Indholdsfortegnelse

Indledning og motivation

Den første type af firewall der blev udviklet var “simple” pakke filtre. I denne første generation kigges der udelukkende på headeren i de enkelte pakker for at træffe en beslutning, om hvorvidt pakken skal accepteres eller forkastes. Dette foregår ved at hver pakke holdes op imod et regelsæt, der angiver hvilke parametre der skal være opfyldt for at pakken bliver tilladt eller blokeret. For at afgøre dette kan der kigges på parametre såsom afsender/modtager IP adresse, type og portnummer for TCP/UDP.
Simple pakke filtre kigger således på den enkelte pakke, og ikke den kontekst, som pakken optræder i. Hvis man for eksempel åbner for port 80 til en webserver bag firewallen, er det også nødvendigt at tillade forbindelser fra port 80 til omverden, fordi det ønskes at folk udefra skal have svar tilbage. Denne nødvendighed kan resultere i lange og mindre læsbare regelsæt, og ekstra arbejde for administratoren der også skal tillade de udgående svarpakker, for at brugere kan benytte tjenesten.

Problemet er ikke så stort for servere da de oftest opererer på faste porte. Problemet er større med en klient bag en firewall, som åbner forbindelse til en maskine udenfor, evt. på en veldefineret port, men selv modtager svaret på en tilfældig port i intervallet 1024-65535 jfr. konvention[1]. Med TCP er det imidlertid muligt at undgå dette problem ved kun at tillade indgående pakker, hvor SYN bitten ikke er sat og dermed blokere for at oprette nye indgående forbindelser.

Tilsvarende er dog ikke muligt for UDP, som ikke har en SYN-bit, idet protokollen ikke er forbindelsesorienteret. Hvis man vil være sikker på at få svar tilbage på UDP pakker, er man i princippet nødt til at åbne samtlige porte i hele intervallet i firewallen for UDP pakker, hvilket ikke er hensigtmæssigt med hensyn til sikkerhed.

Stateful firewalling

For at imødekomme begrænsningerne ved simple pakke filtre, blev konceptet stateful inspection opfundet. Målet er at skabe mere præcise, læsbare og gennemskuelige firewallregler, så det er så let som muligt at reviewe dem og finde eventuelle fejl i implementeringen af sikkerhedspolitikken.

Teknikken går ud på at gemme information om *forbindelser* i en tabel, således at det er muligt at slå op i tabellen og afgøre, om en given pakke f.eks. er en del af en allerede eksisterende forbindelse, eller begyndelsen på en ny. Forbindelserne inddeles i logiske “states”, på en generel måde uanset de forskellige typer af forbindelser (TCP, UDP osv), og disse “states” kan efterfølgende bruges i firewallens regelsæt. Det er således muligt at lave mere generelle regler, der f.eks. ikke begrænser sig til en specifik type forbindelse.
Med stateful firewalling er det således muligt at oprette en enkelt regel, der tillader alle indgående pakker, der tilhører allerede eksisterende forbindelser.

TCP

“State” kan bestemmes for TCP udfra det trevejs handshake (SYN, SYN-ACK, ACK) som skal finde sted, før en TCP forbindelse er oprettet. Sammen med den aktuelle state gemmes source/destination IP adresserne, portene og flag. Det er desuden muligt yderligere at gemme sekvensnumre, dette er op til implementeringen. Der kan argumenteres for, at benyttelse og kontrol af sekvensnumre tilføjer ekstra sikkerhed, da det med endnu større troværdighed er muligt at afgøre, om en pakke tilhører en allerede kendt forbindelse, men dette indebærer dog større overhead.

Det er desuden en sværere implementering, da der skal tages højde for, at TCP pakker kan ankomme i forkert rækkefølge.
Ved oprettelse af en ny forbindelse sendes, udover SYN bitten, et tilfældigt sekvens nummer som danner grundlag for den nye forbindelse. Serveren sender et acknowledge number tilbage, og disse numre bruges til at holde styr på rækkefølgen af pakker samt hvor mange bytes der sendes.[2] Numrene ændrer sig for hver pakke idet de inkrementeres, og firewallen kan udnytte denne information og gemme de aktuelle numre i state tabellen[3].
Når der ankommer en ny pakke kan firewallen slå op i tabellen og se om pakken tilhører en kendt state, og hvis det er tilfældet, er det ikke nødvendigt at evaluere flere regler. Dette gøres ved at se på source/destination IP adressen, portene, flag samt evt. om sekvensnumrene stemmer med dem, der er gemt i state tabellen. Hvis der er en match, tilhører pakken en allerede kendt forbindelse og kan klassificeres sådan.

Når en TCP forbindelse lukkes ned, sender den ene vært en pakke med FIN bitten sat og den anden vært kvitterer med en pakke med ACK bitten sat. Det samme gøres så den anden vej også, så forbindelsen bliver lukket i begge “retninger”. Et alternativ er et trevejs nedlukning svarende til det trevejs handshake der foregår ved oprettelse af en TCP forbindelse, som beskrevet tidligere. Disse nedlukninger kan firewallen også holde øje med og løbende gemme i state tabellen, og når det sidste ACK er set, er forbindelsen lukket og den pågældende record slettes fra state tabellen.

UDP

Alt dette er dog ikke muligt for UDP, idet denne protokol ikke er forbindelsesorienteret og ikke benytter en lignende ordning med handshake. Det det er dog stadig muligt at opretholde en state tabel hvor source/destination IP adressen og portene gemmes. Der bliver som før tilføjet en record i state tabellen så snart firewallen har set en UDP pakke den ene vej, som oftest en forbindelse ud af firewallen fra en klient bag den. Da der ikke er nogle indikationer af om det bare er

en enkeltstående pakke, må firewallen gå ud fra, at der eventuelt kommer svar tilbage (eller der skal sendes flere pakker) så der skal tilføjes en record i state tabellen. Hvis der så kommer en UDP pakke udefra, kan der slås op i state tabellen for at se, om der skulle være et match fra den afsender og på den samme port. I så fald dømmes pakken som værende en del af "forbindelsen", og staten kan ændres, da det er bekræftet, at der nu er en forbindelse mellem de to værter.

Da UDP ikke lukker "forbindelsen" pænt efter sig ligesom TCP, er det nødvendigt at sætte en timeout for den enkelte UDP record i state tabellen, som ellers kunne blive meget stor og i værste tilfælde bruge alt den tilgængelige hukommelse. En ofte brugt løsning er at fjerne en record, hvis der ikke har været nogle aktive pakker der tilhører den "forbindelse" i et antal sekunder, f.eks. 60.

Netfilter / iptables

Som et eksempel på en konkret implementering har Netfilter/iptables på Linux følgende states[4]:

- NEW – Når den første pakke i en forbindelse ses. For en TCP forbindelse vil SYN bitten være sat og forbindelsen er endnu ikke etableret.
- ESTABLISHED – Når der er set trafik i begge retninger på en forbindelse (når der er blevet svaret med SYN/ACK for TCP)[5].
- RELATED – En forbindelse der på en eller anden måde er relateret til en allerede eksisterende forbindelse.
- INVALID – Pakker som ikke kan identificeres til at høre til en kendt state, og ikke hører til i NEW, eksempelvis en ICMP pakke angående en vært, der ikke er forsøgt oprettet forbindelse til.

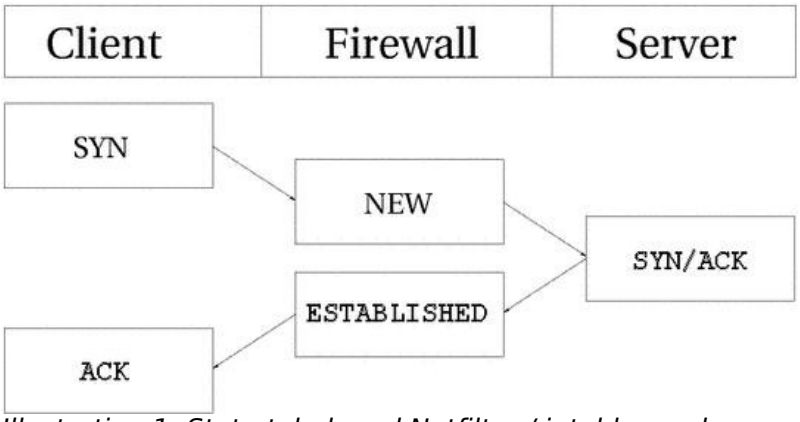

Illustration 1: State tabel med Netfilter / iptables ved oprettelse af TCP forbindelse

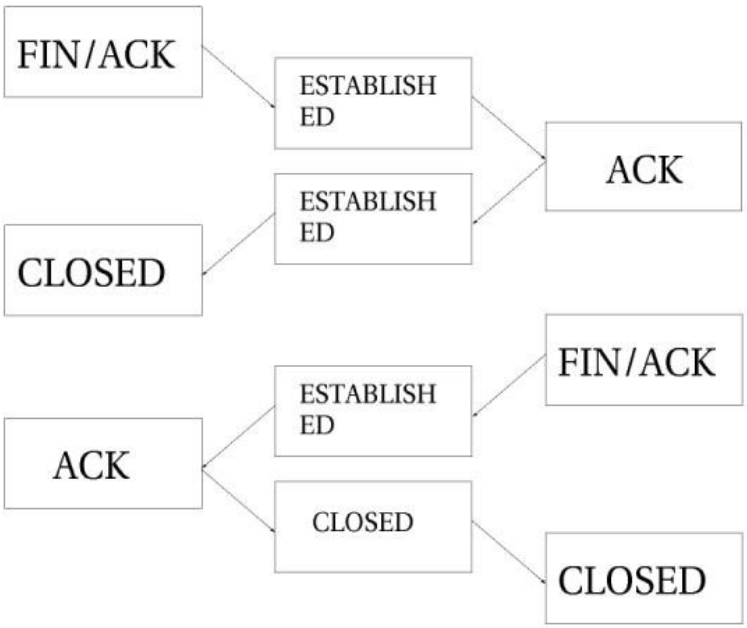

Illustration 2: State tabel med netfilter / iptables ved lukning af TCP forbindelse

Netfilter/iptables benytter ikke sekvensnumre for TCP, men kun det trevejs håndslag.

RELATED staten kan benyttes til på en overskuelig måde at tillade nye forbindelser der udspringer af allerede eksisterende. Eksempelvis er FTP data forbindelser

relateret til FTP kontrol forbindelser: Klienten forbinder på vanlig vis til serveren og skaber kontrol forbindelsen, hvor der kan navigeres rundt i filsystemet, anmodes om filer osv. Når der anmodes om en fil og den skal overføres, eksisterer der to muligheder for data forbindelsen[6]:

1. Serveren opretter (initierer) en forbindelse til brugeren på en separat port.
2. En alternativ ("passive mode") hvor serveren lytter på en midlertidlig separat port, som klienten forbinder til.

Begge muligheder resulterer i en ny forbindelse der bliver oprettet. Hvis firewallen forstår FTP kontrol protokollen kan den "følge med" og oprette states på samme vis som beskrevet tidligere, og placere disse udspringende forbindelser i RELATED staten. Uden stateful inspection ville det være nødvendigt, enten på klient- eller serversiden, at åbne et interval af porte fra arbitrære værter, for at FTP servicen ville fungere. Med stateful inspection accepteres der kun forbindelser fra værter, som rent faktisk via kontrol forbindelsen har fået tildelt den midlertidige port til overførsel af data.

Eksempler

I følgende eksempler antages det, at der benyttes en "afvis som standard" tilgang, hvor alt, der ikke eksplicit er tilladt, er blokeret.

```
iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED -j ACCEPT
```

Eksempel 1: Firewall der giver omverdenen adgang til webserver

I eksempel 1 ses et regelsæt, der tillader adgang til en webserver bag firewallen. Dette gøres ved at tillade indgående pakker tilhørende NEW eller ESTABLISHED states på port 80 (HTTP). Ud af firewallen tillades kun pakker, der tilhører ESTABLISHED staten. Resultatet er at brugere udefra kan anmode om at få vist en hjemmeside, og svaret kan blive sendt ud gennem firewallen. Indgående pakker, der ikke enten er en ny eller hører til en eksisterende forespørgsel, tillades ikke. Det er heller ikke tilladt for webserveren at oprette nye udgående forbindelser eller

svare hosts, som ikke selv har anmodet om det. Dette kan være et ekstra lag af sikkerhed såfremt serveren skulle blive ramt af en virus, worm eller lignende. Hvis det ønskes at åbne for flere services på serveren, er det kun nødvendigt med én regel pr. service for at åbne den indgående forbindelse. Linje 2 i eksemplet, hvor reglen for udgående pakker sættes, tillader alle svarpakker på etablerede forbindelser.

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

Eksempel 2: Simpel "hjemme" firewall

I eksempel 2 ses en “hjemme firewall” der lader brugeren tilgå alle hosts, men kun tillader svar tilbage fra dem, der rent faktisk er blevet efterspurgt. Første linje angiver, at indgående pakker, der tilhører ESTABLISHED eller RELATED states, accepteres. Anden linje angiver, at udgående pakker i samme states samt pakker i NEW tillades – dette gør det muligt at oprette nye forbindelser, f.eks. bede om at få vist en hjemmeside, og første linje vil tillade at svaret kommer ind.

Konklusion

I denne rapport er principperne bag stateful inspection blevet undersøgt. Det er blevet skitseret hvordan de overordnet fungerer, sammenlignet med state-less firewalls, og der er givet eksempler på regelsæt i en konkret implementering. Stateful firewalling gør det muligt at undgå de tidligere nævnte problemer med at tillade “for meget” i firewallen for at få svarpakkerne ind, og det er muligt at skabe enklere regelsæt, der bedre matcher det man ønsker. Dertil kommer mere læsbare og kontrollérbare regelsæt, hvilket er at foretrække.

1 http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml
2 http://en.wikipedia.org/wiki/Transmission_Control_Protocol
3 http://www.informit.com/articles/article.aspx?p=31945&seqNum=3
4 http://www.faqs.org/docs/iptables/userlandstates.html
5 http://www.faqs.org/docs/iptables/tcpconnections.html
6 http://tools.ietf.org/html/rfc959