

Iptables-oppgave

av

Pål Karlsen, Ola Grytting, Kristoffer Svendsen

IKT 208

Tjenesteadministrasjon

Veiledet av

Sigurd Munro Assev

Fakultet for teknologi og realfag

Universitetet i Agder

Grimstad, Oktober 2020

Introduksjon

I denne oppgaven skulle vi bruke iptables til å lage en brannmur. Denne brannmuren skulle stenge alt utenom ping, port 22 (vist en bruker VM), 25 eller 53. I tillegg skal trafikk kun slippes inn om det er en påbegynt sesjon innenfra. iptables skulle også være satt opp til å fungere som en default gateway for våre andre maskiner og ha mulighet for å videresende oss til en av dem om vi prøvde og kommunisere med ip 100.100.100.100. I slutten av oppgaven skulle vi sette opp en NAT som fikk arbeids maskina vår til å se ut som 100.1.1.1 og dei andre labmaskinene som 100.1.1.2.

Contents

1	Iptables	1
2	oppgaver	1
2.1	Stenger all trafikk	1
2.2	Pinging	1
2.3	Porter	2
2.4	Sesjoner	2
2.5	Default gateway	2
3	Konklusjon	3
4	Scriptet	4

List of Figures

1	Pinging	2
2	nmap	3

1 Iptables

Iptables er et brannmur verktøy som tillater system administratoren å konfigurere et IP pakke filter med regler til linux kjerne brannmuren. Filterene er organisert i forskjellige bord som inneholder kjeder av regler for hvordan pakkene skal bli behandlet[1][2]. Reglene blir behandlet i forhold til hvilken rekkefølge dem er organisert i. Det betyr at om du har DROP helt øverst, blir alle reglene under i samme kjede hoppet over, fordi alle pakkene allerede er droppet.

2 oppgaver

Før vi i det hele tatt kunne begynne, måtte vi slå av alle andre brannmurer som var på pc fra før. Dette måtte vi gjøre fordi ellers kunne det oppstå konflikt mellom iptables og brannmuren.

2.1 Stenger all trafikk

Det eneste vi trengte å gjøre her, var å endre INPUT/OUTPUT fra ACCEPT til DROP. Den letteste metoden for å finne ut om dette virker er å åpne ei nettside. Dersom det fungerer så skal du ikke ha noe kontakt med omverdenen.

2.2 Pinging

For å tillate ping ut og motta ping måtte vi først finne ut av hvilken protocol pinging bruker. Etter litt leting fant vi ut at det var icmp. Alt vi da måtte gjøre var og sett opp at maskina har lov til å motta echo-request og echo-reply. så vi brukte "-p icmp -icmp-type 8 -m state --state ESTABLISHED,NEW -j ACCEPT" og "-p icmp -icmp-type 0 -m state --state ESTABLISHED,NEW -j ACCEPT" på INPUT. "-icmp-type 0" betyr echo-reply og "-icmp-type 8" betyr echo-request. på OUTPUT brukte vi "-p icmp -icmp-type 8 -m state --state ESTABLISHED -j ACCEPT" og "-p icmp -icmp-type 8 -m state --state ESTABLISHED -j ACCEPT"

(a)

```
admin@MOV511233 MINGW64 ~
$ ping 10.0.0.206

Pinging 10.0.0.206 with 32 bytes of data:
Reply from 10.0.0.206: bytes=32 time<1ms TTL=64
Reply from 10.0.0.206: bytes=32 time=1ms TTL=64
Reply from 10.0.0.206: bytes=32 time<1ms TTL=64
Reply from 10.0.0.206: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.0.206:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

(b) Pinger arbeids maskina

(c)

```
root@ploopyface:/home/ploopyface# ping 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=13.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=13.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=13.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=13.8 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 13.730/13.862/13.984/0.148 ms
```

(d) Pinger google

Figure 1: Pinging

2.3 Porter

Om man har en VM så man åpne port 22 aller først. Dersom man ikke gjør dette, stenger man seg selv ute fra maskinen og den må resettes.

For å åpne portene så brukte vi kommandoen `"-p tcp -dport 25 -m state --state ESTABLISHED,NEW -j ACCEPT"` i INPUT og `"-p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT"` i OUTPUT. En kan bare endre `"-dport"` og `"--sport"` til den porten man vil skal bli åpnet.

Av en eller annen grunn fikk vi ikke lov til å åpne port 53. Vi er ikke helt sikre på hvorfor vi ikke får lov til dette, men vi mistenker at det enten er på grunn av tidligere oppgaver, eller mer sannsynlig: Labnettet på campus som hindrer oss.

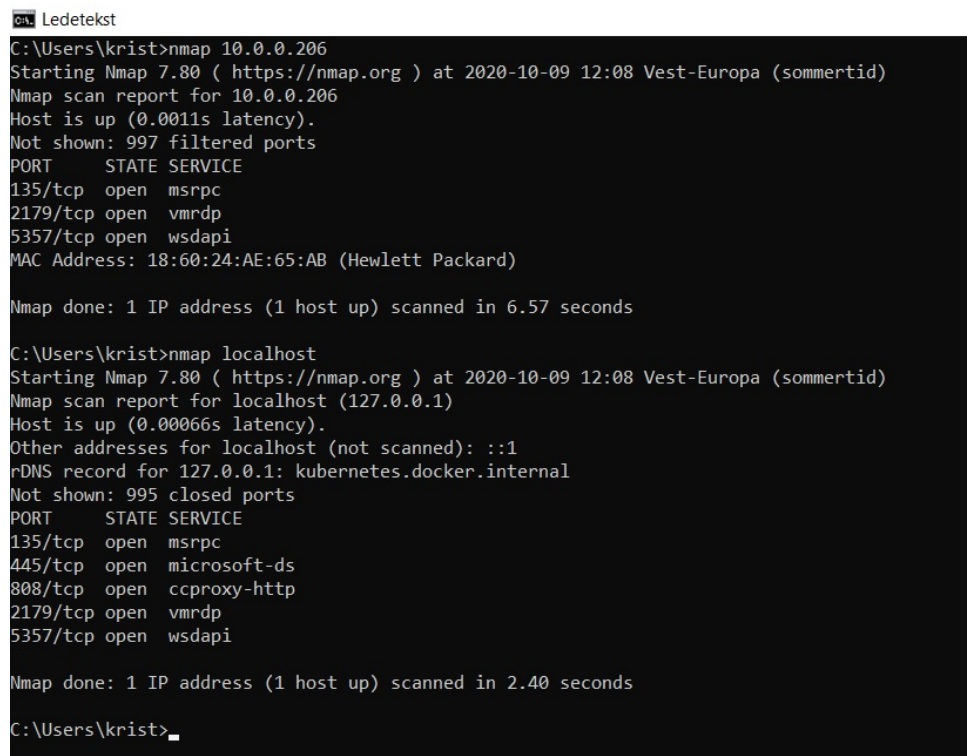
Vi fikk til å åpne port 25 og vi klarte å bruke Postfix (satt opp fra tidligere oppgave) fra en annen maskin.

2.4 Sesjoner

For å kun tillate trafikk i en pågående sesjon i systemmet må en endre på koden litt. Fra `"-m state --state ESTABLISHED,NEW,RELATED"` til `"-m state --state ESTABLISHED,RELATED"`

2.5 Default gateway

Denne delen tok nok mest tid å få til. Her satt vi en god stund å prøvde og finne en måte for å få det til, men vi landet til slutt på `"iptables -t nat -A PREROUTING -d 10.0.0.206 -j DNAT --to-destination 10.0.0.154"`. Det denne gjør er at alt som kommer inn på 10.0.0.206 blir sendt videre til 10.0.0.154. Dette gjorde vi fordi vi ville sjekke om det fungerte.



```
Ledetekst
C:\Users\krist>nmap 10.0.0.206
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-09 12:08 Vest-Europa (sommertid)
Nmap scan report for 10.0.0.206
Host is up (0.0011s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
2179/tcp   open  vmrpd
5357/tcp   open  wsapi
MAC Address: 18:60:24:AE:65:AB (Hewlett Packard)

Nmap done: 1 IP address (1 host up) scanned in 6.57 seconds

C:\Users\krist>nmap localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-09 12:08 Vest-Europa (sommertid)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00066s latency).
Other addresses for localhost (not scanned): ::1
rDNS record for 127.0.0.1: kubernetes.docker.internal
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
445/tcp    open  microsoft-ds
808/tcp    open  ccproxy-http
2179/tcp   open  vmrpd
5357/tcp   open  wsapi

Nmap done: 1 IP address (1 host up) scanned in 2.40 seconds

C:\Users\krist>
```

Figure 2: nmap

som en ser på bildet så nmaper vi 10.0.0.206 som er arbeidsmaskina. Vi så at microsoft-ds port 445 var åpen, og siden vi bruker linux maskin så vet vi at dette ikke er ip 10.0.0.206, men 10.0.0.154.

Vi klarte ikke å få til at 2 maskiner snakker med hverandre gjennom gatewayen. Vi prøvde veldig masse, men ingenting fungerte å vi vet egentlig ikke helt hvordan vi skal gjøre det videre.

3 Konklusjon

Denne oppgaven var ganske utfordrende. Det ble mye egen læring, å lese sei opp på hva som gjorde hva, og mye prøving og mislykking. For å så hva som fungerte og hva som ikke fungerte. Mange av kommandoene til iptables vi fant på nettet, fungerte ikke. Eller: Det kan være de fungerte, men vår fysiske maskin viste hvertfall ikke noe forandring på labnettet.

Hvorfor fikk vi ikke åpnet port 53? Det kan ha med at skolen ikke tillatter oss, men vi vet egentlig ikke helt. Vi er egentlig ikke helt sikre på at sesjon virket. vi viste ikke hvordan vi testet dette, men vi klarte værtfall å bruke postfix på arbeidsmaskina gjennom en av våre andre maskiner.

Hvordan vi setter opp DNAT og SNAT var ganske vrient å få til. Vi klarte å forwarde alt som kom på arbeidsmaskinen til en annen maskin, men vi klarte ikke å gjøre sånn at når andre maskiner vil ha kontakt med 100.100.100.100 så får de kontakt med enn av de andre lab maskinenene. hvordan dette skulle gjøres fant vi ikke ut av. Vi prøvde masse forskjellig, men vi så ingen forskjell og vi fikk det aldri helt til. Kanskje vist vi hadde lest oss litt mer opp hadde vi fått det til, men det er et stort kanskje. Det hadde kanskje også vært lettere om vi hadde brukt de virtuelle maskinene fra starten av, slik oppgaven var tiltenkt. Dette kommer vi til å gjøre i neste oppgave, slik at vi enklere kan følge oppgavebeskrivelsen.

4 Scriptet

```
#!/usr/bin/env bash

# ip addressene som vi skal bruke

# tillater routing

ip route add 100.100.100.100 via 10.0.0.206 dev eno1

# sletter alle tidligere regler for så å lage nye
iptables -F
iptables --table nat --flush

# regler for input
# åpner porter
iptables -A INPUT -p tcp --dport 22 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp --dport 25 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

# åpner for motta og sende pings
iptables -A INPUT -p icmp --icmp-type 8 -m state --state ESTABLISHED,NEW,RELATED -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 0 -m state --state ESTABLISHED,NEW,RELATED -j ACCEPT
iptables -A INPUT -p tcp --tcp-flags ALL SYN -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
# dropper alt som ikke har blitt spesifisert over
iptables -A INPUT -j DROP

# regler for output
# åpner porter
iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 53 -m state --state ESTABLISHED -j ACCEPT

# åpner for motta og sending av ping
iptables -A OUTPUT -p icmp --icmp-type 0 -m state --state ESTABLISHED,NEW,RELATED -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 8 -m state --state ESTABLISHED,NEW,RELATED -j ACCEPT
iptables -A OUTPUT -p tcp --tcp-flags ALL SYN -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# dropper alt som ikke har blitt spesifisert over
iptables -A OUTPUT -j DROP

# forward regler
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -s 10.0.0.0/8 -j ACCEPT
iptables -A FORWARD -j ACCEPT

# nat regler
iptables -t nat -A PREROUTING -d 10.0.0.206 -j DNAT --to-destination 10.0.0.154
iptables -t nat -A POSTROUTING -s 10.0.0.154 -j SNAT --to-source 10.0.0.206
iptables -A POSTROUTING -t nat -j MASQUERADE
```

References

- [1] KORBIN BROWN. *The Beginner's Guide to iptables, the Linux Firewall*. AUGUST 27, 2020. URL: <https://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall/>.
- [2] Justin Ellingwood. *How the iptables Firewall Works*. Mai 02, 2014. URL: <https://www.digitalocean.com/community/tutorials/how-the-iptables-firewall-works>.