Bezpieczeństwo systemów informatycznych
ĆWICZENIE Firewall

1. Firewall

1.1 Filtrowanie pakietów IP w systemie Linux (netfilter/iptables)

Filtr pakietów to oprogramowanie, które sprawdza nagłówki pakietów przepływających przez stos protokołów. Decyduje o ich losie, **akceptując** (ang. *accept*) lub **odrzucając** (ang. *drop*) poszczególne pakiety. Taką funkcjonalność posiadają np. routery filtrujące.

W systemie Linux filtrowanie pakietów IP jest wbudowane w jądro systemu operacyjnego. Od wersji 2.3.15 jądra odpowiedzialny jest za to moduł o nazwie netfilter, oferujący niezależną filtrację pakietów przychodzących, wychodzących oraz routowanych. Moduł ten pracuje jako swoisty rdzeń umożliwiający dołączanie różnych modułów konfiguracji i administracji filtracji pakietów, np. modułu iptables. Z kolei narzędzie iptables służy do dynamicznego konstruowania reguł filtracji. Reguły przechowywane są w pamięci jądra, a ich stałą dostępność można zapewnić przechowując wybrane konfiguracje w plikach (wykorzystując skrypty iptables-save i iptables-restore).

Moduł iptables zarządza trzema niezależnymi listami reguł filtracji, nazywanymi **łańcuchami** INPUT (wejściowy), OUTPUT (wyjściowy) i FORWARD (routing). Moduł ten można zlokalizować poleceniem:

```
locate ip_tables.ko
```

Następnie wystarczy załadować go poleceniem:

```
modprobe ip_tables
```

Pierwszą rzeczą jest ustawienie domyślnej polityki obsługi pakietów dla łańcuchów, np.:

```
iptables -P OUTPUT DROP
```

`-P` = Policy

co jest zgodne z regułą domyślnej odmowy dostępu, lub np.:

```
iptables -P OUTPUT ACCEPT
```

co byłoby naruszeniem tej reguły.

Funkcje modułu iptables zobrazujemy na następujących przykładach poleceń:

wyświetlenie reguł filtracji dla łańcucha FORWARD:

```
iptables -L FORWARD
```

`-L` = List rules

zezwolenie na przepuszczanie pakietów protokołu TCP:

```
iptables -A FORWARD -p tcp -j ACCEPT
```

`-A` = Add rule
`-p` = protocol
`-j` = job to do

usunięcie w/w reguły z łańcucha FORWARD:

```
iptables -D FORWARD -p tcp -j ACCEPT
```

`-D` = Delete rule

wstawienie reguły na 1-sze miejsce w łańcuchu FORWARD:

```
iptables -I FORWARD 1 -p tcp -j ACCEPT
```

`-I` = Insert rule

usunięcie reguły nr 3 z łańcucha FORWARD:

```
iptables -D FORWARD 3
```

usunięcie wszystkich reguł łańcucha FORWARD:

```
iptables -F FORWARD
```

`-F` = Flush chain

zezwolenie na nadawanie pakietów protokołu ICMP w pętli zwrotnej:

```
iptables -A OUTPUT -p icmp -s 127.0.0.1 -j ACCEPT
```

`-s` = source address

odrzucanie pakietów protokołu ICMP przychodzących z sieci 199.1.2.0:

```
iptables -A INTPUT -p icmp -s 199.1.2.0/255.255.255.0 -j DROP
```

lub:

```
iptables -A INTPUT -p icmp -s 199.1.2.0/24 -j DROP
```

odrzucanie pakietów innych niż TCP przychodzących na interfejs eth0 skierowanych do innej sieci niż 199.1.2.0:

```
iptables -A INTPUT -i eth0 -p ! tcp -d ! 199.1.2.0/24 -j DROP
```

`-d` = destination address
`-i` = input interface

odrzucanie routowanych pakietów TCP przychodzących na interfejs eth0 skierowanych do innej sieci niż 199.1.2.0:

```
iptables -A FORWARD -i eth0 -p tcp -d ! 199.1.2.0/24 -j DROP
```

odrzucanie wychodzących pakietów TCP z ustawionymi tylko flagami SYN,ACK (przy weryfikacji wszystkich flag nagłówka):

```
iptables -A OUTPUT -p tcp --tcp-flags ALL SYN,ACK -j DROP
```

1.2 NAT i maskarada

Narzędzie iptables umożliwia również konfigurację funkcji translacji (ukrywania) adresów – NAT. Funkcje NAT realizuje w Linuksie moduł jądra o nazwie iptable_nat. Należy rozpocząć od załadowania go w odpowiednim katalogu poleceniem:

```
modprobe iptable_nat
```

Na ogół rozróżnia się translację adresów źródłowych (SNAT) i docelowych (DNAT). Translacja SNAT (maskarada) na routerze pozwala ukryć rzeczywiste adresy sieci wewnętrznej, np. w celu utajnienia ich przed światem zewnętrznym lub przy możliwości stosowania tylko ograniczonego zakresu adresów źródłowych przydzielonych w sieci globalnej. Oto przykłady konfiguracji dla:

pojedynczego adresu wyjściowego:

```
iptables -t nat -A POSTROUTING -o eth0 \
             -j SNAT --to 199.1.1.1
```

predefiniowanej puli portów TCP:

```
iptables -t nat -A POSTROUTING -o eth0 -p tcp \
             -j SNAT --to 199.1.1.1:8000-9000
```

oraz przedziału adresów wyjściowych:

```
iptables -t nat -A POSTROUTING -o eth0 \
             -j SNAT --to 199.1.1.1-199.1.1.99
```

Przy wykorzystaniu DHCP można konfigurację uprościć, np.:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Z kolei translacja DNAT może być użyteczna przy stosowaniu transparentnych usług proxy. Przykład znajduje się w następnym punkcie.

1.3 Transparentne proxy w systemie Linux

Firewall typu „Komputer Twierdza" (BastionHost) przechwytujący komunikację pomiędzy siecią wewnętrzną a zewnętrzną może być zrealizowany za pomocą programu squid, który oferuje transparentne dla użytkowników usługi proxy dla połączeń TCP. W praktyce najczęściej wykorzystuje się funkcję proxy w zestawieniu z WWW cache. Takie zestawienie uzyskuje się przy ustawieniu następujących parametrów konfiguracji (w pliku /etc/squid.conf):

```
http_port 3128     # port na którym nasłuchuje
cache_mem 8 MB     # 8MB cache w pamięci operacyjnej systemu
cache_dir ufs /var/cache/squid 100 16 256  # 100MB cache na dysku
                   # w 16 katalogach z 256 podkatalogami każdy
client_netmask 255.255.255.0 # maska, która filtruje zapis IP klientów
                   #(koniunkcja logiczna) do logów systemowych
cache_effective_user squid   # użytkownik, naktórym ma pracować squid
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Należy jeszcze zdefiniować listę ACL kontroli dostępu (również w pliku /etc/squid.conf), np.:

```
acl localnet src 10.1.1.0/255.255.255.0
http_access allow localnet
http_access deny all
```

Uruchomienie WWW cache sprowadza się do zainicjowania katalogów buforowania squid -z i uruchomienia usługi skryptem RunCache. Aby uzyskać transparentne proxy należy jeszcze przekierować wszystkie pakiety protokołu HTTP do programu squid, np. za pomocą narzędzia iptables:

```
iptables -t nat -A PREROUTING -p tcp --dport 80 \
                        -j REDIRECT --to-port 3128
```

1.4 Rozszerzenia iptables

NetFilter jest zbudowany z modułów, które rozszerzają możliwości zapory firewall. Kilka z nich jest używane domyślnie, na przykład tcp, udp. Poniżej krótko omówiono większość z dostępnych modułów pakietu NetFilter. Wiele z wymienionych modułów dostępna jest w dodatku *patch-o-matic*, który nie zawsze jest całkowicie instalowany.

1.4.1 Krótkie opisy modułów

1. Dotyczące IPSec:
   - ah – dopasowują się do pola "spi" w nagłówku AH pakietu,
   - esp – dopasowują się do pola "spi" w nagłówku ESP pakietu,
   - policy – dopasowują się do "policy" pakietu.
2. Dopasowujące się do nagłówka pakietu IP:
   - dscp – dopasowuje się do 6 bitów DSCP znajdujących się w polu ToS,
   - ecn – sprawdza bit ECN w nagłówku pakietu,
   - ipv4options (*) – sprawdza różne opcje nagłówka pakietu, np. source routing, record route,
   - tcpmss – sprawdza pole MSS (Maximum Segment Size),
   - tos – pole ToS (Type of Service),
   - ttl – pole TTL (Time To Live).
3. Nadzorujące połączenia (stanowość):
   - state – umożliwia zidentyfikowanie istniejących połączeń oraz nowych,
   - conntrack – rozszerza możliwości modułu state,
   - helper – pozwala określić "pomocnika" do przekazywania połączenia, np. standardowym pomocnikiem jest moduł pozwalający przekazywać połączenia ftp-data w trybie aktywnym.

4. Określające typ pakietu:
   - icmp – dopasowuje się do pakietów typu ICMP,
   - tcp – dotyczy pakietów typu TCP,
   - udp – dotyczy pakietów typu UDP,
   - unclean(*) – dotyczy pakietów zniszczonych i niepoprawnych.
5. Określające limity:
   - connlimit(*) – umożliwia określenie maksymalnej liczby połączeń do konkretnego adresu IP z jednego adresu IP klienta,
   - connrate(*) - umożliwia określenie maksymalnego/minimalnego transferu,
   - hashlimit – umożliwia określenie maksymalnego transferu do danej usługi/serwera.
6. Znakowanie pakietów:
   - mark – umożliwia znalezienie oznakowanego pakietu,
   - connmark – umożliwia znalezienie każdego pakietu związanego z oznakowanym połączeniem.
7. Ułatwiające tworzenie reguł:
   - iprange – umożliwia wpisanie zakresu adresów IP,
   - mac – umożliwia sprawdzenie adresu sprzętowego karty sieciowej MAC,
   - multiport – umożliwia wpisanie zakresu portów,
   - owner – dla lokalnych połączeń umożliwia określenie który użytkownik wysłał dany pakiet,
   - pkttype – określa typ pakietu (unicast, multicast, broadcast)
   - set(*) – wykorzystanie stworzonych zbiorów adresów (polecenie ipset),
   - time(*) – określenie daty i/lub czasu,
   - comment – umożliwia dopisanie komentarza do każdej reguły.
8. Rozszerzone sprawdzanie pakietów:
   - length – sprawdzające wielkość pakietu,
   - string(*) – dopasowujące się do zawartości pakietu w polu DATA,
   - ipp2p(*) – dopasowujące się do ruchu generowanego przez aplikacje typu P2P.
9. Określające częstość:
   - limit – określa jak często reguła będzie dopasowana, np. 3 razy na sekundę – 3/s,
   - nth(*) – określa co jaką liczbę pakietów będzie dopasowana reguła, np. co 5 pakiet,
   - random(*) – określa dopasowanie reguły do pakietu z zadanym prawdopodobieństwem, np. 50% pakietów.
10. Monitorujące:
    - recent – tworzy listy adresów wykorzystających łącze (przechodzących przez tą regułę), wyniki zapisywane są w katalogu /proc/net/ipt_recent/NAZWA,
    - account(*) – zlicza ruch do określonych adresów, wyniki w katalogu /proc/net/ipt_account/NAZWA,
    - quota(*) – określa quota'ę dla określonych pakietów.
11. Inne:
    - condition(*) – pozwala warunkować stosowanie reguły, poprzez plik w katalogu /proc/net/ipt_condition/NAZWA
    - osf(*) – odczytuje pasywnie "odcisk palca" (fingerprint) konkretnego adresu IP i zapisuje w pliku /proc/sys/net/ipv4/osf
    - psd(*) – pozwala wykryć skanowanie portów TCP i UDP.

(*) moduły te nie znajdują się w domyślnej instalacji systemu SuSE Linux 10.0 i nie są dostępne w laboratorium ćwiczeniowym.

1.4.2 Krótkie opisy celów (TARGETs)

Cele określają co zostanie zrobione z danym pakietem, domyślnymi celami są akceptacja (ACCEPT) i odrzucenie (DROP) pakietu. Jednak tworcy zapory postanowili rozszerzyć listę dopuszczalnych celów, które również mogą być tworzone przez nas samych dla naszych potrzeb.

Lista celów została podzielona na kategorie:

1. Zmiana adresów IP:
   - BALANCE(*) – podobne do DNAT, ale równomiernie rozkłada obciążenie na wiele adresów IP,
   - DNAT – Destination NAT,
   - MASQUERADE – ukrywanie źródłowych adresów IP,
   - NETMAP – statyczna zamiana adresów całych podsieci,
   - REDIRECT – przekierowanie ruchu na lokalny komputer,
   - SAME – podobne do DNAT/SNAT, ale zawsze przydziela te same adresy IP konkretnym klientom,
   - SNAT – Source NAT.
2. Znakowanie pakietów:
   - CONNMARK – znakuje połączenia,
   - IPMARK(*) – znakowanie pakietów na podstawie adresu IP,
   - MARK – znakuje pakiety.
3. Zmieniające nagłówek pakietu:
   - DSCP – umożliwia zmianę 6 bitów DSCP pola ToS,
   - ECN – umożliwia usunięcie bitu ECN,
   - IPV4OPTSSTRIP(*) – usunięcie wszystkich opcji IP z nagłówka pakietu,
   - TCPMSS – ustawienie pola MSS,
   - TOS – ustawienie pola ToS,
   - TTL – ustawienie pola TTL.
4. Śledzenie pakietów:
   - NOTRACK – wyłącza śledzenie połączenia,
   - TARPIT(*) – przechwytuje połączenia i zawiesza je, w celu zabezpieczenie się przez skanowaniem typu Code Red i Nimda,
   - TRACE(*) – włącza śledzenie połączenia.
5. Logowanie:
   - LOG – logowanie pakietów,
   - ULOG – logowanie w przestrzeni użytkownika,
6. Inne:
   - REJECT – odrzuca pakiety wysyłając określony kod błędu do nadawcy,
   - ROUTE(*) – pozwala nadpisać wpisy tablicy tras,
   - SET(*) – dodaje/usuwa adresy z zakresów (polecenie ipset),
   - XOR(*) – pozwala zastosować proste zabezpieczenie danych pakietu, poprzez wykonanie funkcji XOR na danych i określonym haśle.

1.4.3 Szczegółowe informacje

Wszystkie dodatkowe informacje odnoście każdego z wyżej wymienionych modułów i celów można uzyskać z plików pomocy.

Uzyskanie pomocy o module:

```
iptables -m <modul> -h
```

Uzyskanie pomocy o celu:

```
iptables -j <cel> -h
```

1.4.4 Przykłady

Blokada więcej niż 2 połączeń na port 23

```
iptables -A INPUT -p tcp --syn --dport 23 -m connlimit --connlimit-
above 2 -j REJECT --reject-with icmp-host-unreachable
```

Umożliwienie tylko jednego na minutę połączenia z portem 22

```
iptables -A INPUT -p tcp --dport 22 -m hashlimit --hashlimit 1/min --
hashlimit-mode srcip --hashlimit-name ssh -m state --state NEW -j
ACCEPT
```

Blokada pakietów większych niż 1000 bajtów

```
iptables -A INPUT -m length --length 1000 -j DROP
```

Akceptacja 3 połączeń na minutę

```
iptables -A INPUT -m limit --limit 3/min -j ACCEPT
```

Stworzenie statystyk połączeń z portem 80

```
iptables -A INPUT -p tcp --dport 80 -m recent --name www --set -j
ACCEPT
```

Utworzenie zbiorów adresów prywatnych i ich blokowanie

```
ipset --create priv nethash
ipset --add priv 10.0.0.0/8
ipset --add priv 172.16.0.0/12
ipset --add priv 192.168.0.0/16
ipset --add priv 169.254.0.0/16
iptables -A INPUT -m set --set priv src -j DROP
```

🕮 Literatura

http://netfilter.samba.org
http://www.linuxdoc.org/HOWTO/mini/TransparentProxy.html
http://www.squid-cache.org
http://mr0vka.eu.org/tlumaczenia/index.htm
http://www.netfilter.org

Zadania:

Każde zadanie należy wykonać uprzednio czyszcząc wszystkie wpisy zapory sieciowej. Dodatkowo każdy wpis należy przetestować czy działa zgodnie z założeniami.

1. Ograniczyć maksymalną wielkość pakietu ICMP-echo do 1kB.
2. Ograniczyć do 3 na minutę liczbę pakietów ICMP-echo.
3. Utworzyć statystyki odbierania pakietów.
4. Ograniczyć żądania HTTP do 2kB wielkości i do 3 na minutę.
5. Logować pakiety, które naruszają regułę numer 4, ale ich nie usuwać.
6. Zmienić wartość pola TTL dla każdego pakietu na 56 i sprawdzić wykorzystując sniffer.
7. Usuwać pakiety większe niż 3 kB zwracając komunikat błędy ICMP-net-unreachable.
8. Zmienić wartość pola MSS – sprawdzić wykorzystując sniffer.
9. Wykorzystać moduł comment w celu opisania reguł zapory sieciowej.
10. Wykorzystać moduł pkttype w celu zablokowania pakietów ICMP-echo wysyłanych na adres rozgłoszeniowy. Wcześniej należy zdjąć blokadę jądra systemu:

```
echo ”0” > /proc/sys/net/ipv4/icmp_echo_ignore_broadcast
```