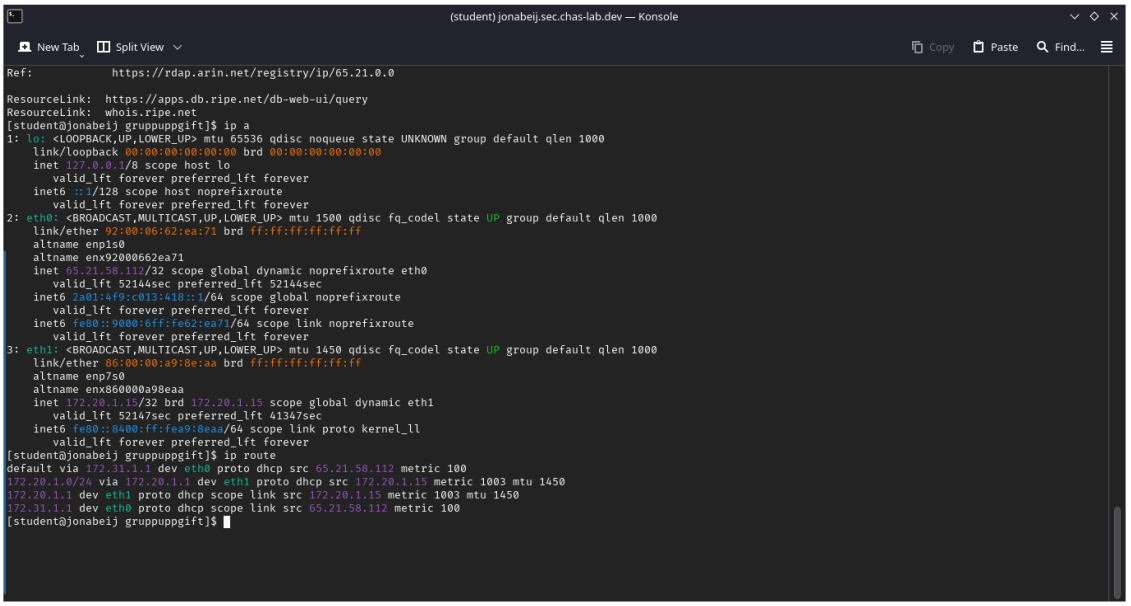
Virtuell
l nätverksmiljö med Wireshark-analys och säkerhetsrapport

Inledning

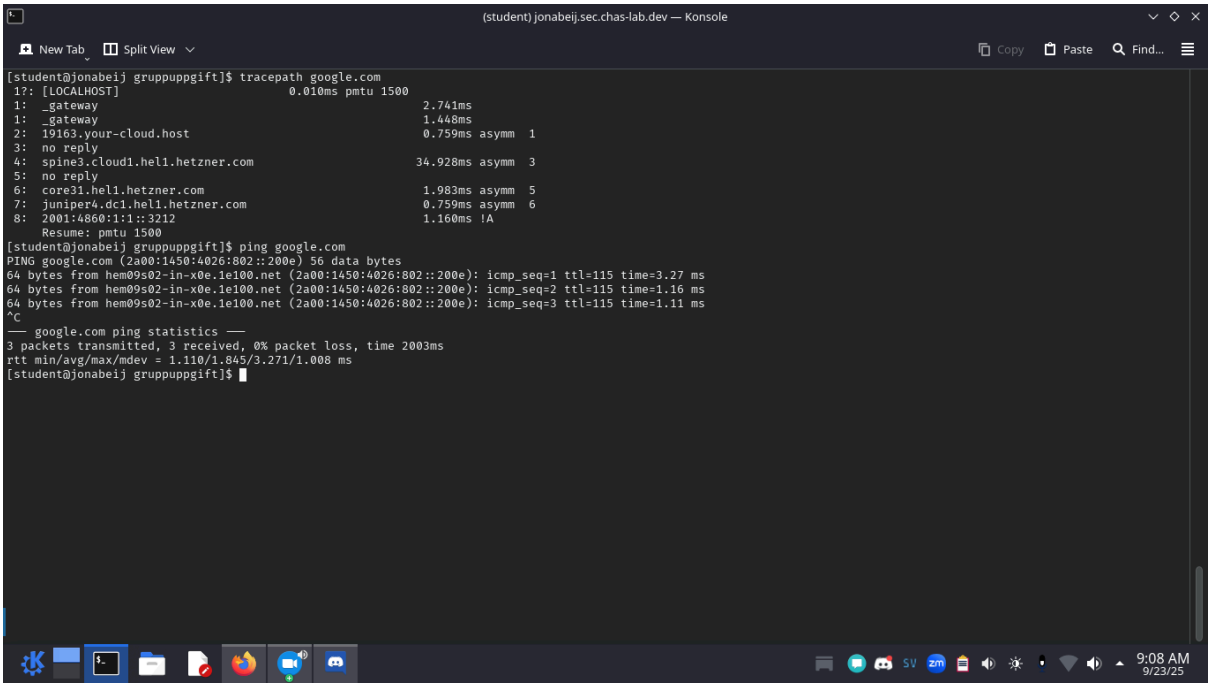
Syftet med laborationen är att praktiskt träna på nätverkssäkerhet i en servermiljö genom att simulera en brandväggsserver och en loggserver i ett gemensam internnät. Genom övningen får man arbeta med grundläggande nätverksanalys, brandväggskonfiguration med firewalld, central loggning med rsyslog samt insamling och analys av nätverkstrafik med tcpdump och Wireshark. Laborationen ger även möjlighet att reflektera över hur dessa komponenter används för att stärka säkerheten utifrån CIA triaden och andra relevanta säkerhetsprinciper. Arbetet genomförs i grupp med tydliga roller och avslutas med en rapport och en kort guide som ska kunna följas av en ny kollega.

Moment 1 - Nätverksgrunder

Jonas Ip adresser:
ip a + ip route



tracepath google.com + ping google.com



Abdi Ip adresser:
ip a + ip route

```
[student@abdiabdi ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 92:00:06:62:e9:e6 brd ff:ff:ff:ff:ff:ff
    altname enp1s0
    altname enx92000662e9e6
    inet 37.27.221.183/32 scope global dynamic noprefixroute eth0
       valid_lft 69713sec preferred_lft 69713sec
    inet6 2a01:4f9:c012:9993::1/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::9000:6ff:fe62:e9e6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc fq_codel state UP group default qlen 1000
    link/ether 86:00:00:a9:8e:d7 brd ff:ff:ff:ff:ff:ff
    altname enp7s0
    altname enx860000a98ed7
    inet 172.20.1.12/32 brd 172.20.1.12 scope global dynamic eth1
       valid_lft 69717sec preferred_lft 58917sec
    inet6 fe80::8400:ff:fea9:8ed7/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
[student@abdiabdi ~]$ ip route
default via 172.31.1.1 dev eth0 proto dhcp src 37.27.221.183 metric 100
172.20.1.0/24 via 172.20.1.1 dev eth1 proto dhcp src 172.20.1.12 metric 1003 mtu 1450
172.20.1.1 dev eth1 proto dhcp scope link src 172.20.1.12 metric 1003 mtu 1450
172.31.1.1 dev eth0 proto dhcp scope link src 37.27.221.183 metric 100
[student@abdiabdi ~]$
```

Tracepath Abdi ip adress

```
[student@abdiabdi ~]$ tracepath 65.21.58.112
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              3.229ms
 1:  _gateway                                              2.168ms
 2:  14178.your-cloud.host                                 0.396ms asymm  1
 3:  no reply
 4:  spine2.cloud1.hel1.hetzner.com                       10.353ms asymm  3
 5:  no reply
 6:  spine3.cloud1.hel1.hetzner.com                        3.914ms asymm  5
 7:  no reply
 8:  19163.your-cloud.host                                 0.633ms asymm  7
 9:  jonabeij.sec.chas-lab.dev                             0.995ms !H
     Resume: pmtu 1500
[student@abdiabdi ~]$
```

Erik A Ip adresser:
ip a + ip route

```
[student@erikaldu ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 92:00:06:62:eb:03 brd ff:ff:ff:ff:ff:ff
    altname enp1s0
    altname enx92000662eb03
    inet 46.62.145.93/32 scope global dynamic noprefixroute eth0
       valid_lft 52942sec preferred_lft 52942sec
    inet6 2a01:4f9:c012:9a3f::1/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::9000:6ff:fe62:eb03/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc fq_codel state UP group default qlen 1000
    link/ether 86:00:00:a9:8e:7c brd ff:ff:ff:ff:ff:ff
    altname enp7s0
    altname enx860000a98e7c
    inet 172.20.1.21/32 brd 172.20.1.21 scope global dynamic eth1
       valid_lft 52946sec preferred_lft 42146sec
    inet6 fe80::8400:ff:fea9:8e7c/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
[student@erikaldu ~]$ ip route
default via 172.31.1.1 dev eth0 proto dhcp src 46.62.145.93 metric 100
172.20.1.0/24 via 172.20.1.1 dev eth1 proto dhcp src 172.20.1.21 metric 1003 mtu 1450
172.20.1.1 dev eth1 proto dhcp scope link src 172.20.1.21 metric 1003 mtu 1450
172.31.1.1 dev eth0 proto dhcp scope link src 46.62.145.93 metric 100
[student@erikaldu ~]$
```

ping + tracepath google.com

```
[student@erikaldu ~]$ ping google.com
PING google.com (2a00:1450:4026:803::200e) 56 data bytes
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=1 ttl=116 time=1.88 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=2 ttl=116 time=0.995 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=3 ttl=116 time=0.873 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=4 ttl=116 time=0.858 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=5 ttl=116 time=0.853 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=6 ttl=116 time=0.839 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=7 ttl=116 time=0.840 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=8 ttl=116 time=0.883 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=9 ttl=116 time=0.895 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=10 ttl=116 time=0.884 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=11 ttl=116 time=0.899 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=12 ttl=116 time=0.876 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=13 ttl=116 time=0.887 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=14 ttl=116 time=0.874 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=15 ttl=116 time=0.909 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=16 ttl=116 time=0.957 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=17 ttl=116 time=0.944 ms
64 bytes from hem09s03-in-x0e.1e100.net (2a00:1450:4026:803::200e): icmp_seq=18 ttl=116 time=0.878 ms
^C
--- google.com ping statistics ---
18 packets transmitted, 18 received, 0% packet loss, time 17350ms
rtt min/avg/max/mdev = 0.839/0.945/1.880/0.229 ms
[student@erikaldu ~]$ tracepath google.com
 1?: [LOCALHOST]                      0.008ms pmtu 1500
 1:  _gateway                                              1.928ms
 1:  _gateway                                              1.456ms
 2:  16632.your-cloud.host                                 0.515ms asymm  1
 3:  no reply
 4:  spine4.cloud1.hel1.hetzner.com                       28.014ms asymm  3
 5:  no reply
 6:  core32.hel1.hetzner.com                               0.708ms asymm  5
 7:  juniper4.dc1.hel1.hetzner.com                         0.915ms asymm  6
 8:  2001:4860:1:1::3212                                   1.127ms !A
     Resume: pmtu 1500
```

Adnan ip adresser:

ip a + ip route

```
[student@adnaabdi ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 92:00:06:62:eb:0f brd ff:ff:ff:ff:ff:ff
    altname enp1s0
    altname enx92000662eb0f
    inet 65.21.240.84/32 scope global dynamic noprefixroute eth0
       valid_lft 58110sec preferred_lft 58110sec
    inet6 2a01:4f9:c013:9932::1/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::9000:6ff:fe62:eb0f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc fq_codel state UP group default qlen 1000
    link/ether 86:00:00:a9:8e:71 brd ff:ff:ff:ff:ff:ff
    altname enp7s0
    altname enx860000a98e71
    inet 172.20.1.22/32 brd 172.20.1.22 scope global dynamic eth1
       valid_lft 58116sec preferred_lft 47316sec
    inet6 fe80::8400:ff:fea9:8e71/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
[student@adnaabdi ~]$
```

ping + tracepath google.com

```
[student@adnaabdi ~]$ ping google.com
PING google.com (2a00:1450:4026:802::200e) 56 data bytes
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=1 ttl=115 time=1.81 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=2 ttl=115 time=1.26 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=3 ttl=115 time=1.23 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=4 ttl=115 time=1.19 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=5 ttl=115 time=1.21 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=6 ttl=115 time=1.20 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=7 ttl=115 time=1.20 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=8 ttl=115 time=1.22 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=9 ttl=115 time=1.16 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=10 ttl=115 time=1.17 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=11 ttl=115 time=1.23 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=12 ttl=115 time=1.18 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=13 ttl=115 time=1.16 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=14 ttl=115 time=1.16 ms
64 bytes from hem09s02-in-x0e.1e100.net (2a00:1450:4026:802::200e): icmp_seq=15 ttl=115 time=1.23 ms
^C
--- google.com ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14020ms
rtt min/avg/max/mdev = 1.160/1.240/1.806/0.153 ms
[student@adnaabdi ~]$ tracepath google.com
 1?: [LOCALHOST]                      0.034ms pmtu 1500
 1:  _gateway                                              2.765ms
 1:  _gateway                                              1.710ms
 2:  14125.your-cloud.host                                 0.602ms asymm  1
 3:  no reply
 4:  spine2.cloud1.hel1.hetzner.com                        2.657ms asymm  3
 5:  no reply
 6:  core32.hel1.hetzner.com                               2.427ms asymm  5
 7:  juniper4.dc1.hel1.hetzner.com                         0.722ms asymm  6
 8:  no reply
 9:  2001:4860:1:1::3212                                   1.263ms !A
     Resume: pmtu 1500
[student@adnaabdi ~]$
```

Daniels ip adresser:
ip a + ip route

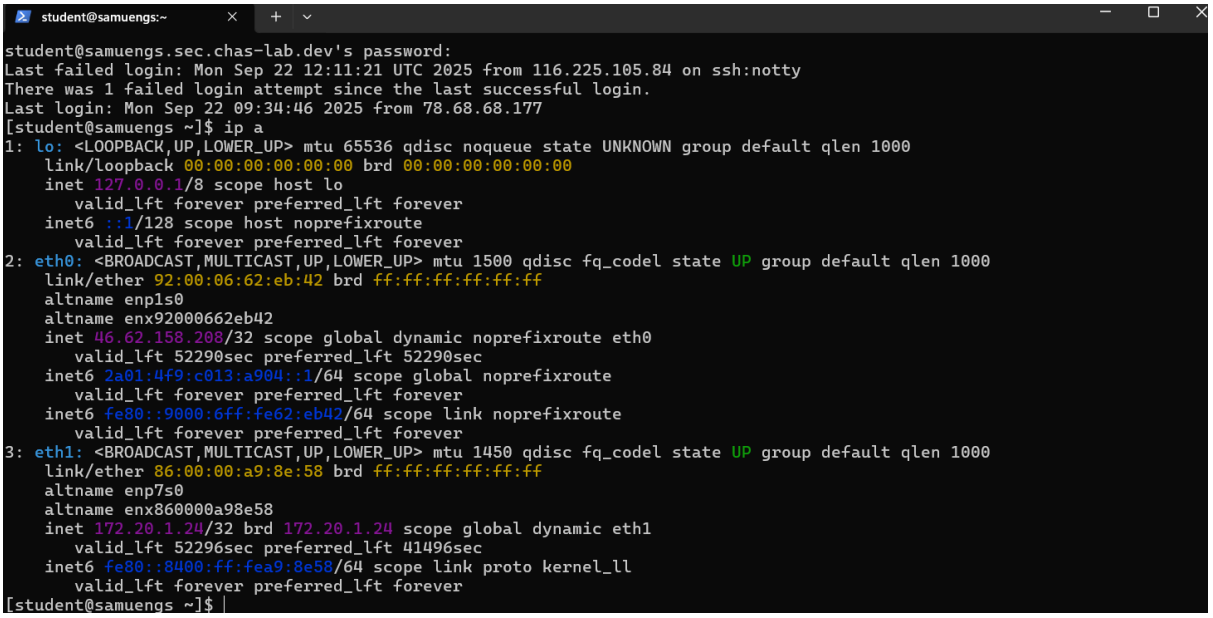
```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 92:00:06:62:ea:53 brd ff:ff:ff:ff:ff:ff
    altname enp1s0
    altname enx92000662ea53
    inet 37.27.204.46/32 scope global dynamic noprefixroute eth0
       valid_lft 51392sec preferred_lft 51392sec
    inet6 2a01:4f9:c013:af8c::1/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::9000:6ff:fe62:ea53/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc fq_codel state UP group default qlen 1000
    link/ether 86:00:00:a9:8e:b2 brd ff:ff:ff:ff:ff:ff
    altname enp7s0
    altname enx860000a98eb2
    inet 172.20.1.14/32 brd 172.20.1.14 scope global dynamic eth1
       valid_lft 51397sec preferred_lft 40597sec
    inet6 fe80::8400:ff:fea9:8eb2/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
-bash: iproute: command not found
[student@daniwage ~]$
```

Publik IP: 37.27.204.46
Intern IP: 172.20.1.14

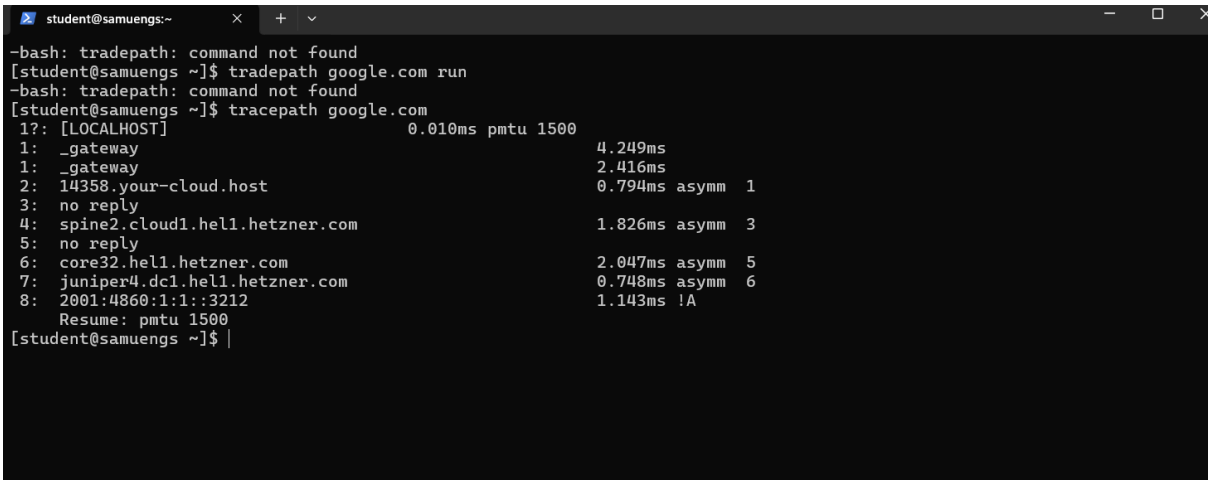
Ping google.com + tracepath google.com

```
PING google.com (2a00:1450:4026:808::200e) 56 data bytes
64 bytes from hem08s10-in-x0e.1e100.net (2a00:1450:4026:808::200e): icmp_seq=1 ttl=116 time=2.79 ms
64 bytes from hem08s10-in-x0e.1e100.net (2a00:1450:4026:808::200e): icmp_seq=2 ttl=116 time=1.02 ms
64 bytes from hem08s10-in-x0e.1e100.net (2a00:1450:4026:808::200e): icmp_seq=3 ttl=116 time=1.09 ms
64 bytes from hem08s10-in-x0e.1e100.net (2a00:1450:4026:808::200e): icmp_seq=4 ttl=116 time=0.996 ms
64 bytes from hem08s10-in-x0e.1e100.net (2a00:1450:4026:808::200e): icmp_seq=5 ttl=116 time=1.04 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.996/1.387/2.793/0.703 ms
 1?: [LOCALHOST]                      0.007ms pmtu 1500
 1:  _gateway                                              3.841ms
 1:  _gateway                                              1.753ms
 2:  13477.your-cloud.host                                 0.444ms asymm  1
 3:  no reply
 4:  spine1.cloud1.hel1.hetzner.com                       12.439ms asymm  3
 5:  no reply
 6:  core31.hel1.hetzner.com                               1.016ms asymm  5
 7:  juniper4.dc1.hel1.hetzner.com                         0.813ms asymm  6
 8:  2001:4860:1:1::3212                                   1.248ms !A
     Resume: pmtu 1500
[student@daniwage ~]$
```

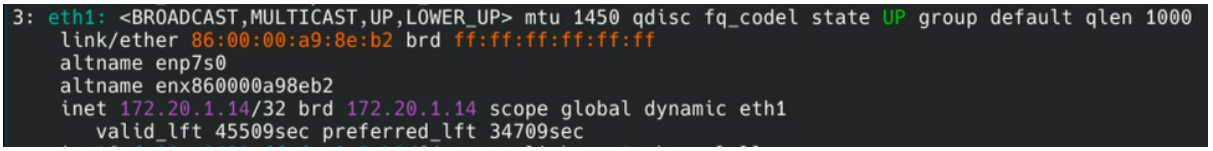
Samuels ip adresser:
ip a + ip route



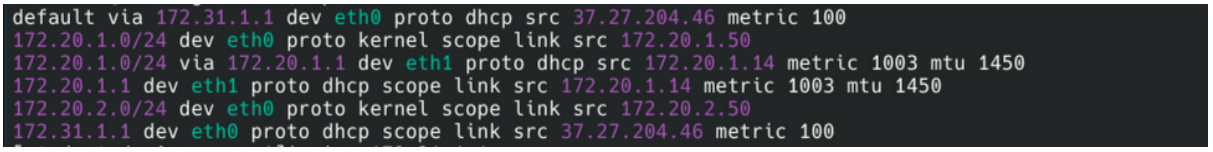
Ping google.com + tracepath google.com



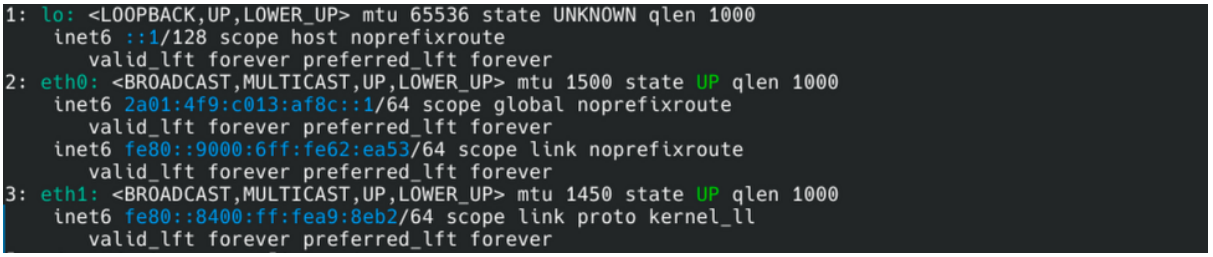
Vi har gått igenom våra student servrar lite med de olika kommandon som uppgiften önskade. Med ip a kan vi se att vår externa IP-adress ligger under interfacet eth0, och vår interna IP finns under interfacet eth1. Det kan vi utläsa för att dessa två IP-adresser avslutas med /32, som innebär att det bara är en specifik IP-adress. Då är det “min” IP.

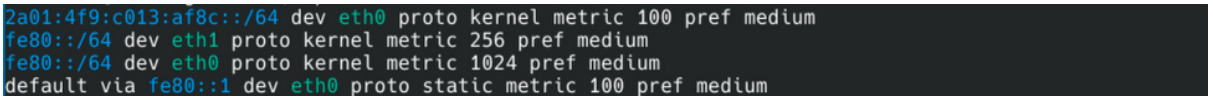


Med ip route kan vi se att vi (i det här fallet jag, textförfattaren) är på det interna nätverket 172.20.1.0 med subnetmasken 255.255.255.0 (/24). Om vi tolkar resultatet i övrigt rätt är den externa routern ut mot internet på IP 172.31.1.1.



Det här bör innebära att min server är på nätverket 172.20.1.0 och har IP 172.20.1.14. På nätverket finns det i så fall plats för 254 användbara värdadressen.
Om vi provar ip -6 a och ip -6 route så kan vi dra slutsatsen att IPv6 kan användas för kommunikation externt, men det används inte på det interna nätverket.





Kommunikation

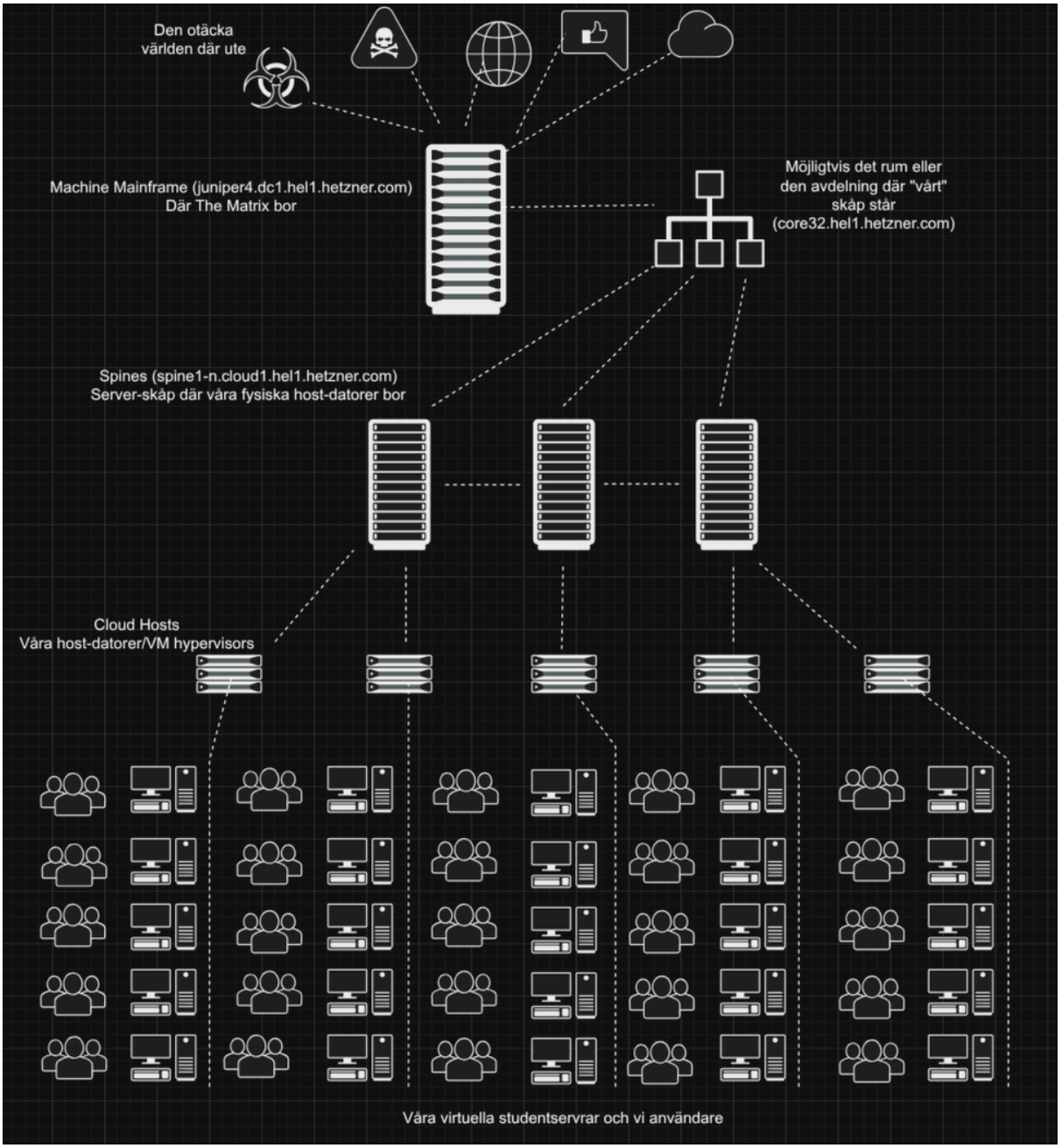
```
[student@daniwage ~]$ ping -c5 -W1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=2.34 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=0.955 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=0.988 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=0.960 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=1.07 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.955/1.262/2.338/0.539 ms
[student@daniwage ~]$ tracepath 8.8.8.8
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              3.804ms
 1:  _gateway                                              1.489ms
 2:  13477.your-cloud.host                                 0.367ms asymm  1
 3:  no reply
 4:  spine1.cloud1.hel1.hetzner.com                        5.581ms asymm  3
 5:  no reply
 6:  core32.hel1.hetzner.com                               1.829ms asymm  5
 7:  juniper4.dc1.hel1.hetzner.com                         0.917ms asymm  6
 8:  142.251.195.234                                       1.102ms asymm  7
 9:  no reply
10:  no reply
^C
[student@daniwage ~]$
```

På tracepath kan vi se att vi börjar på vår virtuella server och skickar därifrån över signalen till den fysiska dator som kör vår virtuella maskin. Därifrån går vi vidare till “spine”, som kan vara routern på den server-rack som håller i vår fysiska host. Nästa steg är en router med namnet “core”. Det kan tänkas vara det rum eller den avdelning som vår server-rack tillhör. Därefter kommer signalen till en sista router innan internet, “juniper”. När signalen sedan släpps ut skickas vi iväg till Google.
Om vi kör tracepath till några av våra klasskamrater så ser vi att vissa går över till en annan “spine”-router. Det kan betyda att deras server är hostad i en annan server i ett annat rack, men det kanske mer troligt är att serverhallen lastbalanserar mitt tracepath-kommando.

```
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              4.323ms
 1:  _gateway                                              1.649ms
 2:  13477.your-cloud.host                                 0.401ms asymm  1
 3:  no reply
 4:  spine1.cloud1.hel1.hetzner.com                       20.681ms asymm  3
 5:  no reply
 6:  14178.your-cloud.host                                 1.879ms asymm  5
 7:  abdiabdi.sec.chas-lab.dev                             2.355ms !H
```

```
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              3.043ms
 1:  _gateway                                              2.203ms
 2:  13477.your-cloud.host                                 0.391ms asymm  1
 3:  no reply
 4:  spine1.cloud1.hel1.hetzner.com                       19.159ms asymm  3
 5:  no reply
 6:  spine4.cloud1.hel1.hetzner.com                       10.581ms asymm  5
 7:  no reply
 8:  27834.your-cloud.host                                 2.352ms asymm  7
 9:  abdufkhi.sec.chas-lab.dev                             3.297ms reached
```

Nätverksskiss



Reflektionsfrågor:

Publik vs. intern IP

Vad är skillnaden mellan en publik och en intern IP-adress?

Skillnaden är att en publik IP-adress är unik och synlig på internet för att identifiera ditt nätverk utåt, medan en intern (eller privat) IP-adress är unik inom ditt lokala nätverk och tilldelas av din router för att kommunicera med andra enheter i samma nätverk. Routern använder sedan sin publika adress för att vidarebefordra trafiken mellan internet och dina interna enheter.

Varför används interna IP-adresser i er labbmiljö (172.20.1.0/24)?

Privata ip-adresser kan inte nås direkt från det publika nätverket som gör att labben blir skyddade från obehörig åtkomst och attacker utifrån. För att skapa en isolerad och säker miljö. Interna ip-adresser gör att studenter kan träna på nätverkskonfiguration utan att exponera servar direkt mot internet. Det minskar risken från riktiga attacker utifrån.

I en verklig kommunal IT-miljö: hur kan man använda interna nät för att minska exponeringen mot internet?

Man kan placera interna system, exempelvis personalsystem, ekonomisystem i interna nät som inte är direkt åtkomliga från internet. Endast vissa tjänster exponeras via brandväggar eller proxyservrar. Detta minskar attackytan och skyddar känslig information.

Subnetting och säkerhet

Vad betyder subnetting, och varför delar man upp nät i mindre delar?

Det betyder att dela upp ett större nätverk i flera mindre delar (subnät) med egna nät-ID och broadcastadresser. Det gör att man kan använda adresser mer effektivt och kontrollera trafiken bättre.

Hur kan subnetting minska risken för attacker eller begränsa spridning vid intrång? Ge ett konkret exempel.

Man kan säga om en angripare får tillgång till en dator i ett nät kan subnetting begränsa spridningen. exempelvis om en kommun har ett nät för administrativa system och ett separat nät för elever/lärare, så hindrar subnetting att en angripare som tar över en elevdator automatiskt får åtkomst till känsliga administrativa server.

TCP/IP-modellen i praktiken

Identifiera på vilken nivå (lager) som *ping* och *tracepath*

Ping använder icmp (internet control message protocol), vilket hör till nätverkslagret lagar 3.

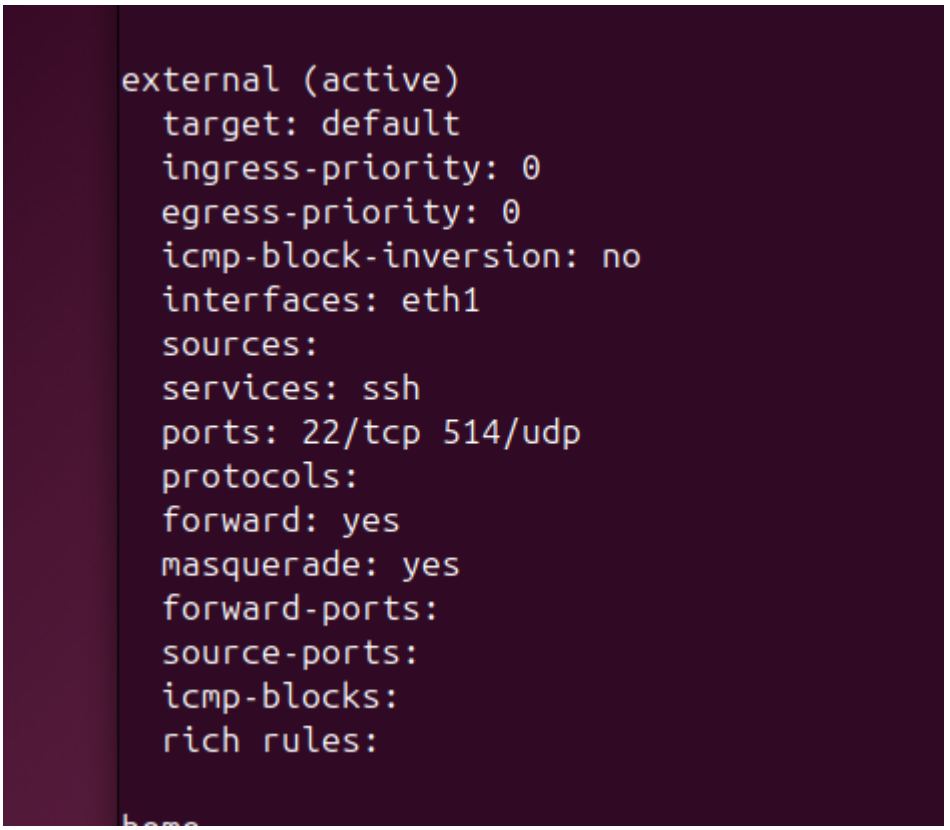
Tracepath använder också icmp och ibland UDP och ligger därmed också på nätverkslagret lagar 3.

Hur hjälper förståelsen av TCP/IP-modellen er att förstå varför vissa paket syns i *tcpdump* och andra inte?

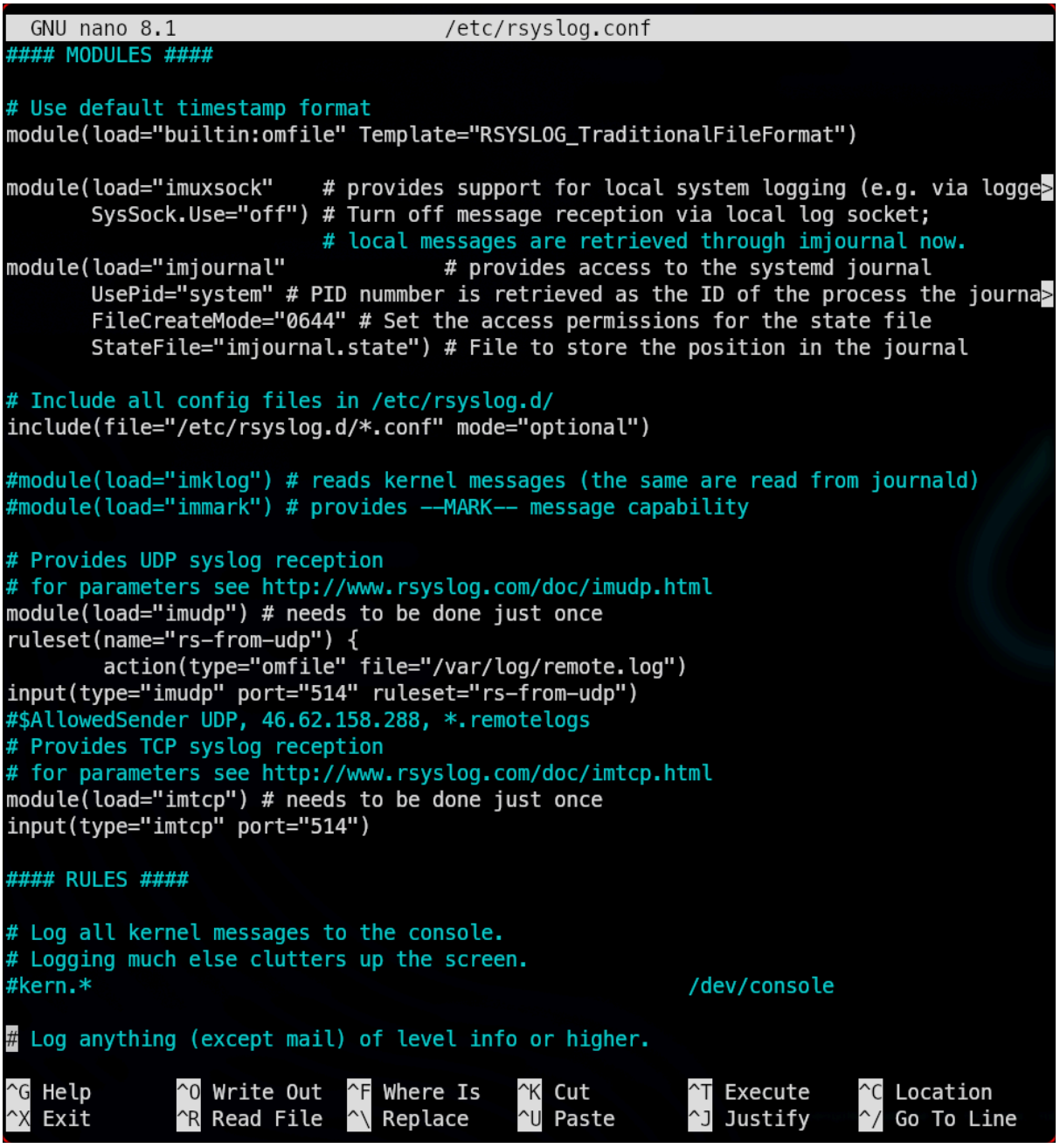
Man kan säga att TCP/IP modellen visar att olika protokoll arbetar på olika lagar. När man kör tcpdump kan man välja att lyssna på vissa protokoll till exempel TCP, UDP, ICMP. Om man filtrerar på TCP så ser man inte ICMP ping eftersom det ligger på nätverkslagret. Förståelsen av modellen gör det lättare att tolka varför vissa paket syns och varför andra inte.

Moment 2 - Brandvägg & loggning

1. Brandväggen är en del av säker konfiguration och rsyslog är en logghantering åtgärd.
2. Drop innebär att paketet stoppas och att man inte skickar in reply till sändaren. Det är en säker standardinställning som inte släpper in något alls men den styrkan blir en svaghet om man hamnar på andra sidan av zonen.
3. Man kan skicka meddelande genom logger med olika prioriteringsnivåer beroende på hur allvarligt situationen är. I loggarna ser man tydligt vad som är normal aktivitet mot skum. Blev extra tydligt vid den simulerade DDoS-attacken då hela loggen fylldes med försök direkt från samma IP.
4. Med firewallD så använder man en sak som kallas zoner där man har lagt till portar så att datorn kan skicka loggar till en annan server och blockera oönskad trafik från andra portar som inte är öppna att ta emot trafik vilket gör den mer avancerad medans UFW är en enklare brandvägg som inte är lika kraftfullt,
5. Våra loggar innehåller all sorts information från alla källor. Genom att aktivt analysera dom kan man dela in informationen och får mer vettig information från dom
6. På stora system finns det inte en chans att man hinner med att kolla på all aktivitet helt enkelt på grund av mängden så man måste implementera någon form av larm som hjälper en att sålla.



Servern har en loggkonfiguration som bestämmer vad som ska loggas, hur loggarna ska se ut och vart de ska skickas.

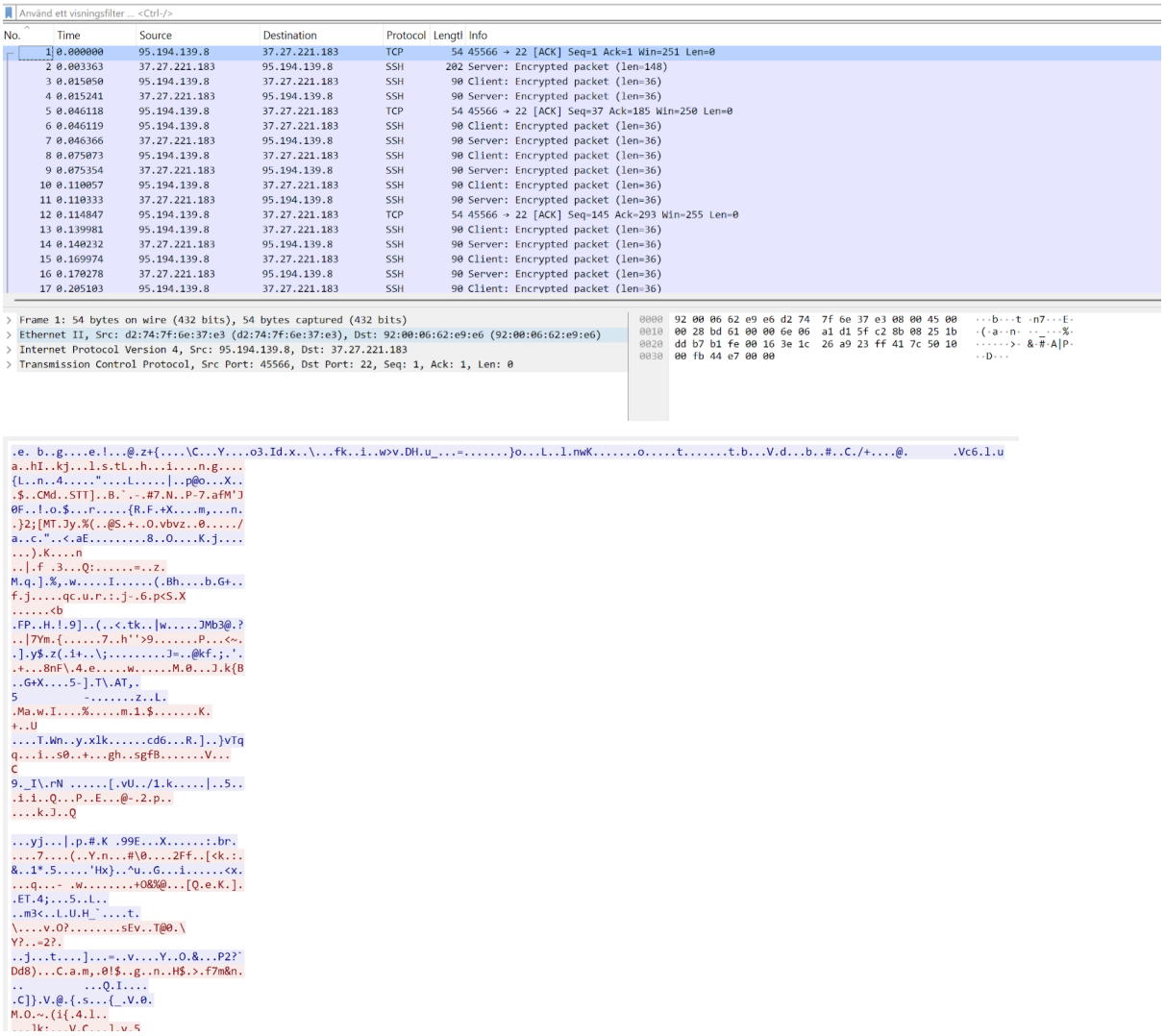


Moment 3 – Trafikanalys & rapport

Samla in trafik med tcpdump och analysera i Wireshark (port 22)

Med tcpdump -i eth0 port 22 -w ssh.pcap fångades trafiken vid inloggning via SSH.
Analys: I Wireshark syns först en TCP-handshake, därefter märkt som Encrypted packet. Ingen nyttodata läsas i klartext. Detta bekräftar att SSH använder kryptering.

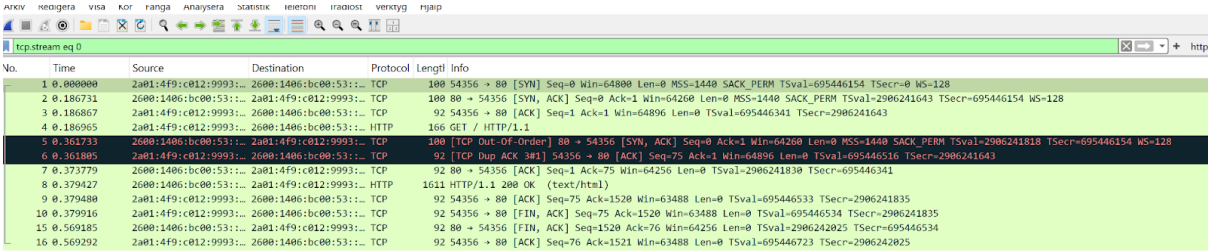
Skärmbild av ssh.pcap:



Analysera okrypterad HTTP-trafik. (Port 80)

Med tcpdump -i eth0 port 80 -w http.pcap och curl http://example.com genererades HTTP-trafik.
Analys: I Wireshark syns en HTTP GET-förfrågan samt svaret från servern HTML-koden för sidan syns i klartext, vilket betyder att HTTP inte skyddar kommunikationen.

Skärmbild av http.pcap:



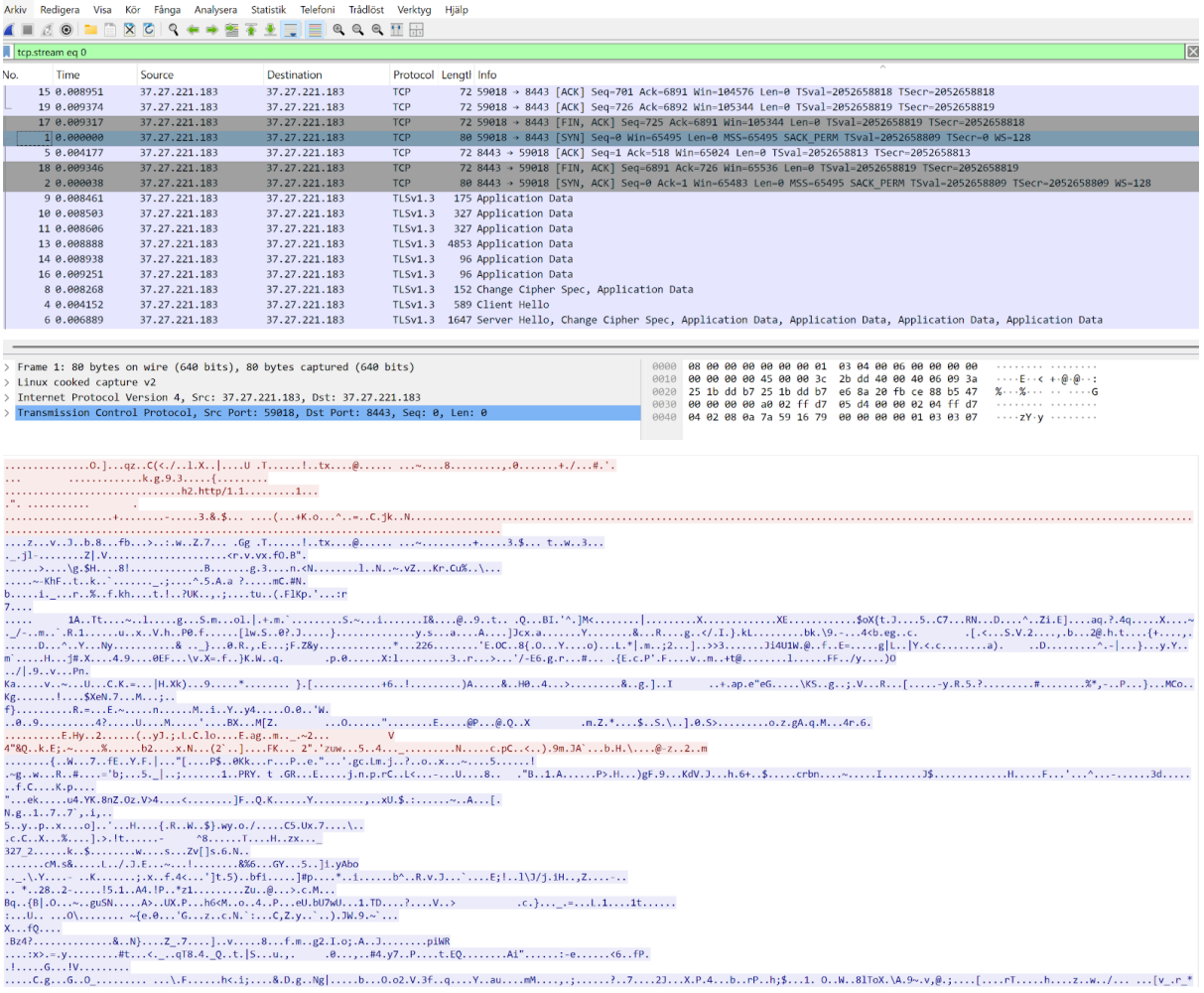

```
GET / HTTP/1.1
Host: example.com
User-Agent: curl/8.9.1
Accept: */*


HTTP/1.1 200 OK
Content-Type: text/html
ETag: "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"
Last-Modified: Mon, 13 Jan 2025 20:11:20 GMT
Cache-Control: max-age=86000
Date: Wed, 01 Oct 2025 16:09:39 GMT
Content-Length: 1256
Connection: keep-alive

<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
        color: #38488f;
```

Trafikkryptering med TLS
(Port 8443, TLS)

En självsignerade certifikatfil (cert.pem och key.pem) skapades med OpenSSL. Därefter startades en HTTPS-server:

Analys: I Wireshark syns TLS-handshake (Klient Hello och Server Hello) följt av Encrypted Application Data.

Skärmbild av https.pcap:



Diskussion

Varför behövs HTTPS?

För att skydda data mot avlyssning. Med HTTPS kan känslig information (t.ex. lösenord och personuppgifter) skickas säkert över internet.

Vad händer med självsignerat certifikat?

Klienten varnar för att certifikatet inte är betrott. Även om trafiken är krypterad kan man inte vara säker på serverns identitet.

Hur påverkar det säkerheten och användarna?

Självsignerade certifikat kan lura oerfarna användare, men i professionell drift används alltid certifikat från en betrodd CA. På så sätt får man både kryptering och identifiering.

Jämförelse mellan SSH och HTTP

Analysen visar tydligt skillnaden mellan okrypterad och krypterad trafik.

SSH och HTTPS skyddar kommunikationen, vilket innebär att innehållet inte kan avläsas av en angripare. Däremot HTTP skickar allt i klartext och är därför osäkert att använda i

praktiken. Ett giltigt certifikat är viktigt för att undvika varningar och garantera tillit, men även ett självsignerat certifikat krypterar trafiken.

Moment 4 – Simulerad DoS-attack och motåtgärder

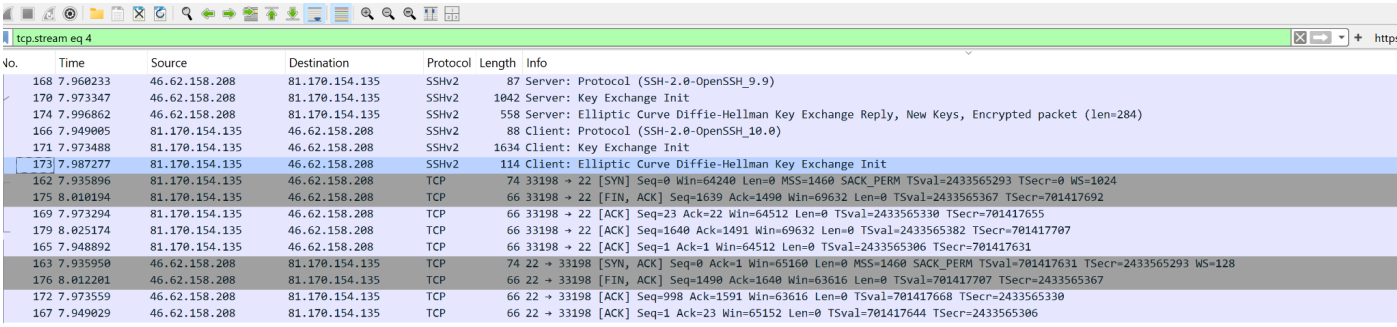
1. Simulera en attack
2. Observera i loggar

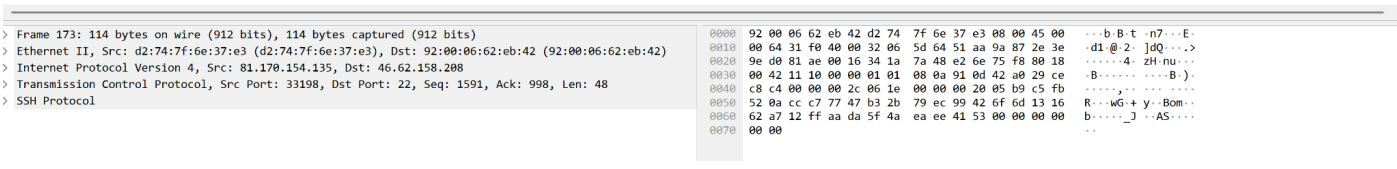
```
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51644 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51646 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51656 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51672 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51682 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51690 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51706 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51710 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51716 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51726 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51728 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51742 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51756 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51772 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51780 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51784 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51798 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51810 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51824 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51836 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51840 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51848 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51856 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51858 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51866 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51882 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51896 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51912 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51914 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51916 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:36 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51920 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51924 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51926 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51930 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51944 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51960 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51972 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51974 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51976 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51982 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51990 on [46.62.158.208]:22 penalty: failed authentication
Oct  3 08:10:37 samuengs sshd[842]: drop connection #0 from [81.170.154.135]:51992 on [46.62.158.208]:22 penalty: failed authentication
```

sudo tail -f remote.log | grep sshd

sudo tcpdump -i eth0 port 22 -w ssh.pcap

3. Sökte i loggar för att hitta rader som visar droppade paket.
4. Analysera i tcpdump/Wireshark
5. 5. Pcap-filen och öppna i Wireshark. Identifiera mönstret (många likadana ICMP-paket på kort tid).





Reflektion – CIA-triaden, säkerhetsprinciper och förbättringsförslag

Arbetet präglades till stor del av konfidentialitet, eftersom fokus låg på att utestänga obehöriga användare med hjälp av brandväggen och att säkerställa innehållet på servern genom loggar som lagras på flera servrar. Vi uppmärksammade även vikten av att känna till trafikvägarna, då dessa kan förändras beroende på belastning. Att ha kontroll över informationsvägarna är avgörande för både spårbarhet och felsökning.

Flera av säkerhetsåtgärderna, särskilt inom brandväggskonfigurationen, grundades i principen om "least privilege". Detta innebär att endast nödvändiga portar hålls öppna. I vårt fall tilläts endast SSH (port 22) och UDP (port 514), medan övriga portar blockerades för att minimera riskerna för intrång.

Kort guide till en nyanställd kollega som ska återupprepa samma installation

Det man vill förmedla till en nyanställd är att man borde läsa igenom vad Firewalld är med vad zoner innebär i programmet. Sen att man läser hur man kan ändra innehållet i dom och vad olika ändringar gör, En annan sak som rekommenderas att läsa om är vad konfigurationer är för något i rsyslogs så att man vet vad anledningen till att man har dom för att få servrar att prata med varandra. Läser man om de områdena innan man börjar arbetet så blir arbetet enklare då man förstår vad varje del är till för vilket gör att man kan få det att fungera snabbare.

Våra tre viktigaste lärdomar

Det kommer komma förhinder i arbetet, även där man inte tror.

Samarbete ska funka fint för att underlätta för alla. Att boka tider och komma på dem.

Att man inte kan vara expert på allt och lita på sina kollegors kompetens.

Arbetsfördelning

Projektledare/tidshållare (Abdi)

Nätverksspecialist (IP, routing, subnetting) (Daniel)

Brandväggsansvarig (firewalld på firewall-node) (Samuel)

Loggansvarig (rsyslog på log-node) (Jonas)

Trafikanalytiker (tcpdump/Wireshark) (Abdi)

Dokumentationsansvarig (sammanställer rapporten) (Erik)