Øvelse 21 – Tillad indgående trafik fra etablerede forbindelser

Information

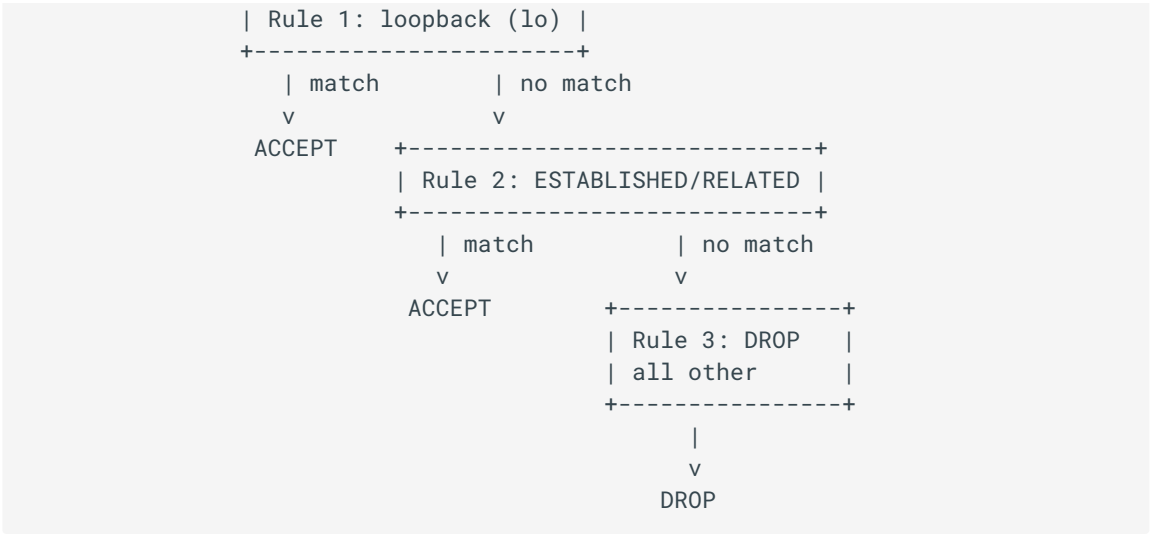
Formålet med denne øvelse, er at vise hvordan en *stateful* firewall kan benyttes til at sikre svar på udadgående kommunikation.

Iptables er en *"stateful"* firewall. Det betyder, at den blandt andet kan holde styr på, hvilke forbindelser operativsystemet har oprettet til andre enheder på netværket. Dette gør det muligt for iptables at tillade svar på forbindelser, som din maskine selv har initieret, samtidig med at den blokerer al uautoriseret indgående trafik. I denne øvelse skal vi arbejde med netop dette.

I forrige øvelse blev alt indadgående trafik blokeret. Hvilket kan være en smule uhensigtmæssigt. Det bloker nemlig for alt trafik, heriblandt og for trafik på *loopback interfacet*, og det forhindre svar på de udadgående forbindelser vi forsøger at etablere. Så i denne øvelse, skal vi tillade den trafik vi ønsker.

> **Hvordan fungerer regelkæden i denne øvelse?**
>
> I iptables behandles netværkstrafik ved, at pakker sendes gennem en regelkæde (*chain*), hvor reglerne evalueres **oppefra og ned**.
> Når en pakke matcher en regel, udføres handlingen (fx `ACCEPT` eller `DROP`), og resten af kæden ignoreres.
>
> I denne øvelse opbygges INPUT-kæden gradvist efter følgende princip:
>
> 1. Tillad nødvendig systemtrafik (loopback)
> 2. Tillad svar på forbindelser, som maskinen selv har oprettet (`ESTABLISHED,RELATED`)
> 3. Bloker al øvrig indgående trafik
>
> Denne struktur implementerer en *allow list*-tilgang, hvor kun eksplicit tilladt trafik accepteres, mens alt andet afvises.

Input kæde evalueres ovenfra og ned:



Instruktioner

Du skal lave en større konfiguration af firewallen, så den kan være en god ide, at udskrive regelkæden efter hver ændring, for at valider rækkefølgen.

1 Nulstil firewall-regler

1. Eksekver kommandoen: `sudo iptables -F`

> Flusher (rydder) alle eksisterende iptables-regler, så vi starter fra en ren konfiguration.

---

2 Tillad loopback-trafik

1. Eksekver kommandoen: `sudo iptables -A INPUT -i lo -j ACCEPT`

   > Tillader trafik på loopback-interface (`lo`), hvilket sikrer, at systemet kan kommunikere med sig selv.

2. Eksekver kommandoen: `sudo iptables -A OUTPUT -o lo -j ACCEPT`

   > Tillader udgående trafik på loopback-interface, så interne processer fungerer korrekt.

> **Verificering af loopback-regler**
>
> Efter at have tilføjet reglerne for loopback-trafik, kan du kontrollere, at de er blevet oprettet korrekt ved at vise firewallens regelkæder:
>
> ```
> sudo iptables -L -v -n
> ```
>
> Denne kommando viser iptables-reglerne i et mere detaljeret format:

> - `-L` → viser regelkæderne
> - `-v` → viser ekstra information, herunder interfaces og pakkecounters
> - `-n` → viser adresser og porte numerisk (hurtigere og mere præcist)
>
> I outputtet kan du se, at loopback-reglerne er aktive ved at kigge i kolonnerne **in** og **out**:
>
> - `in lo` → trafik der kommer ind via loopback-interfacet
> - `out lo` → trafik der sendes ud via loopback-interfacet
>
> Dette bekræfter, at firewall-reglerne kun gælder for lokal systemkommunikation og ikke for eksterne netværksinterfaces. Du bør løbende bekræfte regel kæden, efter at havde tilføjet en ny regel.

3️ Tillad nødvendig udgående trafik

1. Eksekver kommandoen: `sudo iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT`
   > Tillader udgående HTTP-trafik (port 80), som bruges til at hente websider via `curl` eller en browser.
2. Eksekver kommandoen: `sudo iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT`
   > Tillader udgående HTTPS-trafik (port 443), som er nødvendig for sikre webforbindelser.
3. Eksekver kommandoen: `sudo iptables -A OUTPUT -p udp --dport 53 -j ACCEPT`
   > Tillader udgående DNS-opslag via UDP (port 53), hvilket er nødvendigt for at oversætte domænenavne til IP-adresser.
4. Eksekver kommandoen: `sudo iptables -A OUTPUT -p  tcp --dport 53 -j ACCEPT`
   > Tillader udgående DNS-opslag via TCP (port 53), som bruges til større DNS-forespørgsler.

4️ Tillad svar på etablerede forbindelser

1. Eksekver kommandoen: `sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`
   > Tillader indkommende pakker, der er en del af eksisterende eller relaterede forbindelser (så svar på forespørgsler kommer tilbage).
2. Eksekver kommandoen: `sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`
   > Tillader udgående pakker, der hører til allerede etablerede forbindelser.

---

5 Bloker al øvrig indgående trafik

1. Eksekver kommandoen: `sudo iptables -A INPUT -j DROP`

   > Blokerer al anden indgående trafik, hvilket øger sikkerheden ved at afvise uønskede forbindelser.

Med `-m conntrack` (conntrack = connection tracking), ved iptables at der skal holdes styr på, hvilke forbindelser der er etableret, og hvem der har initieret dem.
Med `--ctstate ESTABLISHED,RELATED -j ACCEPT`, tillades indadgående pakker fra servere, hvor klienten (OS) selv har oprettet forbindelsen.

Efter at have tilføjet reglen, vil iptables tillade indgående trafik fra eksterne servere, der er svar på forbindelser, din maskine selv har oprettet. Dette betyder, at du kan få svar på webanmodninger, SSH-forbindelser eller andre tjenester, som du selv har startet, mens indgående forbindelser udefra, som du ikke selv har initieret, vil blive blokeret.

I øvelsen har i indtil videre arbejdet udfra *allow list*-tilgangen, hvor der specificeret konkret, hvornår der må etableres netværksforbindelser til og fra hosten.

> **Bemærk**
>
> Hvis du udfører denne konfiguration via en eksisterende SSH-forbindelse, vil forbindelsen typisk **ikke blive afbrudt**.
> Dette skyldes, at SSH-sessionen allerede er oprettet og derfor klassificeres som `ESTABLISHED` af iptables connection tracking.
>
> Da reglen `--ctstate ESTABLISHED,RELATED -j ACCEPT` placeres før `DROP`-reglen, vil trafikken til den aktive SSH-session fortsat blive accepteret.
> Nye indgående SSH-forbindelser vil derimod blive afvist, medmindre port 22 eksplicit tillades tidligere i regelkæden.

Test med curl

Efter du har oprettet reglen, kan du teste dens funktionalitet ved at prøve at etablere en forbindelse til en tjeneste, såsom en webserver.

Brug et værktøj som Curl til at sende en anmodning til en ekstern server (f.eks. www.google.com) og se, om svaret kommer igennem, mens en uautoriseret indgående forbindelse (f.eks. en ping) bliver afvist.

Du kan læse mere om iptables i forberedelsen til i dag, eller på iptables manual-siden.

I næste øvelse bygges der videre på denne øvelse.

Links

- iptables man page
- Curl

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