

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	5
Exercise 6: Managing Services (Please use pnidX-svr-mu	5
Exercise 7: Configure the following network (figure 1) using ifconfig and route add	14
Exercise 8: Configure the following network (figure 1) using ip and nmcli . .	14
Exercise 9: Configure the following network (figure 1) using GUI	14
Part 4: Network Scanning	17
Exercise 1: Configure the networks of figure 1	18
Exercise 2: NMAP	21
Exercise 3: Nessus network device identification	26
Exercise 4: OpenVAS Network device identification	34
Part 5: Sniffing, Virtual Private Network (VPN)	35
Exercise 1: Configure and set the networks shown below (figure1 and 2) . .	35
Exercise 2: Getting started with network monitoring tools	35
Exercise 3: TCPDUMP	35
Exercise 4: Wireshark	35
Exercise 5: Experimenting with network monitoring tools	42
Exercise 6: Set up a host-to-host VPN using preshared key	43
Exercise 7: Set up a host-to-host VPN using RSA keys	43
Exercise 8: Set up a network-to-network VPN using preshared key	43
Exercise 9: Set up a network-to-network VPN using RSA secrets keys . . .	43

Abbildungsverzeichnis

Abbildung 1: aktivierte bzw. deaktivierte Dienste eines runlevels	6
Abbildung 2: Runlevel, in denen iptables eingeschaltet bzw. ausgeschaltet sind	6
Abbildung 3:	7
Abbildung 4: Runlevel 2,3,4,5 werden deaktiviert	7
Abbildung 5: chkconfig iptables on off	7
Abbildung 6: vi /etc/yum.repos.d/local.repo	8
Abbildung 7: mounten	8
Abbildung 8: yum install tftp	8
Abbildung 9: yum install tftp	9
Abbildung 10: service xinetd start	9
Abbildung 11:	9
Abbildung 12:	10
Abbildung 13:	11
Abbildung 14: yum install vsftpd	11
Abbildung 15: Runlevel 2 von vsftpd wird deaktiviert	12
Abbildung 16:	12
Abbildung 17:	12
Abbildung 18:	12
Abbildung 19:	12
Abbildung 20:	13
Abbildung P4 figure 1 LAN	17
Abbildung P5 ex. 1 Zenmap Subnetz 64	18
Abbildung P5 ex. 1 Zenmap Subnetz 96	19
Abbildung P5 ex. 1 Zenmap Subnetz 128	19
Abbildung P5 ex. 1 Zenmap Subnetz 160	20
Abbildung P5 ex. 1 Zenmap Subnetz 32	20

Abbildung P5 ex. 1 Zenmap alle Subnetze	21
Abbildung P4 ex. 2 nmap command 1	22
Abbildung P4 ex. 2 nmap command 2	22
Abbildung P4 ex. 2 nmap externes Logfile	23
Abbildung P4 ex. 2 nmap command 3	23
Abbildung P4 ex. 2 nmap command 4	24
Abbildung P4 ex. 2 nmap command 5	24
Abbildung P4 ex. 2 nmap command 6	24
Abbildung P4 ex. 2 nmap command 7	25
Abbildung P4 ex. 2 nmap command 8	25
Abbildung P4 ex. 2 nmap command 9	26
Abbildung P4 ex. 2 nmap command 10	26
Abbildung P4 ex. 3 installation Nessus	29
Abbildung P4 ex. 3 Konfiguration Nessus	30
Abbildung P4 ex. 3 Registrierung Nessus	31
Abbildung P4 ex. 3 Aktivierung Nessus	31
Abbildung P4 ex. 3 License Nessus	32
Abbildung P5 ex. 4 Wireshark Filter 1	36
Abbildung P5 ex. 4 Wireshark Filter 2	36
Abbildung P5 ex. 4 Wireshark Filter 3	37
Abbildung P5 ex. 4 Wireshark Filter 4	37
Abbildung P5 ex. 4 Wireshark Filter 5	38
Abbildung P5 ex. 4 Wireshark Filter 7	39
Abbildung P5 ex. 4 Wireshark Filter 8	40
Abbildung P5 ex. 4 Wireshark ftp Login Passwort	40
Abbildung P5 ex. 4 Wireshark Filter 9	41
Abbildung P5 ex. 4 Wireshark ssh Datenpaket	42
Abbildung P5 ex. 5 nmap offene Ports anzeigen	42
Abbildung P5 ex. 5 Telnet login	43

Part 3: Network Troubleshooting Utilities

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

Please type and explain the meaning of the following commands:

1) # chkconfig

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

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

```
[root@localhost ~]# chkconfig
NetworkManager 0:off 1:off 2:on 3:on 4:on 5:on 6:off
abrt-ccpp 0:off 1:off 2:off 3:on 4:off 5:on 6:off
abrtdaemon 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
audited 0:off 1:off 2:on 3:on 4:on 5:on 6:off
autoofs 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
nfslock 0:off 1:off 2:off 3:on 4:on 5:on 6:off
ntpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
ntpdate 0:off 1:off 2:off 3:off 4:off 5:off 6:off
oddjobd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
portreserve 0:off 1:off 2:on 3:on 4:on 5:on 6:off
postfix 0:off 1:off 2:on 3:on 4:on 5:on 6:off
psacct 0:off 1:off 2:off 3:off 4:off 5:off 6:off
quota_nld 0:off 1:off 2:off 3:off 4:off 5:off 6:off
rdisc 0:off 1:off 2:off 3:off 4:off 5:off 6:off
[...]
[root@localhost ~]
```

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@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
LocalRepository LocalRepo
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package tftp.x86_64 0:0.49-7.el6 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

=====
| Package           | Arch      | Version        | Repository | Size |
=====
| Installing:      |          |               |            |       |
| tftp              | x86_64   | 0.49-7.el6    | LocalRepo  | 32 K |
=====

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

Abbildung 8: yum install tftp

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

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

Dependencies Resolved

=====
| Package           | Arch   | Version      | Repository | Size |
|=====             | ====== | ======       | ======     | ===== |
| Installing:      |        |              |            |       |
| xinetd           | x86_64 | 2:2.3.14-39.el6_4 | LocalRepo | 121 k |
|=====             |        |              |            |       |
| Transaction Summary |          |              |            |       |
|=====             |        |              |            |       |
| Install 1 Package(s) |          |              |            |       |
|=====             |        |              |            |       |
| Total download size: 121 k
| Network bandwidth usage: 259 k
| Is this ok [y/N]: y
| Downloading Packages:
|   Running rpm_check_debug
|   Running transaction test
|   Transaction test succeeded
|   Running transaction
|     Installing : 2:xinetd-2.3.14-39.el6_4.x86_64
|     Verifying  : 2:xinetd-2.3.14-39.el6_4.x86_64
|=====             |        |              |            |       |
| Installed:        |        |              |            |       |
| xinetd.x86_64 2:2.3.14-39.el6_4 |          |              |            |       |
|=====             |        |              |            |       |
| Complete!         |        |              |            |       |
|=====             |        |              |            |       |
```

Abbildung 9: yum install xinetd

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

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

Abbildung 10: service xinetd start

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

Abbildung 20

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

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

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

Part 4: Network Scanning

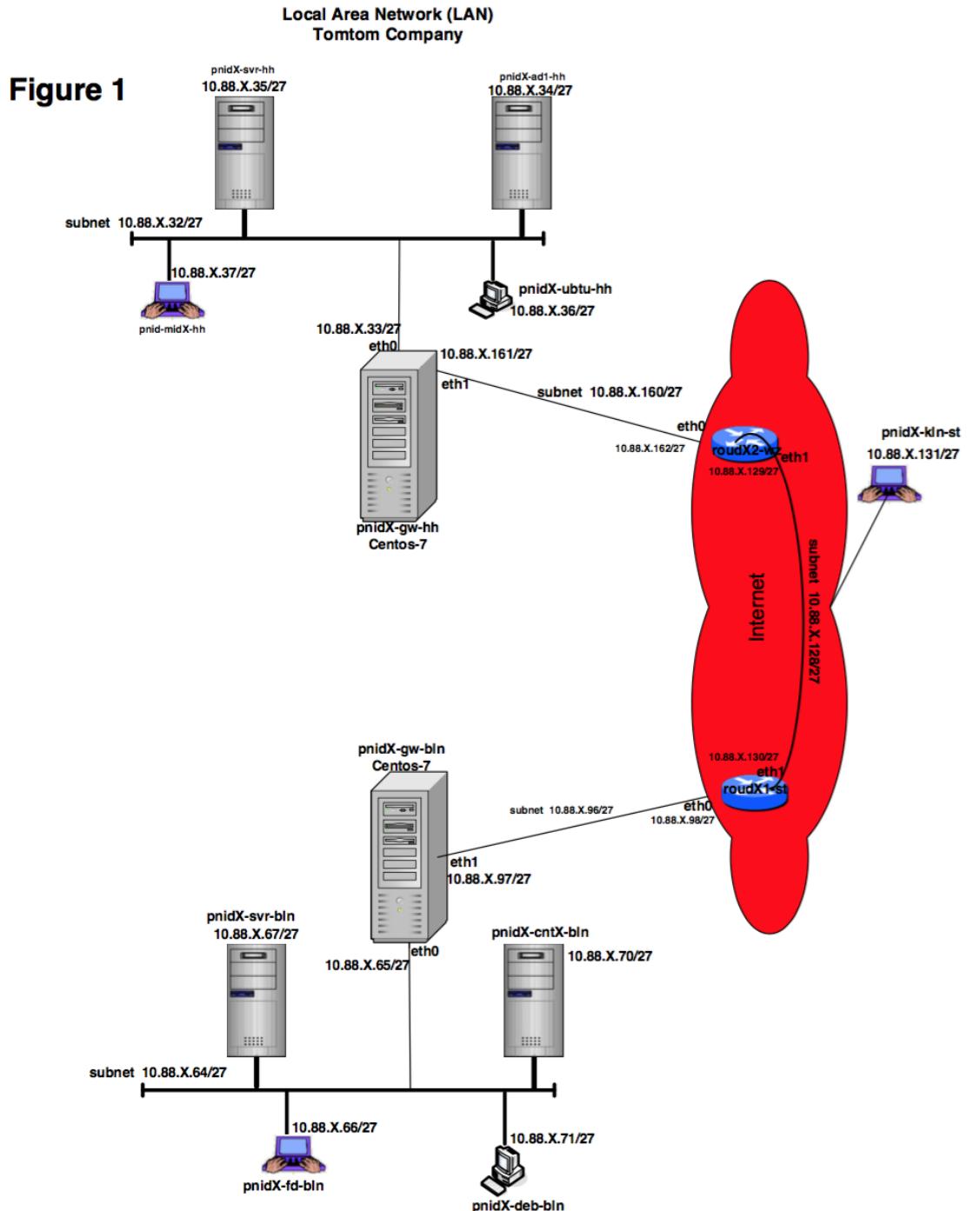


Abbildung 21: Netzwerk mit allen Hosts und Subnetzen

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

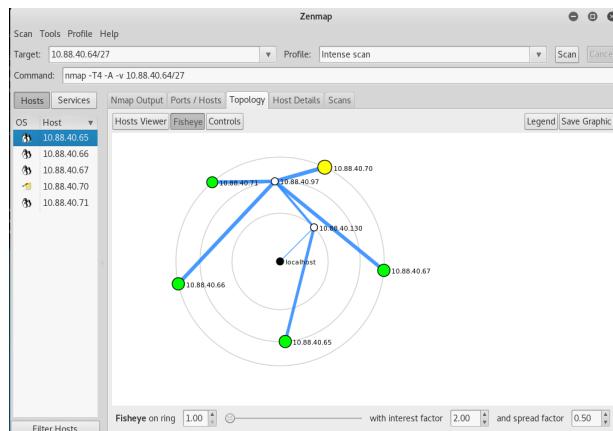


Abbildung 22: Subnetz 10.88.40.64/27

Scan des Subnetzes 10.88.40.96/27

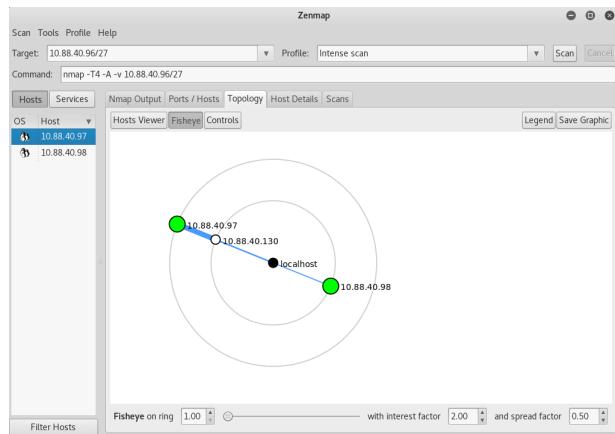


Abbildung 23: Subnetz 10.88.40.96/27

Scan des Subnetzes 10.88.40.128/27

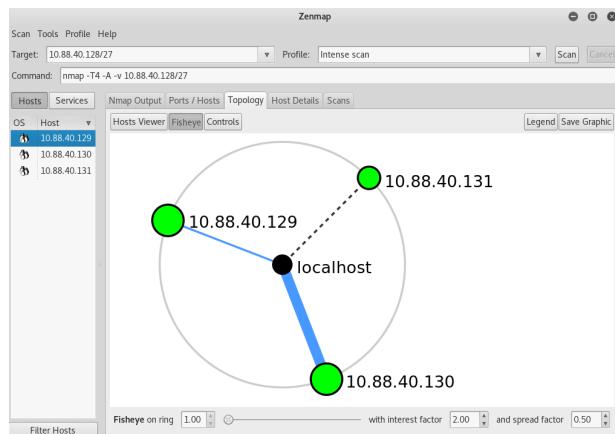


Abbildung 24: Subnetz 10.88.40.128/27

Scan des Subnetzes 10.88.40.160/27

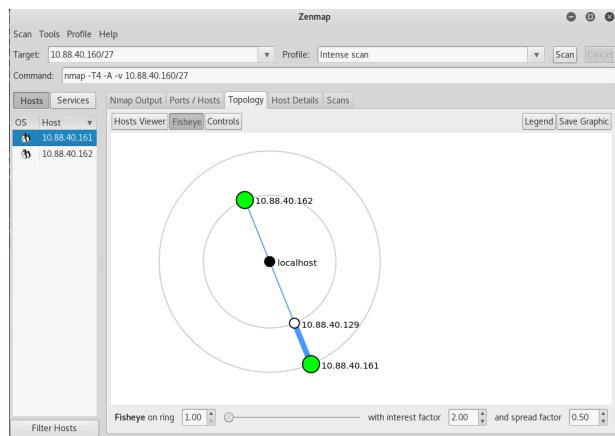


Abbildung 25: Subnetz 10.88.40.160/27

Scan des Subnetzes 10.88.40.32/27

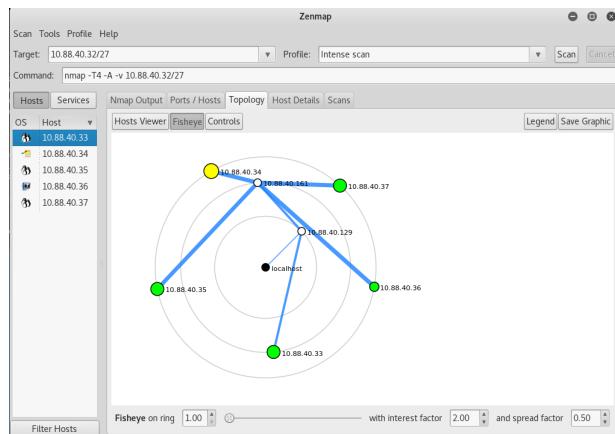


Abbildung 26: Subnetz 10.88.40.32/27

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

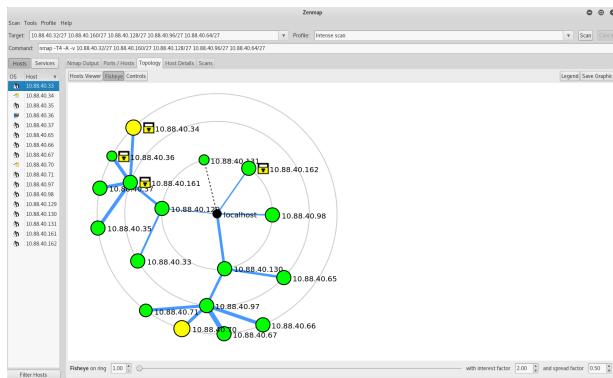


Abbildung 27: Alle Subnetze

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
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid
servers with --dns-servers
Nmap scan report for 10.88.40.130
Host is up (0.00062s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
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 28: 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:~#
File Edit View Search Terminal Help
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
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 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 29: 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 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 (0.000910s latency).
All 1000 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

```

Plain Text ▾ Tab Width: 8 ▾ Ln 17, Col 97 ▾ INS

Abbildung 30: 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 31: `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 OS info not available
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 32: 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 33: 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 34: 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 35: 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 36: nmap -sP -PS53 10.88.40.130/27

Question 9: Please type and explain the nmap command: nmap -sS -v 10.88.X.?/25

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 37: 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 38: 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

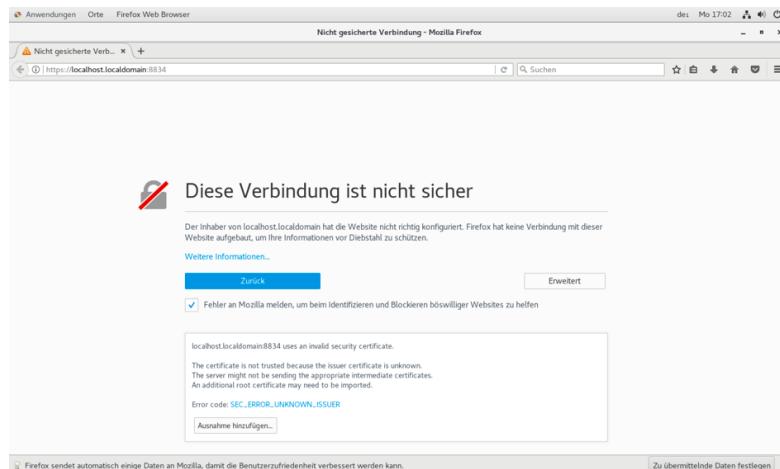
Um die Binary zu installieren wird der Befehl: `rpm -Uvh /home/hanifka/Schreibtisch/Nessus-6.11.2-es7.x86_64.rpm` im Terminal ausgeführt.

```
[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

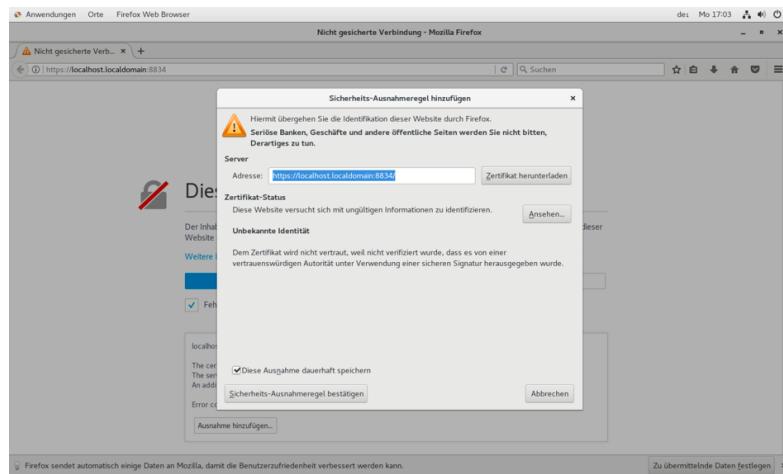
Processing the Nessus plugins...
[########################################] [100%]
All plugins loaded (1sec)
- You can start Nessus by typing /bin/systemctl start nessusd.service
- Then go to https://localhost.localdomain:8834/ to configure your scanner
[root@localhost ~]#
```

Abbildung 39: 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.

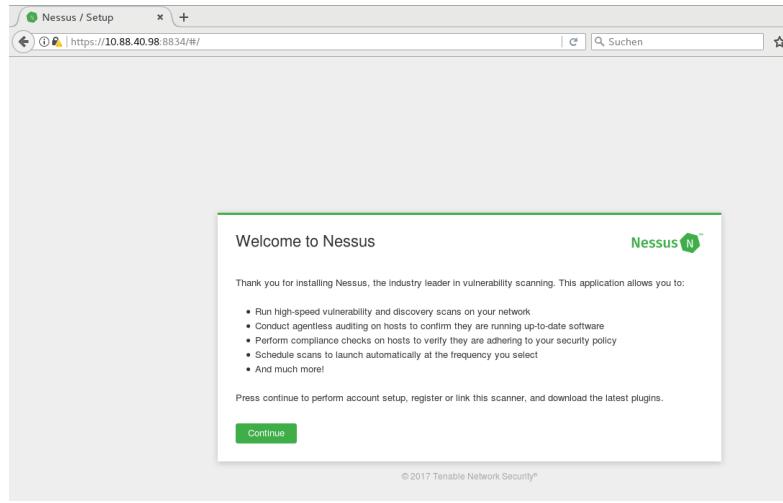


Abbildung 40: Nessus Konfiguration Startseite

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.

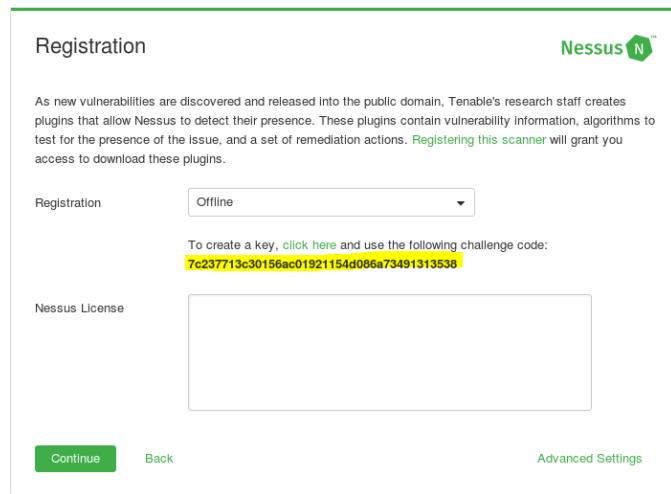


Abbildung 41: Nessus Registrierung

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.

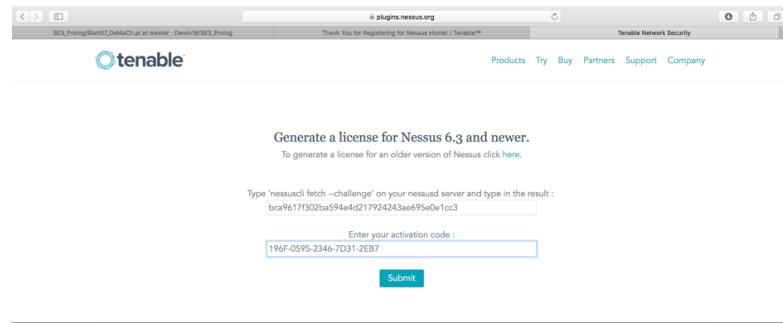


Abbildung 42: Nessus Aktivierungskey

Nach dem Klicken auf Submit erhalten wir unsere Nessus License.

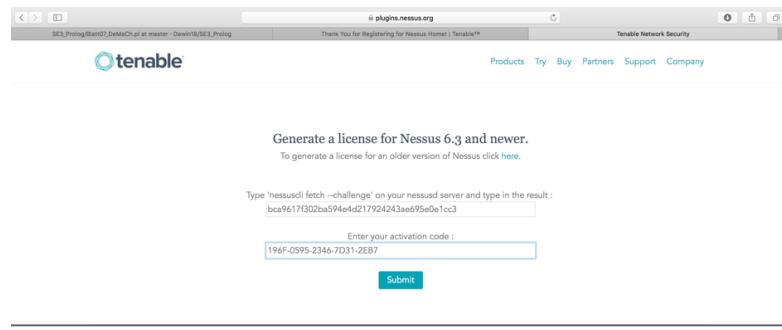
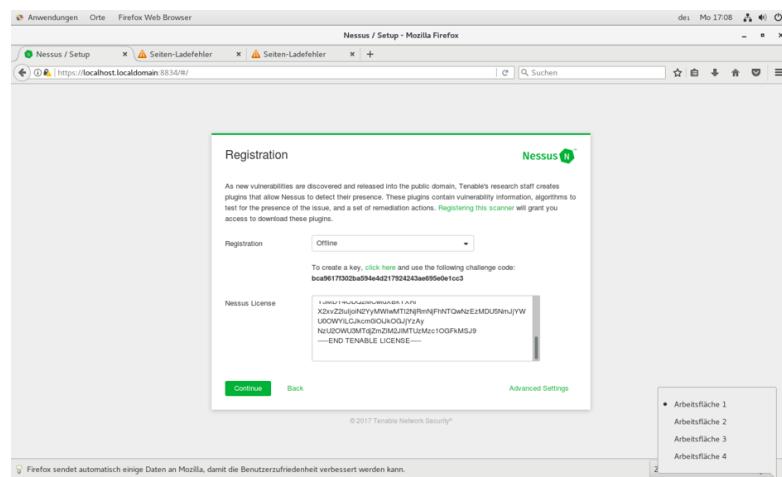
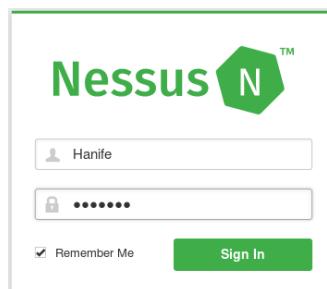


Abbildung 43: Nessus License

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.

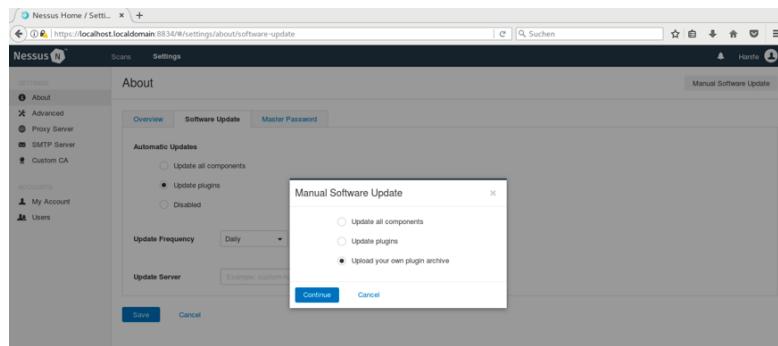


Solution of 4) Install the plugins

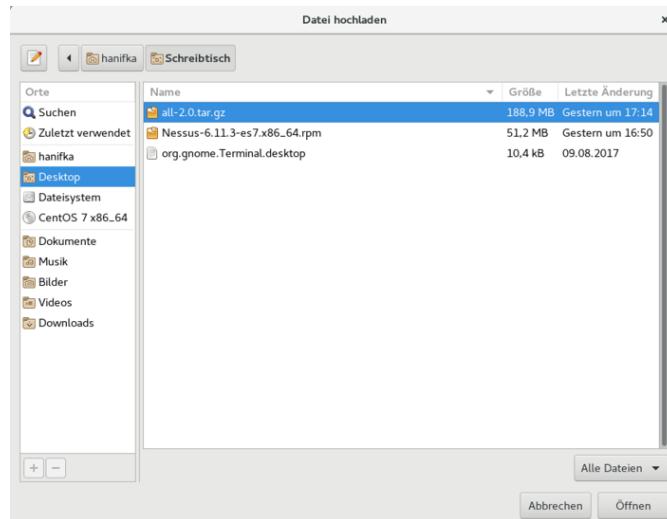
Um die Plugins zu installieren, laden wir die notwendige Datei all-2.0.tar.gz runter und kopieren diese anschließend in die virtuelle Machine. Dort kann sie wie folgt installiert werden übers Terminal oder manuell über die GUI.

```
[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)

Solution of 6)

Solution of 7)

Exercise 4: OpenVAS Network device identification

Part 5: Sniffing, Virtual Private Network (VPN)

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

Exercise 2: Getting started with network monitoring tools

Exercise 3: TCPDUMP

Exercise 4: Wireshark

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

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

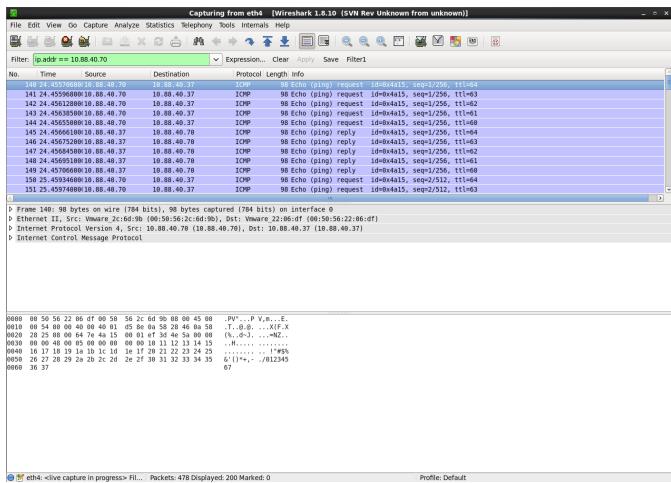


Abbildung 44: Wireshark Filter $\text{ip.addr} == 10.88.40.70$

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: $\text{ip.addr} == 10.88.40.70 \&& \text{frame.len} > 400$. Mit dem Teil $\text{frame.len} > X$ können wir die Datenpakete nach Bytegröße X Filtern. Für X gilt, $X < 2^{32} \wedge X \in \mathbb{N}$.

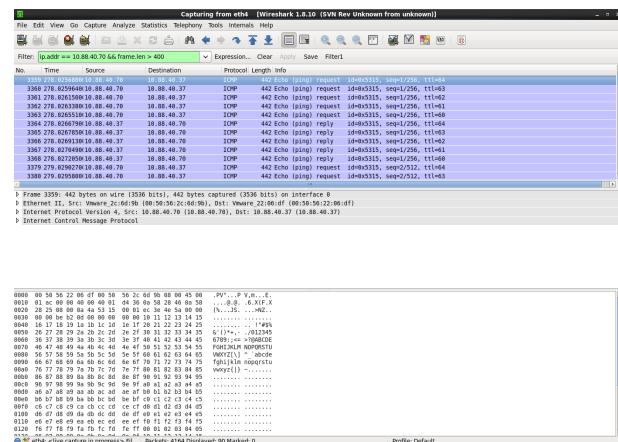


Abbildung 45: Wireshark Filter $\text{ip.addr} == 10.88.40.70 \&& \text{frame.len} > 400$

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 46: 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.

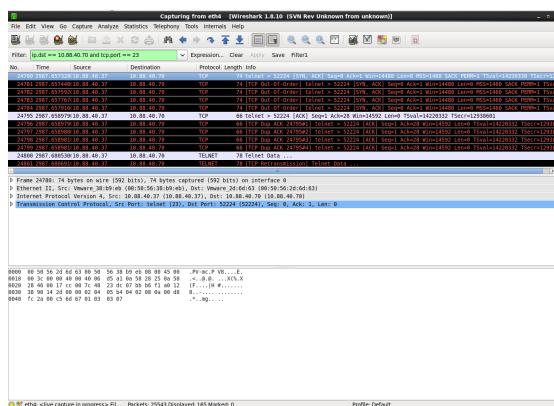


Abbildung 47: Wireshark Filter $ip.dst == 10.88.40.70 \text{ and } tcp.port == 23$

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.

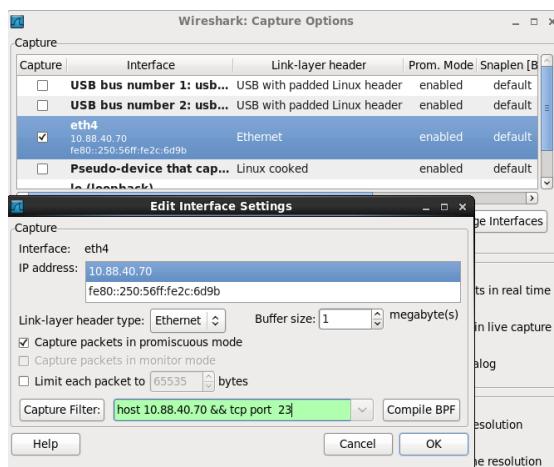


Abbildung 48: Wireshark Filter host 10.88.40.70 && tcp port 23

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.

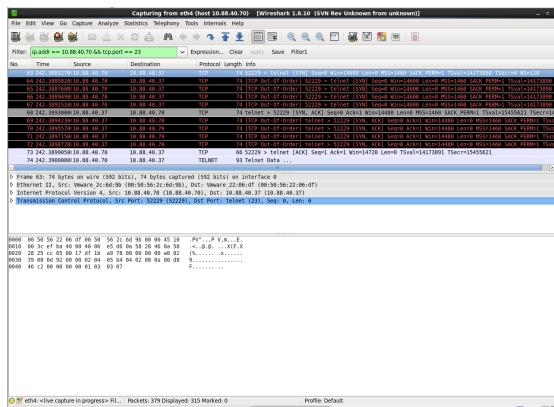


Abbildung 49: Wireshark Filter ip.addr == 10.88.40.70 && tcp.port == 23

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.

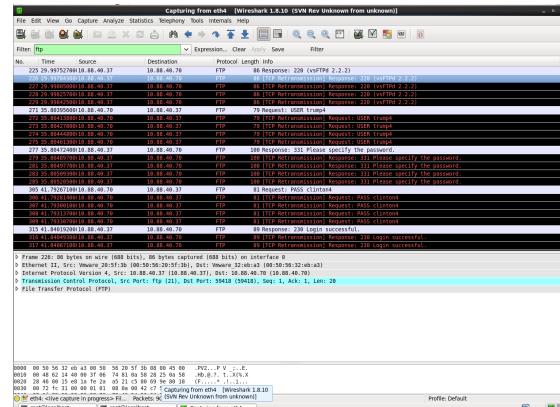


Abbildung 50: Wireshark Filter ftp

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

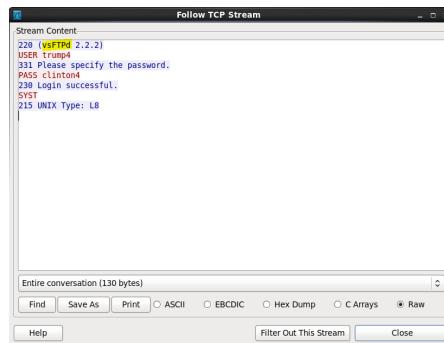


Abbildung 51: Wireshark ftp Login Passwort

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.

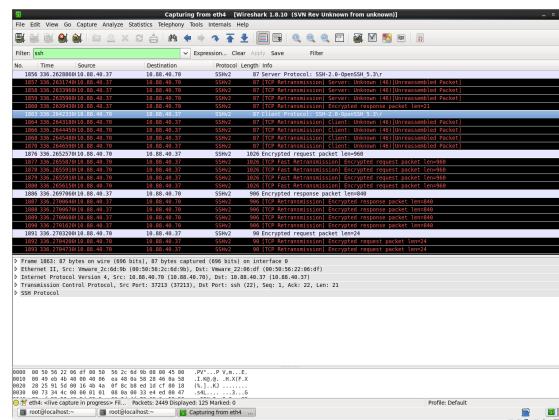


Abbildung 52: Wireshark Filter ssh

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

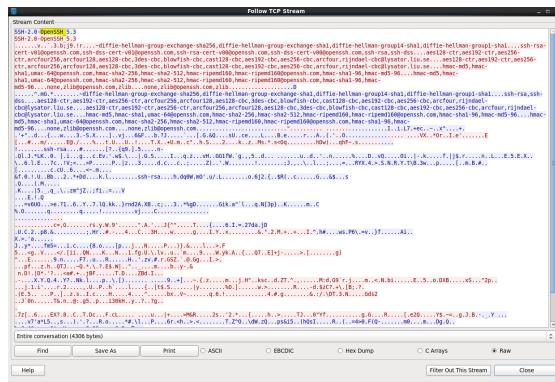


Abbildung 53: Wireshark verschlüsselte Datenfragmente

Exercise 5: Experimenting with network monitoring tools

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

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

```
[root@localhost ~]# nmap -sF -o 10.88.40.37
Starting Nmap 5.51 ( http://nmap.org ) at 2018-01-04 14:33 CET
nmap: warning: unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify
--dns-servers.
Nmap scan report for 10.88.40.37
Host is up (0.0025s latency).
Not shown: 997 closed ports
PORT      STATE      SERVICE
21/tcp    open|filtered  ftp
22/tcp    open|filtered  ssh
23/tcp    open|filtered  telnet
80/tcp    open|filtered  http
113/tcp   open|filtered  rpdbind
Too many fingerprints match this host to give specific OS details
Network Distance: 4 hops
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.02 seconds
[root@localhost ~]#
```

Abbildung 54: 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 55: 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

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