Rapport

Introduktion till Linux och små nätverk, 7,5 hp
VT-2025

Projektuppgift

Laboration 6: Serverar och brandvägg

av

Sixten Peterson (050402-XXXX)

Akademin för teknik och miljö
Avdelningen för industriell utveckling, IT och samhällsbyggnad

Högskolan i Gävle
S-801 76 Gävle, Sweden

Datorpost:

education@snicon.rip

Innehållsförteckning

1. Inledning

Laborationen är den sjätte och sista som utförs i kursen, och omfattar fyra deluppgifter för att undersöka hur portskanning och brandväggskonfiguration går till. Först kontrolleras vilka portar som är öppna, därefter installeras tjänsterna apache2 och NFS, slutligen begränsas åtkomsten till portarna med hjälp av ufw.

1.1. Bakgrund

Detta är den sjätte och sista laborationen som utförs i kursen ”Introduktion till Linux och små nätverk” (DVG001). Laborationen består av fyra delar, samtliga delar utförs i labbmiljön som tagits fram under den första laborationen.

Nedan följer en lista som sammanfattar uppgifterna för laborationen:

1. Under uppgift ett kontrolleras vilka portar som är öppna på labbmaskinen med hjälp av `netstat` och `nmap`. Ytterligare utförs en kort analys om hur resultaten skiljer sig mellan `netstat` och `nmap` samt vad skillnaderna kan bero på.
2. För uppgift två installeras webbservern `apache2` på labbmaskinen. Därefter ändras innehållet i standardwebbsidan (`index.html`) och slutligen besöks sidan för att bekräfta att innehållet har ändrats.
3. Uppgift tre går ut på att en `NFS-server` och en `NFS-klient` installeras och konfigureras för att göra katalogen `/srv/data` läsbar för hela det lokala nätverket samt skrivbar för endast en maskin.
4. Slutligen handlar uppgift fyra om att ställa in en brandvägg på så vis att alla maskiner kan komma åt `SSH` och `HTTP`, men att `NFS`-tjänsterna bara är tillgängliga över det lokala nätverket. Yttermera ska ska `limit` användas för att begränsa antalet uppkopplingar mot `SSH` från samma adress.

1.2. Syfte

Laborationen syftar till att dels analysera vilka portar som är öppna på en maskin och dels utforska hur en brandvägg kan användas för begränsa tillgång till olika servertjänster/portar.

2. Planering och genomförande

Genom hela laborationen sker kontinuerligt dokumentation. För de praktiska momenten nyttjas en SSH-anslutning och vid eventuella problem används föreläsningsmaterialet och en sökmotor. Genomförandet är uppdelat i fyra olika delar för att strukturera de olika deluppgifterna.

2.1. Planering

Arbetet dokumenteras kontinuerligt i realtid i samband med att laborationen utförs för att säkerställa så ackurat information i rapporten som möjligt. Vid eventuella problem eller funderingar nyttjas i förstahand `man` eller föreläsningsmaterialet och i andra hand en sökmotor såsom DuckDuckGo eller Google. Där officiell dokumentation för distrubitionen finns tillgänglig på internet prioriteras denna högst. Kommunikation upprättas mot labbmiljön som kör Debian 12 Bookworm på en gamal Dell Inspiron 570 genom en `SSH`-anslutning från min Macbook Pro med Sequoia 15.5. För anslutning till `NFS` används en PC som använder Ubuntu 24.04.

2.2. Genomförande

Genomförandet är uppdelat i fyra olika delar som speglar de olika delarna av uppgiftsbeskrivningen. Nedan följer även en redogörelse av det som utförts innan samtliga delar.

Först upprättades en anslutning till labbmiljön genom `ssh hig-25sipe01@192.168.1.250` i terminalen på min Macbook. Därefter skrevs lösenordet för kontot in. Väl ansluten användes `sudo apt update` och `sudo apt upgrade` för att installera de senaste säkerhetsuppdateringarna och förbereda inför installation av nya paket i enlighet med kursboken [1, pp 116, 120].

2.2.1. Del ett – Kontroll av tjänster/portar

Först installerades paketet `net-tools` genom `sudo apt install net-tools`. Därefter användes `netstat --verbose --tcp --numeric-ports` och senare `netstat --verbose --tcp --numeric-ports`, se Figur 1. Efter att ha utför portskanningen i netstat installerades nmap på Macbooken för att därefter utföra en portskanning på labbmiljön från Macbooken, för utdata se Figur 2.

```
[hig-25sipe01@dvg001:~$ netstat --verbose --tcp --numeric-ports
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0    216 dvg001.localdomain:22   192.168.5.2:51514       ESTABLISHED
[hig-25sipe01@dvg001:~$ netstat --verbose --udp --numeric-ports
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
```

Figur 1: Portskanning för såväl TCP som UDP genom netstat. Med hjälp av växeln –numeric-ports visas även portnummer.

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-14 12:54 CEST
Nmap scan report for dvg001 (192.168.1.250)
Host is up (0.031s latency).
rDNS record for 192.168.1.250: dvg001.localdomain
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE    SERVICE
22/tcp   open     ssh
1433/tcp filtered ms-sql-s
1521/tcp filtered oracle
3306/tcp filtered mysql
5432/tcp filtered postgresql

Nmap done: 1 IP address (1 host up) scanned in 10.89 seconds
```

Figur 2: Portskanning med hjälp av nmap, visar dels SSH-tjänsten men även en del olika databas-servrar.

Hur kommer det sig att netstat och nmap visar olika? Tjänsterna `ms-sql-s`, `oracle`, `mysql` och `postgresql` visas i tillståndet `filtered`. `Filtered` innebär att `nmap` inte kan avgöra om portarna är öppna eller inte, detta beror på att paketen filtreras bort innan de når porten [2]. Utifrån vad jag kunnat observera så misstänker jag att routerns brandvägg kastar bort paketen, detta då databastjänsterna inte syns i nmap när jag deaktiverar ”Intrusion Prevention” på min Unifi Dream Machine Pro som agerar router på nätverket, se Figur 3. När jag sedan aktiverar ”Intrusion Prevention” igen så blir utdatan identisk till Figur 2.

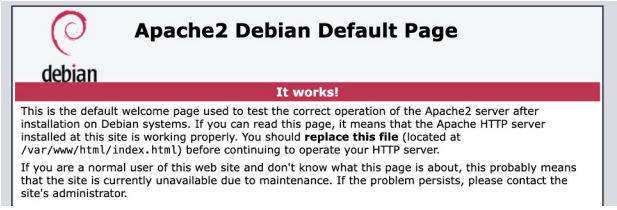
```
Nmap done: 1 IP address (1 host up) scanned in 10.89 seconds
❯ nmap dvg001 -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-14 14:27 CEST
Nmap scan report for dvg001 (192.168.1.250)
Host is up (0.035s latency).
rDNS record for 192.168.1.250: dvg001.localdomain
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 8.34 seconds
```

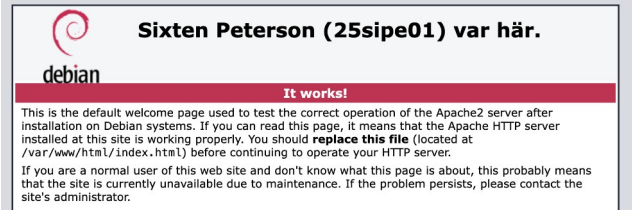
Figur 3: Utdata från nmap efter att "Intrusion Prevention" deaktiverats i inställningarna för routern.

2.2.2. Del två – Installation & konfiguration av Apache2

För att installera `apache2` kördes `sudo apt install apache2`. Därefter besöktes webbsidan genom att i en webbläsare Macbooken besöka adressen `http://dvg001/`, en beskuren bild visar hur webbsidan såg ut innan redigering – se Figur 5. Därefter redigerades span-elementet på webbsidan genom nano, för detta användes kommandot `sudo nano /var/www/html/index.html`. För resultat se Figur 4



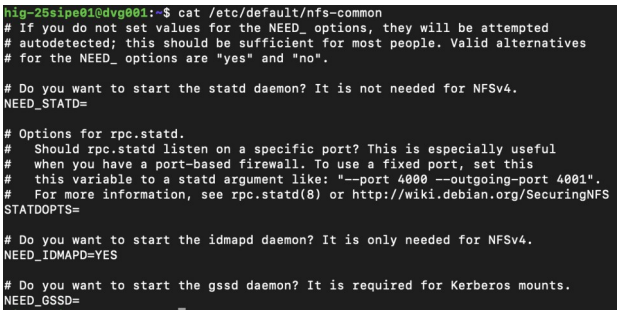
Figur 5: Beskuren bild av standardwebbsidan som ingick vid installation av webbservern.



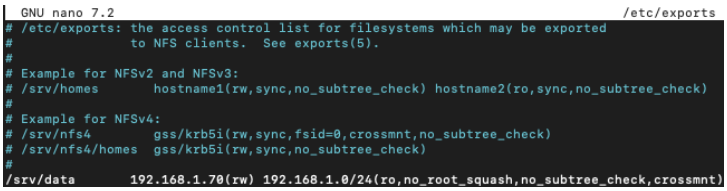
Figur 4: Beskuren bild av standardwebbsidan efter redigering.

2.2.3. Del tre – Installation & konfiguration av NFS

Då jag har en annan maskin på nätverket som kör Ubuntu valde jag att börja med att installera NFS-servern på labbmiljön för att därefter installera klienten på min andra maskin. För att komma igång med installationen användes instruktionerna i uppgfitsbeskrivningen, således kördes `sudo apt install nfs-kernel-server nfs-common rpcbind` för att installera paketen som krävs. Därefter justerades `NEED_IDMAPD` i `/etc/default/nfs-common` för att säkerställa översättning mellan användarnamn i enlighet med uppgiftsbeskrivningen, se Figur 6. När detta var gjort skapades en ny katalog (`/data`) i `/srv`-katalogen med hjälp av `sudo mkdir /srv/data`. Därpå gjordes den nya mappen tillgänglig över NFS, se Figur 7 - sedan startades NFS-servertjänsten om genom `sudo service nfs-kernel-sever restart`. Slutligen kördes `sudo showmount -e` för att bekräfta att katalogen delats ut, vilket den hade – se Figur 8.



Figur 6: Filen /e/d/nfs-common efter justering.



Figur 7: Innehållet i konfigurationsfilen /etc/exports.

Figur 8: Lista med utdelade kataloger från labbmaskinen.

Efter att NFS-servern konfigurerats blev det dags att testa ansluta till den från ubuntumaskinen, således kör kommande kommandon på just den maskinen. Först installerades `nfs-common` och `rpcbind` på maskinen genom `sudo apt install nfs-common rpcbind`. Därefter skapades katalogen `dvg001` i min användares hemkatalog genom `mkdir /home/snicon/dvg001`. Därefter kördes `sudo mount 192.168.1.250:/srv/data /home/snicon/dvg001`. För att sedan testa att det går att skriva från maskinen kördes `echo ”Free as in free speech not as in free beer” > /home/snicon/dvg001/FOSS.txt`. Då upptäcktes att skrivrättigheterna inte stämda eftersom filen misslyckades att skriva på grund av bristande rättigheter.

Kommande kommandon skrivs från denna punkt i labbmiljön genom SSH. Efter att ha kört `ls -ld /srv/data/` gick det att bekräfta att jag missat ställa in rätt ägare och rättigheter. Först skapades en ny användare – `snicon` – med hjälp av `sudo adduser snicon`, och därefter sattes ägaren till `snicon` genom `sudo chown snicon:snicon /srv/data`. För enkelhetens skull justerades rättigheterna på så vis att ägaren och gruppen (`snicon`) får alla rättigheter medan övriga får enbart läsrättigheter. Detta uppnåddes med hjälp av `sudo chmod g+x` /srv/data och `sudo chmod o-x /srv/data`, resultatet går att se i Figur 9.

Figur 9: Rättigheter för /srv/data/.

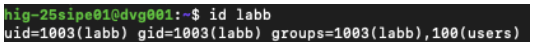
Eftersom jag fortfarande inte fick skrivrättigheter efter att rättigheterna och ägarskapet förändras blev det en hel del felsökande. Först och främst insåg jag att användningen av `sudo` vid `mount` gjorde att `root` blev användaren istället för `snicon`. Således gjordes ett försök att köra `mount 192.168.1.250:/srv/data /home/snicon/dvg001` utan `sudo`, vilket resulterade i följande meddelande: ”mount.nfs: failed to apply fstab options”. I försök att göra det möjligt att montera utdelningen utan att använda `sudo` redigerades `/etc/fstab`, se Figur 10. Därefter kördes `systemctl daemon-reload` för att säkerställa att den nya konfigurationen används. När detta var gjort kördes `mount /home/snicon/dvg001`, vilket monterade utdelningen.

Figur 10: Innehåll i /etc/fstab på ubuntumaksinen (klienten).

Trots denna framgång stämde inte rättigheterna, det var i detta läge som `ls -ld /home/snicon/dvg001` kördes. Ägarskapet stämde inte överens och ett uid och gid visas som ej matchar med användaren på klienten. Detta indikerar att översättningen mellan användarnamn inte fungerade som den skulle, trots att jag tidigare aktiverat NEED_IMAPD enligt uppgiftsbeskrivningen [3, pp 6]. Då jag ställt in att använda NFSv4 i `/etc/fstab` vilket bör säkerställa att översättningen används utifrån min förståelse av uppgiftsbeskrvingingen blev jag ganska konfunderad.

Med utgångspunkten att kunna lösa uppgiften istället för att lägga mer tid på att felsöka översättningen av användarnamn valde jag att istället skapa en ny användare med matchande `uid` och `gid` på såväl server som klient i hopp om att kunna få skrivrättigheter på min klient. Då jag visste att min klient (ubuntumaskinen) bara har en användare sedan tidigare och att servern (labbmiljön) har flera valde jag att skapa en ny användare på servern först. Sedan kontrollerades vilket uid och gid som tilldelats och därefter skapa en ny användare på klienten med samma värden på uid och gid.

Först kördes alltså `sudo adduser labb` (server), därefter användes `id labb` (server), se Figur 11, för att kontrollera vilket uid och gid som används för användaren på servern. När detta var gjort skapades först en ny grupp – labb – med `gid 1003` för att matcha med det som visas i Figur 11, detta genom `sudo groupadd -g 1003 labb` (klient). Efter att gruppen var skapad användes `sudo adduser -u 1003 -g labb labb` (klient) för att skapa användaren med samma uid som det som används på servern. För att slutligen få alla rättigheter att stämma ändrades ägarskapet av `/srv/data` så att labb fick bli den nya ägaren genom `sudo chown labb:labb /srv/data` (server).



Figur 11: Kontroll av uid och gid för användaren labb på servern.

Nu när alla rättigheter stämde växlades användare på klienten från `snicon` till `labb`. Efter detta skapades en ny katalog – `dvg001` i `/home/labb`. Då `/etc/fstab` inte var korrigerad för att visa katalogen för användaren på klienten användes `su snicon` i temrinalen på klienten för att logga in som `snicon` i terminalen vilket möjliggjorde sudorättigheter. Med dessa kördes sudo nano /etc/fstab, och fstabfilen redigerades, se Figur 12. Därefter kördes `mount /home/labb/dvg001`, vilket resulterade i att utdelningen monterades. Efter att ha kört ls -ld /home/labb/dvg001 visades rätt användare och grupp som ägare och det gick bra att skriva till och

läsa från utdelningen. Självklart bekräftades detta genom att köra `echo ”Free as in free speech not as in free beer” > /home/labb/dvg001/FOSS.txt` vilket gick bra, `cat /home/labb/dvg001/FOSS.txt` visade texten, se Figur 13.

```
  GNU nano 7.2
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-jeda5V34na5em8MwpxJNKM9oL3Db3VdwffnIKct10mufL4U5FXRZYasyhfWuufkR / ext4 defaults 0 1
# /boot was on /dev/nvme0n1p2 during curtin installation
/dev/disk/by-uuid/8617392a-d7a5-4d4b-aa87-eb45e76e2de9 /boot ext4 defaults 0 1
# /boot/efi was on /dev/nvme0n1p1 during curtin installation
/dev/disk/by-uuid/A646-F75D /boot/efi vfat defaults 0 1
/swap.img       none    swap    sw      0       0
192.168.1.250:/srv/data /home/labb/dvg001 nfs defaults,noauto,user,nfsvers=4 0 0
```

Figur 12: Redigerad /etc/fstab där hemkatalogen snicon bytts ut till labb.

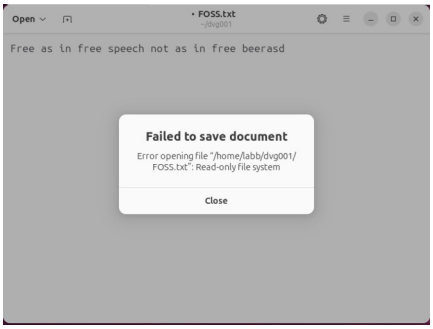
```
labb@Bulbasaur:~/dvg001$ echo "Free as in free speech not as in free beer" > /home/labb/dvg001/FOSS.txt
labb@Bulbasaur:~/dvg001$ cat /home/labb/dvg001/FOSS.txt
Free as in free speech not as in free beer
```

Figur 13: Demonstration av såväl skrivning som läsning till utdelningen.

Slutligen var det dags att säkerställa att skrivrättigheter inte tilldelas andra maskiner än den som specificerats i Figur 7 (`/etc/exports`). För att förenkla processen att testa justerades konfigurationen i /etc/exports enligt Figur 14. Då ip-addressen med skrivrättigheter bytts ut mot en annan än den som min klient tilldelats gick det att testa vad som händer när en annan maskin monterar utdelningen utan att sätta upp monteringen på nytt på ytterligare en annan maskin. Efter att ha kört `sudo exportfs -ra` testades att ansluta till utdelningen och redigera samt spara filen FOSS.txt. Detta resulterade i en varning om att dokumentet ej sparats då filsystemet är begränsat till att bara läsa, se Figur 15. Slutligen ändrades `/etc/exports` tillbaka till hur det var i Figur 7 och `sudo exportfs -ra` kördes.

```
  GNU nano 7.2                                                  /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/srv/data       192.168.1.71(rw) 192.168.1.0/24(ro,no_root_squash,no_subtree_check,crossmnt,fsid=0)
```

Figur 14: Temporärt ändrad /etc/exports för att säkerställa att alla övriga maskiner bara får läsrättigheter.



Figur 15: Misslyckas med att spara ändringar i samband med avsaknad av skrivrättigheter, som förväntat.

2.2.4. Del fyra – Konfiguration av brandvägg

Eftersom denna uppgift består av flera delar bryts uppgiften ner i följande steg:

1. Ge alla maskiner möjlighet att komma åt SSH och HTTP
2. Ge maskiner på det lokala LAN:et möjlighet att komma åt NFS-tjänsterna
3. Ställ in att antalet uppkopplingar mot SSH från samma adress begränsas med limit
4. Ställ in så att maskinen som kör nmap inte kommer åt HTTP

Innan det är möjligt att konfigurera brandväggen måste ufw installeras. För att installera `ufw` kördes `sudo apt install ufw`. För att lösa punkt ett kördes först `sudo ufw allow OpenSSH` och därefter `sudo ufw allow http`. Därefter kördes även `sudo ufw default reject` för att strama åt begränsingarna för anslutningar så mycket som möjligt i syfte att öka säkerheten. När detta var gjort akriverades brandväggen med hjälp av `sudo ufw enable` och slutligen `kördes sudo ufw status` för se över vilka regler som gäller vid denna punkt, se Figur 16. Anslutningen bröts ej men för att säkerställa att allt gått rätt till upprättades en ny anslutning över SSH, vilket gick bra. Ytterligare kunde en portskanning genom `nmap` visa på att enbart `HTTP` och `SSH` var öppna, se Figur 17.

```
hig-25sipe01@dvg001:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
```

Figur 16: Status för brandvägg efter åtgärder för att möta upp till kraven i punkt ett.

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-20 15:10 CEST
Nmap scan report for dvg001 (192.168.1.250)
Host is up (0.00088s latency).
rDNS record for 192.168.1.250: dvg001.localdomain
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Figur 17: Resultat av portskanning efter de nya reglerna för att nå upp till kraven i punkt ett.

Vidare till punkt två, för uppnå detta krav kördes först `sudo ufw allow from 192.168.1.0/24 to any port 2049 proto tcp` för NFS och därefter `sudo ufw allow from 192.168.1.0/24 to any port 111 proto tcp` för `rpcbind`, se Figur 18. Därefter kördes `nmap` från Macbooken genom det lokala nätverket (se Figur 19) och därefter genom en ubuntuserver som befinner sig i ett subnät (se Figur 20). Figurerna redovisar på att reglerna på brandväggen fungerar som önskat i uppgiften. Dessutom har det gått bra att ansluta till utdelningen genom klienten (192.168.1.70).

```
hig-25sipe01@dvg001:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
2049/tcp                   ALLOW       192.168.1.0/24
111/tcp                    ALLOW       192.168.1.0/24
OpenSSH (v6)               ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
```

Figur 18: Status för brandväggen efter nya regler för att uppnå kravet för punkt två.

```
❯ nmap dvg001 -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-20 20:07 CEST
Nmap scan report for dvg001 (192.168.1.250)
Host is up (0.00098s latency).
rDNS record for 192.168.1.250: dvg001.localdomain
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap done: 1 IP address (1 host up) scanned in 42.66 seconds
```

Figur 19: Portskanning av labbmaskinen från Macbook (192.168.1.200), visar samtliga öppna portar.

```
snicon@se-db1:~$ nmap -Pn 192.168.1.250
Starting Nmap 7.80 ( https://nmap.org ) at 2025-05-20 18:10 UTC
Nmap scan report for dvg001.localdomain (192.168.1.250)
Host is up (0.00072s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 4.85 seconds
```

Figur 20: Portskanning från ubuntuserver (192.168.2.20), enbart SSH och HTTP visas som öppna.

Vad gäller punkt tre kördes `sudo ufw limit ssh` för att begränsa, brandväggsstatus går att utläsa i Figur 21. Slutligen kördes `sudo ufw insert 1 deny from`

```
hig-25sipe01@dvg001:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
2049/tcp                   ALLOW       192.168.1.0/24
111/tcp                    ALLOW       192.168.1.0/24
22/tcp                     LIMIT       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
22/tcp (v6)                LIMIT       Anywhere (v6)
```

Figur 21: Status för brandväggen efter ny regel för att uppnå kravet för punkt tre.

`192.168.2.20 to any port 80 proto tcp` för att lägga in regeln före regeln som tillåter trafik från var som helst i syfte att hindra ubuntuservern från att komma åt HTTP på labbmiljön, se Figur 22. När en portskanning utförs från ubuntuservern (192.168.2.20) går det enbart att utläsa att port 22 är öppen, se Figur 23.

```
hig-25sipe01@dvg001:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
2049/tcp                   ALLOW       192.168.1.0/24
111/tcp                    ALLOW       192.168.1.0/24
22/tcp                     LIMIT       Anywhere
80/tcp                     DENY        192.168.2.20
OpenSSH (v6)               ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
22/tcp (v6)                LIMIT       Anywhere (v6)
```

Figur 22: Status för brandväggen efter ny regel för att uppnå kravet för punkt fyra.

```
snicon@se-db1:~$ nmap -Pn 192.168.1.250
Starting Nmap 7.80 ( https://nmap.org ) at 2025-05-20 19:00 UTC
Nmap scan report for dvg001.localdomain (192.168.1.250)
Host is up (0.00068s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 6.55 seconds
```

Figur 23: Portskanning av labbmiljön från ubuntuserver (192.168.2.20), enbart SSH-porten syns som öppen.

3. Beskrivning av slutresultat

Efter utförd laboration har det utförts praktisk övning i portskanning. Ytterligare har tjänsterna Apache2 och NFS installerats och konfigurerats. Dessutom har en ny användare skapats för att ge tillgång till NFS. Slutligen har en brandvägg konfigurerats med regler för att på olika vis begränsa tillgången till portarna på nätverket.

4. Diskussion

Det har varit väldigt intressant att få prova använda nmap i laborationen då detta är något som är helt nytt för mig, programmet kommer tveklöst användas igen i framtiden vid konfiguration av brandvägg för att säkerställa att allt ser rätt ut. Att dessutom få sätta upp sin egna NFS-server var även det intressant, men på fler sätt.

Hemma använder vi Unraid som är ett linuxbaserat operativsystem för att sätta upp ett ”network-attached storage” (NAS), därigenom används Samba och NFS för att dela ut utdelningar. Däremot är denna process bra mycket smidigare och gör grovjobbet åt en, att få en inblick i hur NFS-utdelningar fungerar har därför bidragit till en ökad förståelse till hur det faktiskt fungerar bakom kulisserna - vilket varit mycket givande. Med det sagt, vad jag fick bråka med NFS under laborationen. Det tog en hel del tid att felsöka alltsammans kring de bristande rättigheterna, det känns även tråkigt att inte ha fått igång översättningen mellan användarnamn. Å andra sidan är det skönt att Unraid finns där som ett alternativ att tillförlita sig på för den som är bekväm av sig.

I samband ett grupparbete i en kurs om databaser på Luleå tekniska universitet har jag tidigare satt upp en ubuntubaserad databasserver. Där har jag nyttjat mig av UFW för att begränsa anslutningarna till servern, vilket gjort att denna laboration givit mig en toppenchans att repetera hur UFW används samt lära mig om limitfunktionen.

Sammanfattningsvis har detta varit en fantastiskt givande laboration där nya program/tjänster, felsökning och repetition kombinerats i ett användbart och nyttigt avslut av laborerande i kursen. Skulle jag få chans att göra om laborationen i framtiden utan press om uppkommande tentor och inlämningar hade jag givit mig på att försöka reda ut var jag gått fel i översättningen av användarnamn i NFS.

5. Slutsatser

För att analysera vilka portar som är öppna på en maskin har laborationen visat på att nmap är ett mycket väl fungerande tillvägagångssätt. För att begränsa tillgången till portar har laborationen redovisat hur processen går till genom användning av UFW. Ytterligare har nmap varit ett mycket bra komplement för att bekräfta att konfigurationen av brandväggen gör vad den ska.

6. Referenser

Litteraturförteckning

[1]: R. Hertzog, R. Mas, The Debian Administrator's Handbook, Debian Buster from Discovery to Mastery, 2020.
[2]: Nmap, "Port Scanning Basics," Nmap, [Online]. Available: https://nmap.org/book/man-port-scanning-basics.html. [Accessed: May 21, 2025].
[3]: A. Jackson, DVG001 – introduktion till linux och små nätverk, inlämningsuppgift sex, Assignment instructions, [Online]. Available: https://hig.instructure.com/courses/8261/files/1441478?wrap=1. [Accessed: May 21, 2025]