Øvelse 23 – Tillad ping udefra og udvalgte ICMP-beskeder

Information

Formålet med denne øvelse er at vise, hvordan firewallen kan arbejde med specifikke protokoller – her ICMP – og hvordan man kan træffe et bevidst valg om, hvorvidt en server skal kunne pinges udefra.

ICMP bruges bl.a. til: - Ping (echo-request / echo-reply) - Fejl- og statusbeskeder (fx destination unreachable, time exceeded) - Fejlfinding og netværksdiagnostik

ICMP kan også bruges i rekognosceringsfasen, hvor angribere scanner netværk. At blokere ICMP kan gøre rekognoscering sværere, men det er ikke i sig selv en reel beskyttelse mod angreb.

I denne øvelse udvider du firewall-konfigurationen fra øvelse 21, så:

- Serveren kan pinges udefra
- Udvalgte ICMP-fejlbeskeder tillades
- Alt andet fortsat blokeres af den afsluttende DROP-regel

Vigtigt:

Regler i iptables evalueres oppefra og ned. Hvis en pakke matches af en tidligere regel, stoppes evalueringen, og resten af kæden ignoreres. Sørg derfor for, at ICMP-reglerne placeres før den afsluttende DROP-regel.

Instruktioner

1 Tillad ping udefra (ICMP echo-request)

Indsæt en regel, der tillader ICMP echo-request (type 8) før DROP-reglen:

```
sudo iptables -I INPUT 3 -p icmp --icmp-type echo-request -j ACCEPT
```

> Hvorfor -I INPUT 3?
>
> -I indsætter reglen på en bestemt placering i kæden. Ved at indsætte på linje 3 placeres ICMP-reglen efter: 1. Loopback-reglen 2. Reglen for ESTABLISHED,RELATED men før den afsluttende DROP-regel.

Bemærk:
Når du indsætter flere regler på samme position (“3”), vil den senest indsatte komme øverst blandt dem. Det er ok her, så længe alle ICMP-regler ligger før DROP.

Echo-reply (type 0) vil automatisk blive tilladt via den eksisterende ESTABLISHED,RELATED-regel fra øvelse 21.

2️ Tillad ICMP type 3 – Destination Unreachable

```
sudo iptables -I INPUT 3 -p icmp --icmp-type destination-unreachable -m conntrack --ctstate RELATED -j ACCEPT
```

3️ Tillad ICMP type 11 – Time Exceeded

```
sudo iptables -I INPUT 3 -p icmp --icmp-type time-exceeded -m conntrack --ctstate RELATED -j ACCEPT
```

4️ Tillad ICMP type 12 – Parameter Problem

```
sudo iptables -I INPUT 3 -p icmp --icmp-type parameter-problem -m conntrack --ctstate RELATED -j ACCEPT
```

5️ Gennemse regelkæden

Udskriv regelkæden og verificér rækkefølgen:

```
sudo iptables -L -v -n
```

Kontrollér:

- Ligger ICMP-reglerne før DROP?
- Matcher rækkefølgen jeres design?

6️ Test ping udefra

Fra din Kali (eller anden host på samme netværk):

```
ping <serverens-ip>
```

Overvej:

- Kommer der svar?
- Hvad sker der, hvis du sletter echo-request-reglen igen?
- Hvad betyder det for serverens synlighed på netværket?

---

7 Undersøg ICMP-typerne

Undersøg og forklar kort betydningen af:

- Type 3: Destination Unreachable
- Type 11: Time Exceeded
- Type 12: Parameter Problem

Hvorfor kan det være uhensigtsmæssigt at blokere disse?

Refleksion

Overvej følgende:

1. Hvad betyder det sikkerhedsmæssigt, at en server svarer på ping?
2. Er blokering af ping reel sikkerhed eller “security by obscurity”?
3. Ville du tillade ping på:
    - En intern server?
    - En offentlig webserver?
    - En server i en DMZ?
    - Hvilke konsekvenser kan en forkert placering af regler have

Perspektivering – iptables vs. UFW

I denne øvelsesrække har vi arbejdet direkte med `iptables`.
Dette er et lavniveau-værktøj (low level), der giver fuld kontrol over Netfilter i Linux-kernen.

Fordelen ved at arbejde med iptables er, at du får:

- Forståelse for regelkæder (INPUT, OUTPUT, FORWARD)
- Indsigt i rækkefølge og matchning
- Forståelse for stateful firewall (`conntrack`)
- Fuld kontrol over detaljer

Ulempen er, at konfigurationen:

- Kan være kompleks
- Er let at lave fejl i
- Kan være svær at vedligeholde i større miljøer

I praksis anvendes ofte et mere brugervenligt værktøj oven på iptables – fx **Uncomplicated Firewall (UFW)**.

UFW fungerer som et abstraherende lag oven på iptables og gør det nemmere at:

- Åbne og lukke porte
- Arbejde med simple regler
- Administrere firewall uden at håndtere regelrækkefølge manuelt

Eksempel:

```
sudo ufw allow 22/tcp
sudo ufw enable
```

Under overfladen genererer UFW stadig iptables-regler.

Du kan læse mere om UFW her:
https://documentation.ubuntu.com/server/how-to/security/firewalls/

Faglig pointe

Formålet med at arbejde med iptables i dette forløb har været:

- At forstå mekanismen bag en host-baseret firewall
- At kunne analysere og designe en regelstruktur
- At forstå konsekvenserne af regelrækkefølge og state tracking

Når man først forstår mekanismen, kan man trygt anvende værktøjer som UFW.

Last update: 2026-03-20 13:58:28