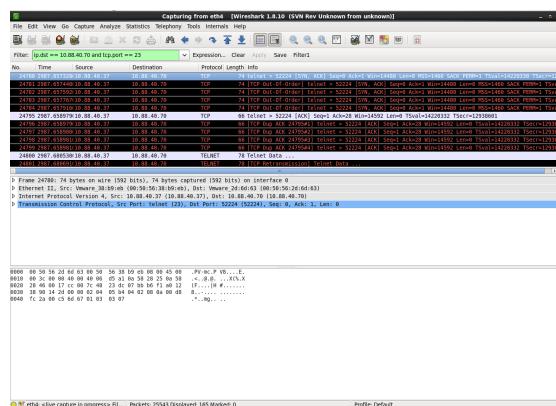


Abbildung 101: Wireshark Filter $ip.dst == 10.88.40.70 \text{ and } tcp.port == 23$

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.

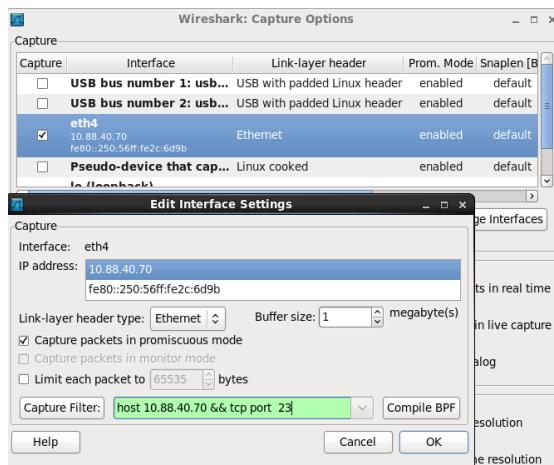


Abbildung 102: Wireshark Filter host 10.88.40.70 && tcp port 23

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.

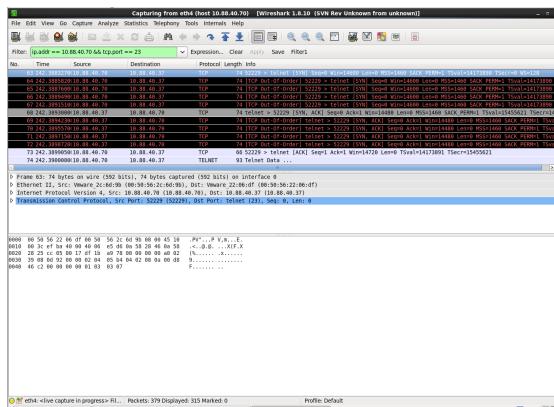


Abbildung 103: Wireshark Filter ip.addr == 10.88.40.70 && tcp.port == 23

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 Wirehsark 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 servie 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 successfull 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.

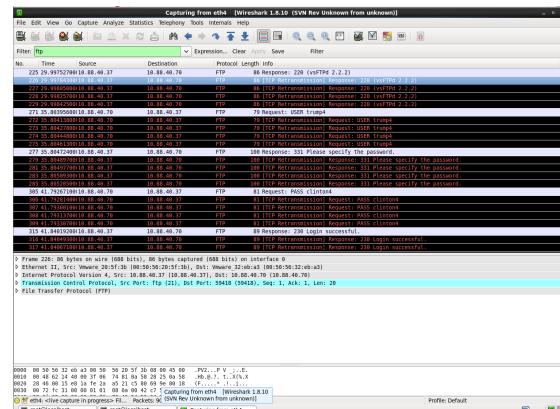


Abbildung 104: Wireshark Filter ftp

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

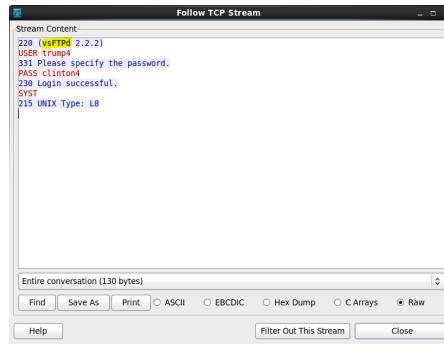


Abbildung 105: Wireshark ftp Login Passwort

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.

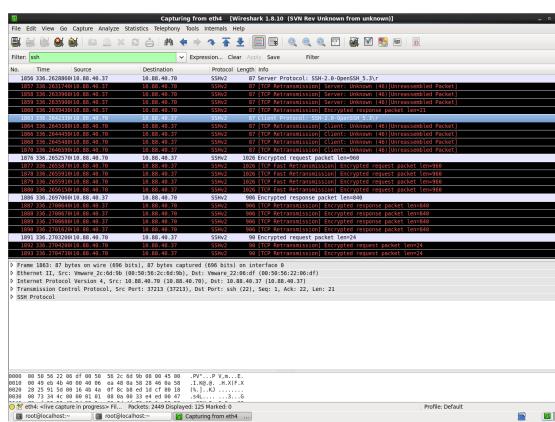


Abbildung 106: Wireshark Filter ssh

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

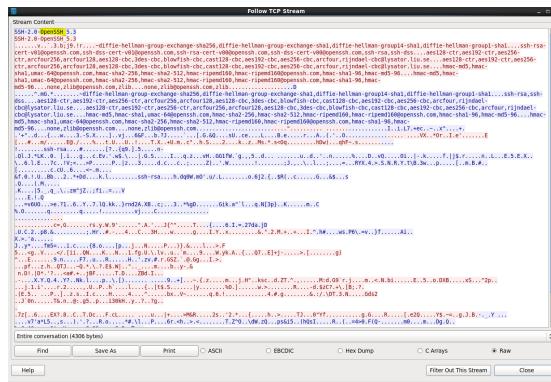


Abbildung 107: Wireshark verschlüsselte Datenfragmente

Exercise 5: Experimenting with network monitoring tools

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

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.

```
[root@localhost ~]# nmap -sF 10.0.8.40:37
Starting Nmap 5.51 ( http://nmap.org ) at 2018-01-04 14:33 CET
Nmap scan report for 10.0.8.40
Host is up (0.0000s latency).
No ports were open or filtered.
No ports were closed or filtered.
No ports were filtered or unfiltered.
PORT      STATE    SERVICE
22/tcp    open     filtered ssh
23/tcp    open     filtered telnet
53/tcp    open     filtered dns
111/tcp   open     filtered rpcbind

TIP: Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
```

Port detection performed. Please report any incorrect results at <http://nmap.org/submit/>.

Time taken: 0.000 seconds (0 hosts up) | 1 host up | scanned in 3.02 seconds

```
[root@localhost ~]#
```

Abbildung 108: 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 109: 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 PRESHARED KEY use ipsec ranbits 256 > filename

Installation von Openswan auf beiden Rechnern Hamburg (pnid4-mid-hh) und Berlin (pnid4-cnt-bln)

```

[root@localhost ~]# mount -t iso9660 /dev/sr0 /dvdrom
mount: block device /dev/sr0 is write-protected, mounting read-only
[root@localhost ~]# yum -y install openswan
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 openswan.x86_64 0:2.6.32-27.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package      Arch    Version       Repository   Size
=====
Installing:
openswan     x86_64  2.6.32-27.el6  LocalRepo   895 k

Transaction Summary
=====
Install      1 Package(s)

Total download size: 895 k
Installed size: 2.6 M
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : openswan-2.6.32-27.el6.x86_64          1/1
  Verifying  : openswan-2.6.32-27.el6.x86_64          1/1

Installed:
  openswan.x86_64 0:2.6.32-27.el6

Complete!
[root@localhost ~]# █

```

Abbildung 110: Installation Openswan

Netzwerk für die Konfiguration der VPN-Verbindung

VPN

HOST TO HOST VPN

Figure 1

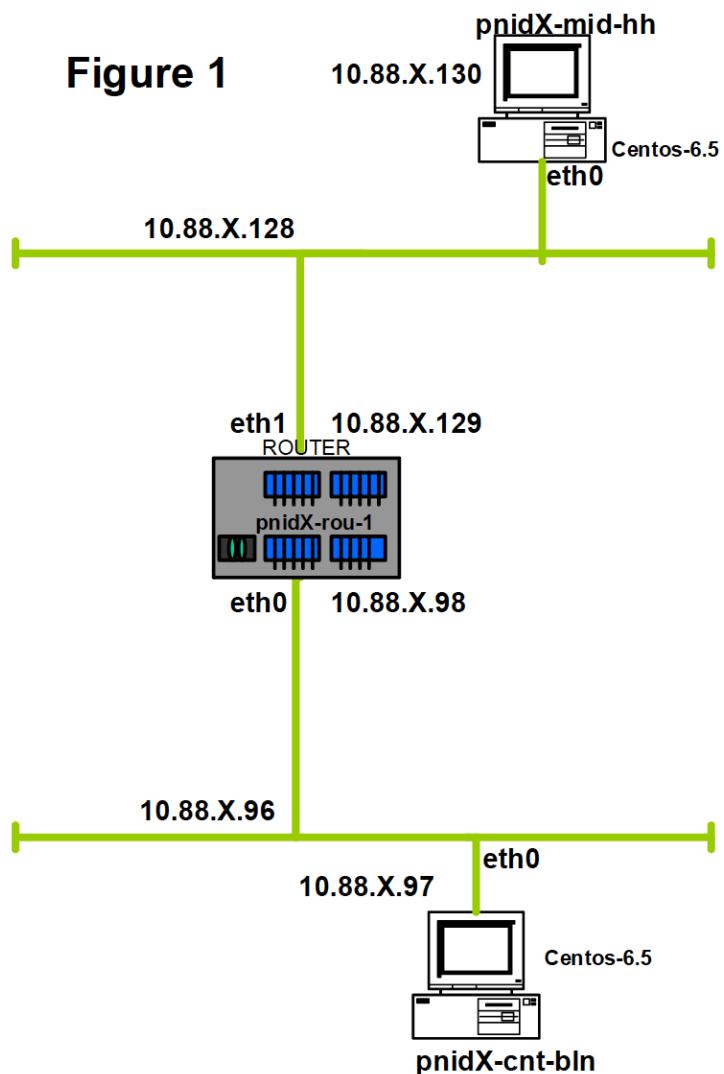


Abbildung 111: Figure 1: HOST-TO-HOST-VPN

- i) Capture http packets between pnidX-mid-hh and pnidX-cnt-blن using Wireshark before establishing a tunnel and save the packet captured. Please include this with

your lab report.

Vor der Einrichtung von Openswan können HTTP-Pakete von beliebigen Angreifern ohne Probleme abgehört werden. VPN soll dafür sorgen, dass dies verhindert wird. Zunächst starten wir auf pnid4-cnt-bln Wireshark, um HTTP-Pakete abzufangen. Um HTTP-Pakete von pnid4-mid-hh abzufangen, geben wir den Filter "dst 10.88.40.130 && tcp && port 80" ein (siehe Abbildung ..).

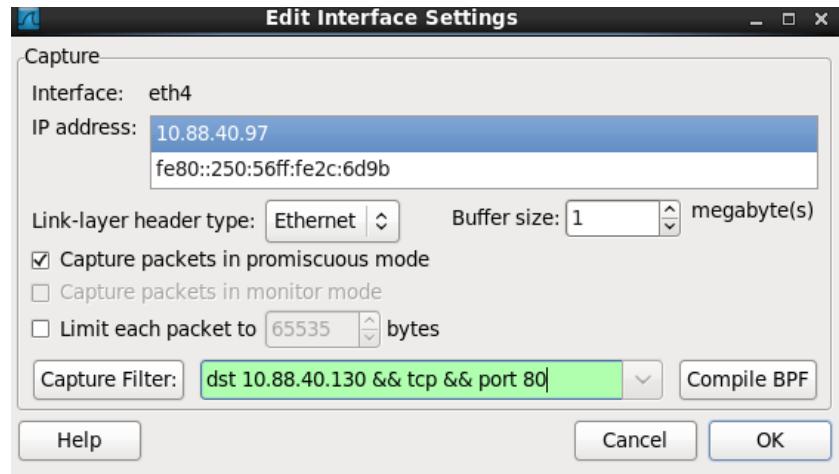


Abbildung 112: dst 10.88.40.130 && tcp && port 80

Auf dem Rechner pnid4-cnt-bln wird im Browser die IP-Adresse von pnid4-mid-hh eingeben, um HTTP-Pakete abzufangen (siehe Abbildung ..)



Abbildung 113: http://10.88.40.130

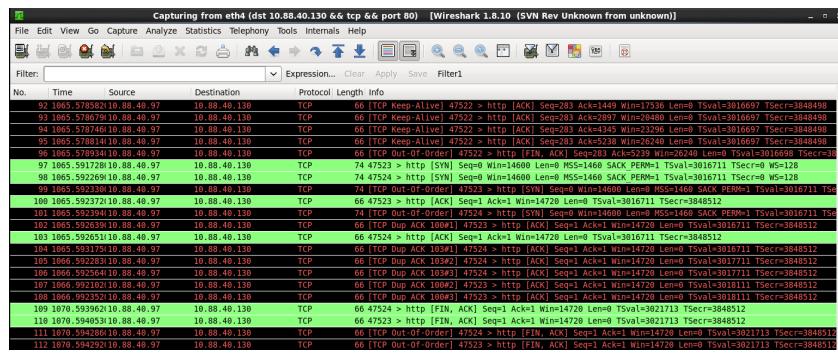


Abbildung 114: Filter: dst 10.88.40.130 && tcp && port 80

ii) Establish a tunneled IPSec connection, using a 256 preshared secret. Confirm that the connection is established.

Capture again http packets between pnidX-mid-hh and pnidX-cnt-blh using Wireshark after establishing a tunnel and save the packet captured. Please include this with your lab report.

Mit Hilfe des Befehls # ipsec ranbits 256 > psk.secrets wird ein zufällig erzeugter Key generiert, der in der Datei psk.secrets gespeichert wird.

```
[root@pnid4-mid-hh ipsec.d]# ipsec ranbits 256 > psk.secrets
```

Abbildung 115: ipsec ranbits 256 >

Die Datei wird so angepasst, dass Sie folgende Struktur bekommt:

% IP-Adresse-mid-hh % IP-Adresse-cnt-bln : PSK "der_Key_welches_über_ipsec_ranbits_generiert wurde "(siehe Abb. ...)

```
[root@pnid4-mid-hh ipsec.d]# cat psk.secrets  
10.88.40.97 10.88.40.130 : PSK 0x08ba6fd0_212c2a15_fa671f53_16fc4b87_dc231639_479d3714_b3f46bb5_58dc7fc2
```

Abbildung 116: psk.secrets

Als nächstes wird die psk.conf Datei erstellt, welches sich ebenfalls im Verzeichnis ipsec.d befindet. Hier erfolgt die Konfiguration des Tunnels. Dabei wird die Authentifizierung über authby angegeben. Mit type wird der Typ der Verbindung (tunnel, transport) bezeichnet. Die betroffenen Kommunikationspartner werden mit left und right dargestellt. Left bezeichnet dabei den Host, auf den wir uns befinden. Right hingegen bezeichnet den dazugehörigen Kommunikationspartner. Left und right entsprechen dabei die IP-Adressen der jeweiligen Kommunikationspartner. Auto bezeichnet die Operation, welche beim starten von IPsec ausgeführt werden soll.

```
[root@pnid4-mid-hh ipsec.d]# vi psk.conf
[root@pnid4-mid-hh ipsec.d]# cat psk.conf
conn psk
    type=tunnel
    auto=add
    authby=secret

    left=10.88.40.97
    #leftsubnet=10.88.40.96/27

    right=10.88.40.130
    #rightsubnet=10.88.40.128/27
[root@pnid4-mid-hh ipsec.d]#
```

Abbildung 117: psk.conf

Damit alle Dateien innerhalb des Verzeichnisses /etc/ipsec.d mit der Endung .conf eingebunden werden können, muss die letzte Zeile (siehe Abb.) auskommentiert werden.

```

[root@pnid4-mid-hh etc]# cat ipsec.conf
# /etc/ipsec.conf - Openswan IPsec configuration file
#
# Manual:      ipsec.conf.5
#
# Please place your own config files in /etc/ipsec.d/ ending in .conf
version 2.0      # conforms to second version of ipsec.conf specification

# basic configuration
config setup
    # Debug-logging controls: "none" for (almost) none, "all" for lots.
    # klipsdebug=none
    # plutodebug="control parsing"
    # For Red Hat Enterprise Linux and Fedora, leave protostack=netkey
    protostack=netkey
    nat_traversal=yes
    virtual_private=
    oe=off
    # Enable this if you see "failed to find any available worker"
    # nhelpers=0

#You may put your configuration (.conf) file in the "/etc/ipsec.d/" and uncomment this.
include /etc/ipsec.d/*.conf

```

Damit beide Kommunikationspartner dieselben Dateien haben, wird die Datei psk.secrets und die Datei psk.conf mit dem Befehl scp zum jeweils anderen Kommunikationspartner geschickt. Scp(Secure Copy) ist ein Protokoll und sorgt für eine verschlüsselte Übertragung von Daten zwischen zwei Computern.

```

[root@pnid4-mid-hh ipsec.d]# scp /etc/ipsec.d/psk.secrets root@10.88.40.97:/etc/ipsec.d/psk.secrets
root@10.88.40.97's password:                                                 100% 105     0.1KB/s  00:00
psk.secrets
[root@pnid4-mid-hh ipsec.d]# scp /etc/ipsec.d/psk.conf root@10.88.40.97:/etc/ipsec.d/psk.conf
root@10.88.40.97's password:                                                 100% 145     0.1KB/s  00:00
psk.conf
[root@pnid4-mid-hh ipsec.d]# 

```

Auf beiden Rechnern wird ipsec gestartet:

```

[root@pnid4-mid-hh /]# ipsec setup reload
ipsec_setup: Stopping Openswan IPsec...
ipsec_setup: Starting Openswan IPsec U2.6.32/K2.6.32-431.el6.x86_64...
ipsec_setup: /usr/libexec/ipsec/addconn Non-fips mode set in /proc/sys/crypto/ips_enabled

```

Abbildung 118: ipsec setup reload

Mit dem Befehl `ipsec auto --add` wird eine neue Verbindung aus der Konfigurationsdatei in die Pluto-Datenbank eingelesen.

```
[root@pnid4-mid-hh /]# ipsec auto --add psk
/usr/libexec/ipsec/addconn Non-fips mode set in /proc/sys/crypto/fips_enabled
[root@pnid4-mid-hh /]# ipsec setup status
IPsec running - pluto pid: 3092
pluto pid 3092
No tunnels up
```

Abbildung 119: ipsec auto –add

Mit dem Befehl `ipsec auto --up psk` wird die Verbindung gestartet. Dabei versucht Pluto in der internen Datenbank der geladenen Konfiguration die benötigte Verbindung aufzubauen.

```
[root@pnid4-mid-hh /]# ipsec auto --up psk
104 "psk" #1: STATE_MAIN_I1: initiate
003 "psk" #1: received Vendor ID payload [Openswan (this version) 2.6.32 ]
003 "psk" #1: received Vendor ID payload [Dead Peer Detection]
003 "psk" #1: received Vendor ID payload [RFC 3947] method set to=109
106 "psk" #1: STATE_MAIN_I2: sent MI2, expecting MR2
003 "psk" #1: NAT-Traversal: Result using RFC 3947 (NAT-Traversal): no NAT detected
108 "psk" #1: STATE_MAIN_I3: sent MI3, expecting MR3
003 "psk" #1: received Vendor ID payload [CAN-IKEv2]
004 "psk" #1: STATE_MAIN_I4: ISAKMP SA established {auth=OAKLEY_PRESHARED_KEY cipher=aes_128 prf=oakley_sha group=modp2048}
117 "psk" #2: STATE_QUICK_I1: initiate
004 "psk" #2: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel mode {ESP=>0x8d14b6d0 <0x724dfaaf xfrm=AES_128-HMAC_SHA1 NATOA=none NATD=none DPD=none}
```

Abbildung 120: ipsec auto –up

Pluto ist der IKE-Daemon, der das IKE-Protokoll implementiert. Pluto erstellt automatische Sicherheitsbeziehungen, die untereinander geteilt werden. Für die Authentifizierung verwendet Pluto Shared Secrets oder RSA Signaturen.

```
[root@pnid4-mid-hh /]# ipsec status
IPsec running - pluto pid: 3092
pluto pid 3092
1 tunnels up
some eroutes exist
```

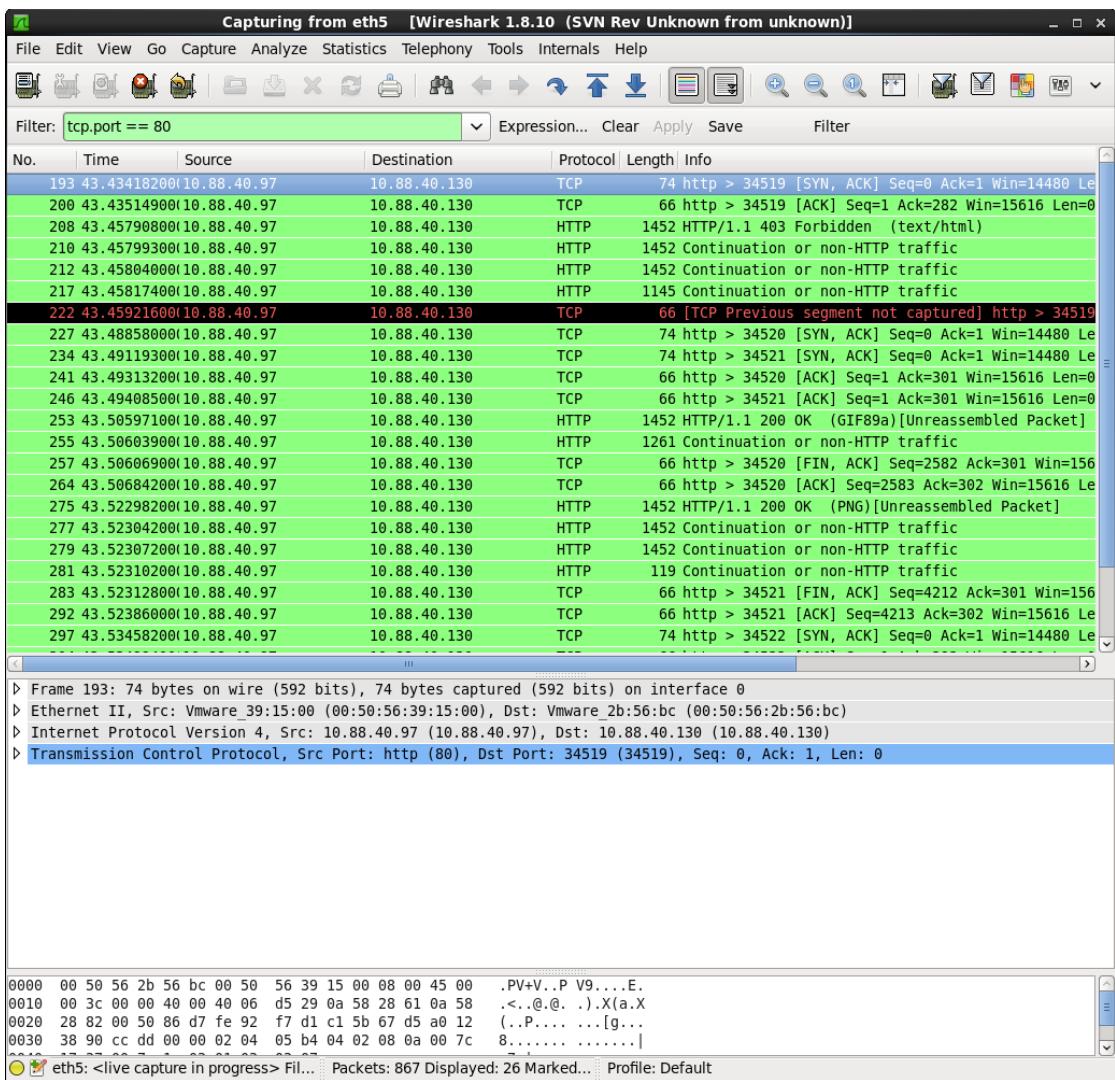
- iii) Secure the computer traffic by creating iptables rules to permit only IPSec traffic between the two computers. All other, non-IPSec packets must be denied.

```
[root@pnid4-mid-hh etc]# iptables -F
[root@pnid4-mid-hh etc]# iptables -A FORWARD -p esp -s 10.88.40.130 -d 10.88.40.97 -j ACCEPT
[root@pnid4-mid-hh etc]# iptables -A FORWARD -p esp -s 10.88.40.97 -d 10.88.40.130 -j ACCEPT
[root@pnid4-mid-hh etc]# iptables -A FORWARD -p udp --dport 500 -s 10.88.40.130 -d 10.88.40.97 -j ACCEPT
[root@pnid4-mid-hh etc]# iptables -A FORWARD -p udp --dport 500 -s 10.88.40.97 -d 10.88.40.130 -j ACCEPT
[root@pnid4-mid-hh etc]# iptables -A FORWARD -p udp --dport 4500 -s 10.88.40.130 -d 10.88.40.97 -j ACCEPT
[root@pnid4-mid-hh etc]# iptables -A FORWARD -p udp --dport 4500 -s 10.88.40.97 -d 10.88.40.130 -j ACCEPT
[root@pnid4-mid-hh etc]# iptables -A FORWARD -j REJECT
```

```
[root@pnid4-mid-hh ipsec.d]# cat ipsec_iptables_config
# Generated by iptables-save v1.4.7 on Thu Jan 18 17:32:12 2018
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p esp -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p esp -j ACCEPT
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p udp -m udp --dport 500 -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p udp -m udp --dport 500 -j ACCEPT
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p udp -m udp --dport 4500 -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p udp -m udp --dport 4500 -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-port-unreachable
COMMIT
# Completed on Thu Jan 18 17:32:12 2018
```

```
[root@pnid4-mid-hh ipsec.d]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
ACCEPT    esp  --  10.88.40.130    10.88.40.97
ACCEPT    esp  --  10.88.40.97    10.88.40.130
ACCEPT    udp  --  10.88.40.130    10.88.40.97      udp dpt:isakmp
ACCEPT    udp  --  10.88.40.97    10.88.40.130      udp dpt:isakmp
ACCEPT    udp  --  10.88.40.130    10.88.40.97      udp dpt:ipsec-nat-t
ACCEPT    udp  --  10.88.40.97    10.88.40.130      udp dpt:ipsec-nat-t
REJECT   all  --  anywhere        anywhere          reject-with icmp-port-unreachable

Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
[root@pnid4-mid-hh ipsec.d]#
```



iv) After a successful IPSec connection has been established, create the following files using the given commands:

File ipsec_ifconfig.dump

ifconfig > ipsec_fconfig.dump

File ipsec_look.dump

ipsec look > ipsec_look.dump

File ipsec_route.dump

Route -n > ipsec_route.dump

The following files must be included with the lab report:

- 1) /etc/ipse.secrets
- 2) /etc/ipse.conf
- 3) /etc/ipsec_iptables - a file which contains your iptables rules
- 4) ipsec_ifconfig.dump
- 5) ipsec_look.dump
- 6) ipsec_route.dump

Files from Berlin:

```
[root@localhost ipsec.d]# ifconfig > ipsec_fconfig.dump
[root@localhost ipsec.d]# ipsec look > ipsec_look.dump
[root@localhost ipsec.d]# route -n > ipsec_route.dump
[root@localhost ipsec.d]# █
```

```
[root@localhost ipsec.d]# cat /etc/ipsec.d/psk.secrets
10.88.40.97 10.88.40.130 : PSK 0x08ba6fd0_212c2a15_fa671f53_16fc4b87_dc231639_47
9d3714_b3f46bb5_58dc7fc2 █
```

```
[root@localhost ipsec.d]# cat /etc/ipsec.d/psk.conf
conn psk
    type=tunnel
    auto=add
    authby=secret

    left=10.88.40.97
    #leftsubnet=10.88.40.96/27

    right=10.88.40.130
    #rightsubnet=10.88.40.128/27
```

```
[root@localhost ipsec.d]# cat /etc/ipsec.d/ipsec_iptables_config
# Generated by iptables-save v1.4.7 on Thu Jan 18 17:32:12 2018
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p esp -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p esp -j ACCEPT
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p udp -m udp --dport 500 -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p udp -m udp --dport 500 -j ACCEPT
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p udp -m udp --dport 4500 -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p udp -m udp --dport 4500 -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-port-unreachable
COMMIT
# Completed on Thu Jan 18 17:32:12 2018
```

```
[root@localhost ipsec.d]# cat /etc/ipsec.d/ipsec_fconfig.dump
eth5      Link encap:Ethernet HWaddr 00:50:56:23:6C:C4
          inet addr:10.88.40.97  Bcast:10.88.40.127  Mask:255.255.255.224
          inet6 addr: fe80::250:56ff:fe23:6cc4/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:10969 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4282 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:770189 (752.1 KiB)  TX bytes:302051 (294.9 KiB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:4226 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4226 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:186016 (181.6 KiB)  TX bytes:186016 (181.6 KiB)
```

```
[root@localhost ipsec.d]# cat /etc/ipsec.d/ipsec_look.dump
localhost.localdomain Thu Jan 18 18:26:56 CET 2018
IPSEC TABLE
ROUTING TABLE
10.88.40.96/27 dev eth5 proto kernel scope link src 10.88.40.97 metric 1
10.88.40.128/27 via 10.88.40.98 dev eth5 proto static
default via 10.88.40.98 dev eth5 proto static
```

```
[root@localhost ipsec.d]# cat /etc/ipsec.d/ipsec_route.dump
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
10.88.40.96     0.0.0.0        255.255.255.224 U     1      0        0 eth5
10.88.40.128    10.88.40.98   255.255.255.224 UG    0      0        0 eth5
0.0.0.0          10.88.40.98   0.0.0.0        UG    0      0        0 eth5
```

Files from Hamburg:

```
[root@pnid4-mid-hh ipsec.d]# ifconfig > ipsec_fconfig.dump
[root@pnid4-mid-hh ipsec.d]# ipsec look > ipsec_look.dump
[root@pnid4-mid-hh ipsec.d]# route -n > ipsec_route.dump
```

```
[root@pnid4-mid-hh ipsec.d]# cat /etc/ipsec.d/psk.secrets
10.88.40.97 10.88.40.130 : PSK 0x08ba6fd0_212c2a15_fa671f53_16fc4b87_dc231639_479d3714_b3f46bb5_58dc7fc2
[root@pnid4-mid-hh ipsec.d]#
```

```
[root@pnid4-mid-hh ipsec.d]# cat /etc/ipsec.d/psk.conf
conn psk
  type=tunnel
  auto=add
  authby=secret

  left=10.88.40.97
  #leftsubnet=10.88.40.96/27

  right=10.88.40.130
  #rightsubnet=10.88.40.128/27
[root@pnid4-mid-hh ipsec.d]#
```

```
[root@pnid4-mid-hh ipsec.d]# cat /etc/ipsec.d/ipsec_iptables_config
# Generated by iptables-save v1.4.7 on Thu Jan 18 17:32:12 2018
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p esp -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p esp -j ACCEPT
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p udp -m udp --dport 500 -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p udp -m udp --dport 500 -j ACCEPT
-A FORWARD -s 10.88.40.130/32 -d 10.88.40.97/32 -p udp -m udp --dport 4500 -j ACCEPT
-A FORWARD -s 10.88.40.97/32 -d 10.88.40.130/32 -p udp -m udp --dport 4500 -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp-port-unreachable
COMMIT
# Completed on Thu Jan 18 17:32:12 2018
```

```
[root@pnid4-mid-hh ipsec.d]# cat /etc/ipsec.d/ipsec_fconfig.dump
eth5      Link encap:Ethernet HWaddr 00:50:56:2B:56:BC
          inet addr:10.88.40.130 Bcast:10.88.40.159 Mask:255.255.255.224
          inet6 addr: fe80::250:56ff:fe2b:56bc/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:12882 errors:0 dropped:0 overruns:0 frame:0
            TX packets:294 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1207838 (1.1 MiB) TX bytes:57575 (56.2 KiB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:2533 errors:0 dropped:0 overruns:0 frame:0
            TX packets:2533 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:128404 (125.3 KiB) TX bytes:128404 (125.3 KiB)
```

```
[root@pnid4-mid-hh ipsec.d]# cat /etc/ipsec.d/ipsec_look.dump
pnid4-mid-hh.localdomain Thu Jan 18 18:09:01 CET 2018
IPSEC TABLE
ROUTING TABLE
10.88.40.96/27 via 10.88.40.129 dev eth5 proto static
10.88.40.128/27 dev eth5 proto kernel scope link src 10.88.40.130 metric 1
default via 10.88.40.129 dev eth5 proto static
[root@pnid4-mid-hh ipsec.d]#
```

```
[root@pnid4-mid-hh ipsec.d]# cat /etc/ipsec.d/ipsec_route.dump
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
10.88.40.96    10.88.40.129   255.255.255.224 UG    0      0      0 eth5
10.88.40.128   0.0.0.0        255.255.255.224 U      1      0      0 eth5
0.0.0.0         10.88.40.129   0.0.0.0        UG    0      0      0 eth5
[root@pnid4-mid-hh ipsec.d]#
```

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