Ściana ogniowa w systemie operacyjnym LINUX

Autor: Gładysz Krystian IVFDS

STRESZCZENIE

Codziennie do sieci Internet podłącza się około kilku do kilkudziesięciu tysięcy nowych komputerów. Sieć Internet jako największa sieć komputerowa, oferująca dużo możliwości m.in.: pocztę elektroniczną, możliwość prowadzenia rozmów on-line, tzw. chaty, bogaty zbiór informacji, gry sieciowe i inne możliwości, co powoduje przyciąganie coraz większej ilości użytkowników. Jednak ten wzrost niesie za sobą również zmniejszenie bezpieczeństwa lokalnych sieci komputerowych jak i pojedynczych komputerów, które są podłączone do sieci. Aby zniwelować rosnące niebezpieczeństwo ze strony sieci globalnej należy zastosować oprogramowanie zwane firewall'em (ścianą ogniową), który jest przeznaczony do ochrony systemów lokalnych przed wszelkiego rodzaju atakami zewnętrznymi na nasz komputer, jak również na całą sieć lokalną. Ściany ogniowe umożliwiają częściowe odizolowanie naszych urządzeń podłączonych do sieci Internet.

SPIS TREŚCI

1. FILTROWANIE PAKIETÓW.

1.1 Co to jest filtrowanie IP?

Filtr pakietów to takie oprogramowanie, które sprawdza nagłówki pakietów w trakcie jak docierają do maszyny, na której działa i decyduje o ich losie. Może zdecydować, że pakiet zostanie odrzucony (drop, tzn. że datagramy są usuwane i zupełnie ignorowane, tak jak by nigdy nie zostały odebrane), zaakceptowany (accept, tzn. pozwoli mu się przejść), lub coś bardziej skomplikowanego. Generalnie chodzi o sprawdzanie nagłówków i decydowanie o ich losie.

Możesz wskazać wiele różnych kryteriów określających, które pakiety chcesz filtrować, oto kilka z nich:

- Typ protokołu, np.: TCP, UDP itp.;
- Numer gniazda
- Typ protokołu: SYN/ACK, dane, ICMP Echo Request itp.;
- Adres źródłowy pakietu, czyli skąd pochodzi;
- Adres docelowy pakietu, czyli dokąd jest wysyłany;

Filtrowanie pakietów jest funkcją warstwy sieciowej, co oznacza, że nie ma ono nic wspólnego z aplikacją wykorzystującą połączenia sieciowe, a dotyczy tylko samych połączeń.

W Linuksie, filtrowanie pakietów jest wbudowane w Kornel. [1, 5, 6]

1.2 Dlaczego filtrujemy pakiety?

Są trzy powodu dla których filtrujemy pakiety:

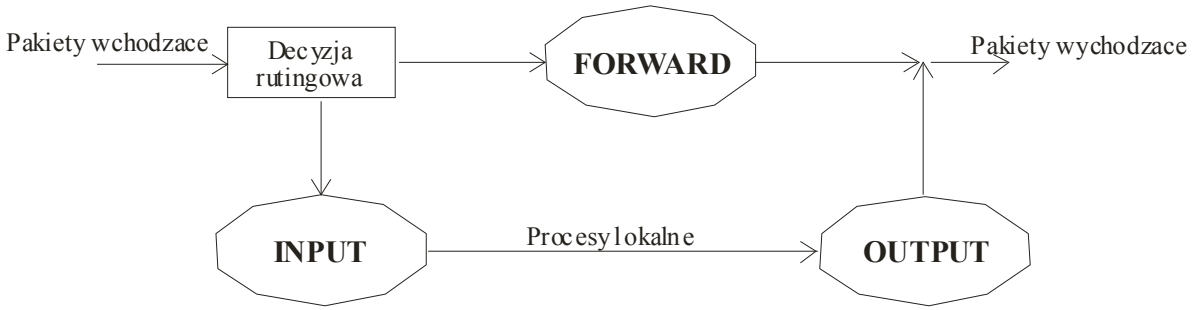
- Kontrola:
  Jeżeli wewnętrzną sieć łączysz z inną siecią np. Internetem za pomocą Linux'a masz okazję wpuścić trochę różnych typów ruchu i odrzucić inne.
- Bezpieczeństwo:
  Jeżeli komputer jest jedynym łączem miedzy Internetem a siecią lokalną, to możesz obłożyć restrykcjami to co jest skierowane do danej sieci lokalnej.
- Czujność:
  Źle skonfigurowana maszyna w sieci lokalnej może zdecydować o skierowaniu kilku pakietów do sieci zewnętrznej, lecz to zagrożenie może możemy wykryć dzięki odpowiednim ustawieniu filtru pakietów, który nas o tym zdarzeniu poinformuje i pozwoli na podjęcie odpowiedniej akcji. [5]

1.3 Budowa filtra.

Generalnie jądro systemu dzieli ruch firewalla na kategorie i do każdej z nich stosuje inny filtr. Dla każdej z tych kategorii otrzymujemy listę reguł nazywanych łańcuchami. Łańcuch natomiast określa nam, co mamy zrobić z pakietem, jeśli spełnia on określone warunki. Jądro po pierwsze podejmuje decyzję o tym gdzie powinien trafić dany pakiet, czyli podejmuje decyzję routingu pakietu. Wyróżniamy tutaj trzy reguły z odpowiednimi łańcuchami:

a) ***INPUT*** - *wchodząca ściana ogniowa* – wchodzący ruch, zanim zostaje zaakceptowany, jest testowany według zasad tejże ściany;
b) ***OUTPUT*** - *wychodząca ściana ogniowa* – wychodzący ruch zanim zostanie wysłany. jest testowany zgodnie z regułami danej ściany;
c) ***FORWARD*** - *przekazująca ściana ogniowa* – ruch, który jest przekazywany poprzez system, jest testowany zgodnie z regułami dla danej ściany.

Oprócz tych trzech standardowych kategorii, użytkownik może definiować także własne kategorie. [1,6]



Rys 1.1 Rysunek pokazuje jak pakiet przechodzi przez filtr.

1.4 Postępowanie pakietów.

Definicja zachowania się jądra w stosunku do pakietu jest tworzone przez budowanie reguł, które następnie przypisywane są odpowiednim łańcuchom. Możemy wyróżnić kilka możliwych postępowań z pakietami:

a) AKCEPT – akceptuje pakiet;
b) DROP – usuwa pakiet i zachowuje się tak jakby go nie było;
c) LOG – loguje pakiet;
d) REJENT – usuwamy pakiet i wysyłamy pakiet ICMP (informuje o odrzuceniu pakietu) do adresu źródłowego;
e) RETURN – skok na koniec łańcucha (w łańcuchu głównym) lub powrót z łańcucha zdefiniowanego przez użytkownika;
f) QUEUE – umieszcza pakiet w kolejce do dalszego przetwarzania;
g) REDIRECT – pakiet zostaje przekierowany na inny port; [5, 6]

1.5 Filtry pakietów do kierowania ich do odpowiednich klas.

a) *TC indem classifier*;
b) *route* – filter oparty o tablice routingu;
c) *fw* – jest to filtr oparty o zaznaczaniu pakietów przez filtr pakietów wbudowany w Kornel, który jest potężnym filtrem dzięki bardzo dużym możliwością netfiltera;
d) *u32* – filtr o dużej wydajności, który oparty jest na tablicy laszującej, pozwala na klasyfikowanie na podstawie zawartości nagłówków;
e) *RSVP classifier* – filtr ten klasyfikuje pakiety na podstawie spełnienia wymagania protokołu RSVP, służącego do przesyłania głosu i video w czasie rzeczywistym. [5, 6]

2. IPCHAINS

Ogólnie można powiedzieć, iż ściana ogniowa (firewall) to system, który ma za zadanie chronić sieć lokalną przed siecią globalną. Przez firewall przechodzi cały ruch sieciowy, zanim wejdzie do sieci lokalnej. Może mieć on postać routera, który filtruje przychodzące i wychodzące pakiety danych, bądź w bardziej złożonej postaci - jest to cała sieć routerów i serwerów, których zadaniem jest zapewnienie bezpieczeństwa poprzez eliminację niepożądanego ruchu sieciowego. Utworzenie routera filtrującego w systemie Linux możliwe jest dzięki doskonałemu narzędziu, jakim jest ipchains.

Polecenie, które służy do wyświetlenia wersji ipchains wygląda następująco:

```
$ ipchains --version
```

2.1 Opcje polecenia IPCHAINS dla operacji na łańcuchach.

Tabela 2.1.

Opcje	Znaczenie
-A	Dodaje nową regułę na koniec łańcucha
-C	Sprawdza pakiet zgodnie z regułami w łańcuchu (używany do testowania definiowanych łańcuchów)
-D	Usuwa wybraną regułę z łańcucha
-F	Oczyszcza wszystkie reguły z łańcucha
-I	Wstawia nową regułę na jakiejś pozycji w łańcuchu
-L	Wpisuje listę wszystkich reguł w łańcuchu
-M	Definiuje parametry maskowania lub wypisuje aktualne ustawienia
-N	Tworzy zdefiniowany przez użytkownika łańcuch o określonej nazwie
-P	Zmienia zasadę postępowania dla wbudowanego łańcucha
-R	Zastępuje regułę na jakiejś pozycji w łańcuchu
-S	Ustawia wartość czasu oczekiwania dla maskowania IP
-X	Usuwa pusty łańcuch
-Z	W danym łańcuchu zeruje liczniki pakietów i bajtów we wszystkich regułach

Opcje polecenia IPCHAINS. [1, 2, 3, 5]

Opcje -F, -L, -N, -P, -X, -Z umożliwiają operowanie całymi łańcuchami. Natomiast pozostałe opcje prócz opcji -M, -S, które służą do operacji na maskaradzie, które są wbudowane w ipchainsa, służą do manipulowania regułami wewnątrz łańcucha.

2.2 Parametry polecenia IPCHAINS do konstruowania filtrów.

Tabela 2.2.

Opcje	Znaczenie
-p *protokół*	Definiuje protokół, może przyjmować wartości numeryczne (takie jak w pliku /etc/protocols) lub może występować jako słowo kluczowe, np.: *tcp, udp, icmp* lub *all*.
-s *adres [/maska] [port] [:potr]*	Definiuje źródło pakietu; *adres* może być nazwą hosta, nazwą sieciową lub numerem IP z opcjonalna *maską* adresową; *port* może być nazwą lub numerem z pliku /etc/services; zakres portów może być określony jako *port:port*; jeśli wartość *port* nie jest określona, reguła dotyczy wszystkich portów.

Opcje	Znaczenie
-d *adres [/maska] [port] [:port]*	Definiuje adres przeznaczenia pakietu.
-j *cel*	Określa standard zasady postępowania lub zdefiniowania przez użytkownika łańcucha, do którego powinna być przekazana kontrola
-i *nazwa*	Określa nazwę interfejsu; można użyć częściowej nazwy, np.: eth+, czyli dana reguła ma zastosowanie do wszystkich interfejsów Ethernet, rozpoczynających się od eth
-b	Wskazuję regułę pasującą do danego pakietu IP w obu kierunkach
-y	Podaje tylko pakiety TCP SYN

Parametry polecenia IPCHAINS. [1, 2, 3, 5]

2.3 Standardowe zasady postępowania w celu dalszego przetwarzania.

Akcja pakietów mówią karmelowi, co robić z pakietami, które pasują do reguły. Aby podać akcję dla danego pakietu w ipchains używamy parametru '-j'. Nazwa akcji nie może przekraczać 8 liter i rozróżniane są tutaj małe i duże litery.

Gdy akcja nie zostanie podana, to taka reguła zachowuje się jako licznik pakietów danego rodzaju, który możemy wyświetlić używając komendy 'ipchains –L –v'.

Wyróżniamy sześć specjalnych akcji dla pakietu. Są one przedstawione w tabeli 2.3.

Każda inna akcja wskazuje na łańcuch zdefiniowany przez użytkownika. Pakiet zacznie przechodzenie przez reguły w tamtym łańcuchu. Jeśli nie zdecyduje on o losie tego pakietu, wróci on z powrotem i zostanie sprawdzony w aktualnym łańcuchu reguł.

Tabela 2.3.

Akcja	Znaczenie
AKCEPT	Zezwala na przejście pakietu
REJECT	Odrzuca pakiet zwracając do nadawcy odpowiedź ICMP, że adres docelowy jest nieosiągalny
DENY	Odrzuca pakiet, a do nadawcy nie jest wysyłany żaden komunikat
MASQ	Maskuje pakiety w ten sposób, iż wyglądają, jakby pochodziły z lokalnego hosta (akcja dopuszczalna tylko dla pakietów, które przechodzą przez łańcuch forward)
REDIRECT	Bez względu na przeznaczenie, pakiet jest dostarczany do portu w lokalnym hoście. Można go zastosować tylko w protokole TCP i UDP. Dodatkowo można podać port po jego nazwie, co pozwoli na przekierowanie go do tego portu (akcja ta jest dostępna tylko dla pakietów które przechodzą przez łańcuch input)
RETURN	Powrót do łańcucha, który wywołał ten łańcuch (Mówiąc prościej, oznacza to wyjście z łańcucha i użycie domyślnych zasad postępowania dla danego łąńcucha)

Zasady postępowania w celu dalszego przetwarzania. [1, 2, 3, 5]

2.4 Podawanie adresów IP: źródłowego i przeznaczenia.

- Pierwszym i najprostszym sposobem jest podawanie pełnej nazwy, takiej jak 'localhost' czy 'www.prz.rzeszow.pl'.
- Drugi sposób to podanie numeru IP, np.: 127.0.0.1 czy 212.134.15.188.

- Trzeci i czwarty sposób polegają na podaniu grupy adresów IP, tak jak np. 192.168.11.0/24 czy 192.162.11.0/255.255.255.0. Oba te sposoby podają adres od 192.168.11.0 do 192.168.11.255 włącznie. Cyfry po znaku mówią, które części adresu IP są ważne. Domyślnie przyjmowane jest '/32' czyli inaczej '/255.255.255.255', czyli ważne są wszystkie cyfry. Do podania wszystkich adresów IP służy polecenie '/0'. [5]

2.5 Inwersja.

Wiele flag może mieć swoje argumenty poprzedzone **'!'**, który jest traktowany jako 'not' czyli 'nie', by sprawdzane adresy nie były równe tym podanym. [5]

2.6 Podawanie portów UDP i TCP.

Gdy podajemy wyżej wymienione porty, możemy podać dodatkowy parametr specyfikujący port lub grupę portów, która nas interesuje. Grupę podaję się używając znaku ':', np.: 1024:1044 – ten przedział dotyczy 21 portów, od 1024 do 1044 włącznie. Jeśli ominiemy dolną granicę jest ona przyjmowana domyślnie na 0, a jeżeli ominiemy górna granicę, jest ona przyjmowana na 65535. Numery portów mogą być również podawane jako nazwy, np.: 'www', który wskazuje port 80.

Porty mogą być też poprzedzony znakiem '!', który spowoduje ich negowanie. [5]

2.7 Podawanie typów i kodów ICMP.

Protokół ICMP również umożliwia podawanie dodatkowego argumentu, ale ponieważ ICMP nie posiada portów, gdyż posiada typy i kody. Możesz podać je w formie nazw ICMP, które można uzyskać przez użycie polecenia: `ipchains –h ICMP` po opcji '-s' lub jako typ i kod numeryczny ICMP, w którym typ występuje po opcji '-s' a kod po opcji '-d'. Nazwy ICMP są raczej długie, więc używa się tylko tylu liter, by wskazać jednoznacznie na którąś ze zdefiniowanych.

Tabela 2.4.

Numer	Nazwa	Wymagane przez
0	echo-reply	Ping
3	destination-unreachable	ruch TCP/IP
5	redirect routing	jeśli nie działa demon routingu
8	echo-request	ping
11	time-exeeded	traceroute

Najbardziej popularne pakiety ICMP.

Nazwy ICMP nie mogą być poprzedzane parametrem '!'. [5]

2.8 Podawanie interfejsu.

Interfejs to fizyczne urządzenie, do którego pakiet dociera lub z którego wychodzi. By wyświetlić listę interfejsów, które są 'up' (działające) należy użyć polecenia 'ifconfig'.

Interfejs do pakietów przychodzących jest uważany za interfejs, z którego przyszły. Odpowiednio interfejs dla pakietów wychodzących to ten, przez który wyjdą pakiety po pokonaniu łańcucha wyjściowego. Pakiety, które przechodzą przez łańcuch przechodzący, trafiają również do interfejsu wyjściowego.

Podanie interfejsu, który jeszcze nie istnieje jest poprawny, gdyż reguła ta nie będzie dotyczyła niczego dopóki interfejs fizyczny nie zacznie działać.

Interfejs, który kończy się znakiem '+' będzie wskazywał na wszystkie interfejsy, które zaczynają się na dany ciąg znaków.

Nazwy interfejsów mogą być poprzedzone znakiem ‘!’ by oznaczyć wszystkie interfejsy oprócz wskazanego. [5]

2.9 Podawanie tylko pakietów TCP SYN.

Pakiety SYN, są to pakiety, które mają ustawioną flagę SYN i zgaszone flagi FIN i ACK. Służą one do blokowania pakietów z prośbą o połączenie, czyli pozwalają na powstrzymanie próby połączenia. Jest to przydatne wtedy, gdy potrzebujemy połączenia TCP tylko w jedną stronę, np.: zezwolenie na połączenie do zewnętrznego serwera WWW, ale nie połączenia z tego serwera.

Możemy użyć ‘!’, który oznacza, że zostanie przepuszczony każdy pakiet oprócz pakietów inicjujących połączenie. [5]

2.10 Przykładowe porty. [4]

Nazwa usługi	Numer portu	Protokół	Komentarz
echo	7	tcp/udp	Usługa echa
discard	9	tcp/udp	Usługa discard (odrzucać)
systat	11	tcp	Użytkownicy aktywni
daytime	13	tcp/udp	Usługa daty i czasu
charget	19	tcp/udp	Generator znaków
ftp-date	20	tcp	Usługa FTP, dane
ftp	21	tcp/udp	Usługa FTP, kontrola
telnet	23	tcp	Usługa telnetu
smtp	25	tcp	Mail
time	37	tcp/udp	Serwer czasu
tftp	69	udp	Trywialny transfer danych
gopher	70	tcp	Usługa gopher
finger	79	tcp	Usługa finger
http	80	tcp	Usługa WWW
kerberos-sec	88	tcp/udp	Kerberos
rtelnet	107	tcp	Usługa zdalnego protokołu telnet
pop2	109	tcp	Protokół urzędu pocztowego – wersja 2
pop3	110	tcp	Protokół urzędu pocztowego – wersja 3
nntp	119	tcp	Protokół transferu wiadomości sieciowych
ntp	123	udp	Protokół czasu sieciowego
snmp	161	udp	Usługa SNMP
snmptrap	162	udp	Pułapka SNMP
print-srv	170	tcp	Postscript sieciowy
irc	194	tcp	Protokół IRC (Internal Relay Chat)
ipx	213	udp	IPX przez IP
https	443	tcp/udp	Usługa generowania podpisu cyfrowego dla serwera WWW
who	513	udp	Usługa who (kto)
printer	515	tcp	Usługa drukowania sieciowego
router	520	udp	Usługa routingu
netnews	532	tcp	Usługa grup dyskusyjnych
wins	1512	tcp	Microsoft Windows Internet Name

2.11 Przykłady stosowania IPCHAINS.

Polecenie dodające regułę:

```
# ipchains -A input -s 192.168.11.1
```

reguła taka służy jako licznik pakietów podróżujących do adresu 192.168.11.1

```
# ipchains -A input -s 127.0.0.1 -p icmp -j DENY
```

reguła dołączania (-A) do łańcucha wejściowego (input) mówiąca, że pakiety nadchodzące z adresu 127.0.0.0 (‘-s 127.0.0.0’) i używające protokołu ICMP (‘-p ICMP’) powinny trafić do anulowania (‘ -j DENY’)

```
# ipchains -A input -s 0/0 -j DENY
```

ta reguła blokuje (DENY) wszystkie wchodzące do łańcucha wejściowego (input) adresy IP (‘-s 0/0’)

```
# ipchains -A input -d 192.168.1.2 25 -j ACCEPT
```

na podstawie tego polecenia będzie dodana nowa reguła (-A) do łańcucha wejściowego (input), w która będą akceptowane pakiety, jeżeli adres docelowy i port są prawidłowe; w powyższym przykładzie na port 25 (SMTP) będzie przyjmowany pakiet (‘ -j ACCEPT’) od lokalnego hosta o numerze IP 192.168.1.2

```
# ipchains -A input -s 0/0 -d 212.168.12.105 127 -p tcp -j DENY -i eth0
```

reguła dołączania (-A) do łańcucha wejściowego (input) mówiąca, że pakiety pochodzące z każdego źródła i dowolnego portu (‘-s 0/0’) skierowane na port nr 127 jednostki o nr IP 212.168.12.105 (‘-d 212.168.12.105 127’), używające portu tcp (‘-p tcp’) i interfejsu eth0 (‘-i eth0’) będą odrzucane (‘-j DENY’).

Polecenia kasujące regułę:

- Pierwszy sposób polega na użyciu numeru reguły, który ma być skasowany (metodę tą stosujemy, gdy znamy numer reguły, która chcemy usunąć)

```
# ipchains –D input 1
```
//polecenie skasuje regułę numer 1 w łańcuchu wejściowym

- Drugi sposób polega na lustrzanym odbiciu polecenia -A, ale zamiast polecenia -A pisujemy -D. (używamy tego polecenia, gdy mamy złożony zestaw reguł i nie chce nam się liczyć, która numer ma reguła, którą chcemy usunąć). Składnie ta musi być dokładnie taka sama jak –A. Jeśli będzie wiele takich samych reguł, tylko pierwsza zostanie skasowana.

```
# ipchains –D input –s 127.0.0.1 –p icmp –j DENY
```
// kasujemy (-D) z łańcucha wejściowego (input) regułę mówiącą, że pakiety nadchodzące z adresu 127.0.0.0 (‘-s 127.0.0.0’) i używające protokołu ICMP (‘-p ICMP’) powinny trafić do anulowania (‘ -j DENY’)

2.12 Przykładowy plik konfiguracyjny.

Warto skonstruować sobie plik zawierający potrzebne reguły i umieścić jego wywołanie w pliku startowym maszyny (np. /etc/rc.d/rc.local lub w innym, zależnie od dystrybucji LINUX).

Oto przedstawiam przykładowy plik w listingu 1.

LISTING

```
1. ipchains -F input
2. ipchains -F output
3. ipchains -F forward
4. ipchains -P input ACCEPT
5. ipchains -P output ACCEPT
6. ipchains -P forward DENY
7. ipchains -A input -s 0/0 -d 212.168.194.1 -j DENY
8. ipchains -A output -s 0/0 -d 212.168.194.1 -j DENY
9. ipchains -A forward -s 192.168.11.1 -d 0/0 -j MASQ
```

```
10. ipchaims -A input -p tcp -s 192.168.11.0/24 -d 212.168.194.1 20 -j
    ACCEPT
11. ipchaims -A input -p tcp -s 192.168.11.0/24 -d 212.168.194.1 21 -j
    ACCEPT
12. ipchaims -A input -p udp -s 192.168.11.0/24 -d 212.168.194.1 21 -j
    ACCEPT
13. ipchains -I input -p tcp -s 0/0 -d 212.168.194.1 23 -j ACCEPT
14. ipchains -I input -p udp -s 0/0 -d 212.168.194.1 25 -j ACCEPT
15. ipchains -I input -p udp -s 0/0 -d 212.168.194.1 37 -j ACCEPT
16. ipchains -I input -p tcp -s 0/0 -d 212.168.194.1 37 -j ACCEPT
17. ipchains -I input -p tcp -s 0/0 -d 212.168.194.1 80 -j ACCEPT
18. ipchains -I input -p tcp -s 192.168.11.0/24 -d 212.168.194.1 110 -j
    ACCEPT
19. ipchains -I input -p tcp -s 0/0 -d 212.168.194.1 194 -j ACCEPT
20. ipchains -I input -p tcp -s 0/0 -d 212.168.194.1 515 -j ACCEPT
21. ipchains -I input -p tcp -s 0/0 -d 212.168.194.1 532 -j ACCEPT
22.
23. ipchaims -A output -p tcp -s 0/0 -d 212.168.194.1 7 -j ACCEPT
24. ipchaims -A output -p udp -s 0/0 -d 212.168.194.1 7 -j ACCEPT
25. ipchaims -A output -p tcp -s 0/0 -d 212.168.194.1 20 -j ACCEPT
26. ipchaims -A output -p tcp -s 0/0 -d 212.168.194.1 21 -j ACCEPT
27. ipchains -I output -p tcp -s 0/0 -d 212.168.194.1 23 -j ACCEPT
28. ipchains -I output -p udp -s 0/0 -d 212.168.194.1 25 -j ACCEPT
29. ipchains -I output -p tcp -s 0/0 -d 212.168.194.1 80 -j ACCEPT
30. ipchaims -A output -p tcp -s 0/0 -d 212.168.194.1 109 -j ACCEPT
31. ipchains -I output -p tcp -s 0/0 -d 212.168.194.1 110 -j ACCEPT
32. ipchains -I output -p tcp -s 0/0 -d 212.168.194.1 194 -j ACCEPT
33. ipchains -I output -p tcp -s 0/0 -d 212.168.194.1 515 -j ACCEPT
34. ipchains -I output -p tcp -s 0/0 -d 212.168.194.1 532 -j ACCEPT
```

Opis:

Tabela 2.5.

Linie	Komentarz
1-3	Oczyszczenie wszystkich reguł we wszystkich łańcuchach
4-6	Ustawia domyślnie akceptacje pakietów na wszystkich łańcuchach: input, output oraz odrzucenie pakietów w łańcuchu forward
7-8	Zablokowanie dostępu do naszej maszyny (serwera), zarówno w filtrze wejściowym (input), jak i wyjściowym (output)
9	Reguła maskowania adresów sprawia, że Twój serwer staje się routerem, który udostępnia poszczególnym komputerom w sieci wewnętrznej połączenie z siecią zewnętrzną
10-21	Po wcześniejszym zablokowaniu wszystkich reguł w filtrze wejściowym, odblokowujemy poszczególne porty, przez które chcemy wpuszczać pakiety do naszego serwera. W powyższym przypadku są to następujące porty: 20 i 21 – odpowiedzialne za ftp, 23 – telnet, 25 – stmp, 37 – time, 80 – http, 110 – pop3, 194 – irc, 515 – sprinter, 532 – netnews. Dodatkowo w liniach 10-12 i 18 są dopuszczane pakiety tylko z adresów od 192.168.11.0 do 192.168.11.255, w pozostałych przypadkach ze wszystkich
23-34	Po wcześniejszym zablokowaniu wszystkich reguł w filtrze wyjściowym, odblokowujemy poszczególne porty, przez które chcemy wypuszczać pakiety z naszego serwera. W powyższym przypadku są to następujące porty: 7 – echo, 20 i 21 – ftp, 23 – telnet, 25 – stmp, 80 – http, 109 – pop2, 110 – pop3, 194 – irc, 515 – sprinter, 532 – netnews.

3. IPTABLES.

Kolejnym profesjonalnym narzędziem administratora do zabezpieczenia swojego komputera, czyli stworzenia ściany ogniowej (firewall'a) jest iptables. Jest on następcą ipchains'a.

3.1 Zasady działania IPTABLES.

Jądro rozpoczyna pracę z trzema predefiniowanymi listami reguł w tabeli filtrującej. Są to łańcuchy input, output i forward. Każdy pakiet docierający do hosta jest sprawdzany pod kątem miejsca przeznaczenia. Na tej podstawie kernel decyduje, czy ma zostać przekazany do sieci położonej gdzieś dalej czy skierowany do niego samego. Pakiet skierowany do tego komputera pozostaje sprawdzony przez reguły łańcucha input. Jeżeli jego obecność zostanie tu zaakceptowana, pakiet będzie dopuszczony do procesu, do którego został skierowany. W przeciwnym wypadku zostanie odrzucony. Jeżeli posiadasz włączone przekazywanie pakietów i pakiet jest przeznaczony dla innego interfejsu sieciowego, pakiet przechodzi przez zestaw reguł łańcucha forward. Reguły łańcucha zadecydują, czy może zostać przesłany dalej czy zostać odrzucony. Procesu uruchamiane na naszym hoscie także mogą być źródłem pakietów wydostających się do Internetu. Takie pakiety przechodzą przez łańcuch output. Po akceptacji docierają do interfejsu sieciowego. [1, 2, 5]

3.2 Opcje polecenia IPTABLES dla operacji na łańcuchach.

Tabela 3.1.

Opcje	Znaczenie
-A	Dodaje nową reguły do łańcucha
-C	!Sprawdza pakiet zgodnie z regułami w łańcuchu (używany do testowania definiowanych łańcuchów)
-D	Kasuje wybraną regułę z łańcucha lub pierwszą pasującą regułę z łańcucha
-F	Oczyszcza wszystkie reguły z łańcucha oraz utworzenie nowego łańcucha
-I	Wstawia reguły na określoną pozycję w łańcuchu
-L	Wpisuje listę wszystkich reguł w łańcuchu
-N	Tworzenie nowego łańcucha o określonej nazwie przez użytkownika
-P	Zmienia zasadę postępowania dla wbudowanego łańcucha
-R	Wymienia regułę na jakiejś pozycji w łańcuchu
-X	Kasuje pusty łańcuch
-Z	W danym łańcuchu zeruje liczniki pakietów i bajtów we wszystkich regułach

Opcje polecenia IPTABLES. [1, 2, 3, 5]

Opcja te możemy podzielić na te, które umożliwiają manipulowanie regułami w środku łańcuchów, są to: -A, -I, -R, -D, natomiast pozostałe opcje służą do operowania na łańcuchach output, input, forward.

3.3 Parametry polecenia IPTABLES do konstruowania filtrów.

Tabela 3.2.

Parametry	Znaczenie
-p *[!]protokół*	Użycie reguły dla konkretnego protokołu, można używać zamiennie nazwy '--protocol';
-s *[!]adres [/maska]*	Definiuje źródło pakietu, można używać zamiennie nazw: '--source' lub '—src';
-d *[!]adres [/maska]*	Definiuje adres przeznaczenia pakietu, można używać zamiennie nazw: '--destination' lub '--dst';
-sport *[!][port]:[port]*	Określenie portu źródłowego
-dpotr *[!][port]:[port]*	Określenie portu docelowego
-j *cel*	Określa standard zasady postępowania lub zdefiniowania przez użytkownika łańcucha, do którego powinna być przekazana kontrola;
-i *[!]nazwa_interfejsu*	Określa nazwę interfejsu sieciowego wejściowego, może być zastąpione przez '—in-interface', używany dla łańcucha INPUT i FORWARD;
-o *[!]nazwa_interfejsu*	Określa nazwę interfejsu sieciowego wyjściowego, może być zastąpione przez '—out-interface', używany dla łańcucha OUTPUT i FORWARD;
-v	Powoduje wyświetlenie bogatszych wyników;
-n	Powoduje wyświetlenie adresu IP i portów tylko jako liczby, nie próbuje zamienić ich na odpowiadające im nazwy
-x	Powoduje, że wszystkie liczby w wyniku pokazywane są dokładnie, bez zaokrąglania;

Parametry polecenia IPTABLES. [1, 2, 3, 5]

W powyższej tabeli w parametrach -s i -d *adres* może być nazwą hosta, nazwą sieciową lub numerem IP z opcjonalna *maską* adresową; *port* może być nazwą lub numerem z pliku /etc/services; zakres portów może być określony jako *port:port*; jeśli wartość *port* nie jest określona, reguła dotyczy wszystkich portów.
Natomiast w parametrze -p *protokół* może przyjmować wartości numeryczne (takie jak w pliku /etc/protocols) lub może występować jako słowo kluczowe, np.: *tcp, udp, icmp* lub *all*.
W opcjach -i oraz -o w *nazwa_interfejsu* można użyć częściowej nazwy, np.: eth+, czyli dana reguła ma zastosowanie do wszystkich interfejsów Ethernet, rozpoczynających się od eth)

3.4 Funkcja NAT.

NAT (Network Address Translation) jest technologią, która umożliwia przekształcanie adresów IP, co umożliwia np.: ukrycie sieci pod jednym adresem IP, tworzenia transparentnych proxy, zmianę prawdziwych adresów hostów, jednym słowem maskowanie adresów IP. Wyróżniamy trzy rodzaje:

a) Jeden do jednego
b) Jeden do wielu
c) Wielu do jednego

Funkcja NAT jest zintegrowane w iptables. Dodatkowo NAT dzielimy na dwa rodzaje:

- NAT źródłowy (SNAT ang. Source NAT) – używamy go, gdy modyfikujemy adres źródłowy np. maskarada;
- NAT docelowy (DNAT ang. Destination NAT) – używamy go, podczas modyfikacji adresu pakietu docelowego np. rozkładanie obciążenia czy transparentne proxy. [6]

3.5 Budowa funkcji NAT.

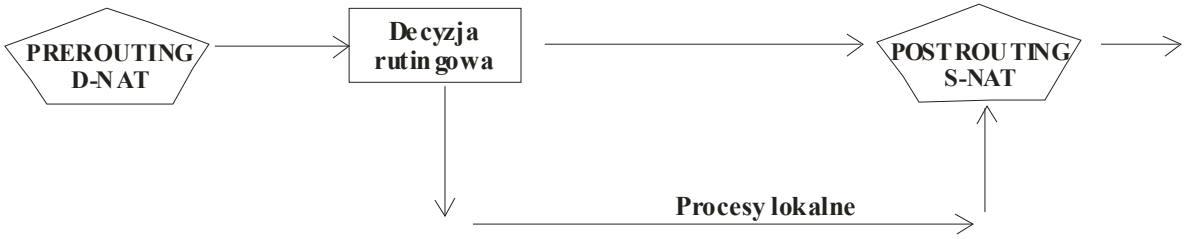
W tabeli NAT wyróżniamy trzy łańcuchy, których reguły są sprawdzane dopóki nie nastąpi zgodność. Przedstawione są one w tabeli 3.3. [6]

Tabela 3.3.

Tablica NAT	Opis
PREROUTING	Wykorzystuje się go do egzaminowania wchodzących pakietów w NAT-cie docelowym (DNAT)
POSTROUTING	Wykorzystuje się go do egzaminowania wychodzących pakietów w NAT-cie źródłowym (SNAT)
OUTPUT	Stosowany dla pakietów wygenerowanych przez procesy lokalne, używany w NAT-cie docelowym (DNAT)

Łańcuchy zawarte w tablicy NAT.

Na poniższym schemacie przedstawione jest gdzie powyższe reguły łańcucha są sprawdzane:



Rys 3.1 Rysunek przedstawia miejsca sprawdzania łańcuchów funkcji NAT

3.6 Opcje IPTABLES używane w NAT.

Podstawową opcja przy używaniu iptables w NAT jest opcja „**–t nat”** (tabela nat). Wyróżniamy tutaj również opcję **„-s”**, **„-d”, „-i”** oraz **„-o”** (ich znaczenie opisane jest w podpunkcie 3.3). Odpowiednio opcjach „-i” i „-o” zostaną wybrane w zależności od łańcucha, na którym dokładamy regułę. Tak, więc w łańcuchu PREROUTING wybieramy interfejs wejściowy, czyli „-i”, a przy łańcuchu POSTROUTING wybieramy interfejs wyjściowy – „-o”. Jeżeli nie podamy właściwie interfejsu, co do łańcucha to iptables zakomunikuje nam to wypisaniem błędu.

Kolejna opcja, która może być nam potrzebna jest opcja **„-p”** (opisana w rozdziale 3.3). Opcja ta przydatna jest nam, kiedy chcemy przekierować konkretny protokół. [6]

4. IPFWADM.

Narzędzie to jest używane do tworzenia reguł ściany ogniowej dla wszystkich jąder starszych od wersji 2.2.0. Składnia poleceń ipfwadm jest skomplikowana, ze względu na możliwość realizowania wielu skomplikowanych zadań.

4.1 Zasady działania IPFWADM.

Polecenie to ma wiele różnych argumentów odnoszących się do konfiguracji firewall'a IP. Ogólnie składnia jest następująca:
ipfwadm *kategoria polecenie parametry [opcje]*
Kategoria musi być podana tylko jedna z kategorii, które opisane są w tabeli 4.1 [1]

4.2 Kategorie IPFWADM dla określenia datagramow.

Tabela 4.1.

Kategorie	Znaczenie
-I	Reguła wejściowa;
-O	Reguła wyjściowa;
-F	Reguła przekazująca datagramy, które do niej pasują;

Kategorie IPFWADM. [1]

4.3 Opcje polecenia IPFWADM dla operacji na łańcuchach.

Tabela 4.2.

Opcje	Znaczenie
-a *[polityka]*	Dodanie nowej reguły;
-i *[polityka]*	Wstawianie nowej reguły;
-d *[polityka]*	Usunięcie istniejącej reguły;
-p *polityka*	Ustawienie polityki domyślnej;
-l	Wylistowanie wszystkich istniejących reguł;
-f	Usunięcie wszystkich istniejących reguł;

Opcje polecenia IPFWADM.

W powyższych poleceniach używa się parametru *polityka*, która przyjmuje następujące formy:

- **akcept** – pozwala na odbiór, przekazywanie lub wysyłanie pasujących datagramów
- **deny** – nie pozwala na odbiór, przekazywanie lub wysyłanie pasujących datagramów
- **rejent** – nie pozwala na odbiór, przekazywanie lub wysyłanie pasujących datagramów i wysyła komunikat błędu ICMP do hosta, który przysłał datagram [1]

4.4 Parametry polecenia IPFWADM do określenia datagramów.

Tabela 4.3.

Parametry	Znaczenie
-P *protokół*	Użycie reguły dla konkretnego protokołu;
-S *adres /maska/[port]*	Definiuje źródłowy adres IP, do którego pasuje ta reguła;
-D *adres /maska/[port]*	Definiuje adres docelowy IP, do którego pasuje ta reguła;

-V *adres*	Określenie adresu interfejsu sieciowego, na którym pakiet jest odbierany (-I) lub, z którego jest wysyłany (-O). Pozwala to na stworzenie reguł dotyczących tylko niektórych interfejsów sieciowych komputera;
-W *nazwa*	Określenie nazwy interfejsu sieciowego. Ten argument działa w ten sam sposób co -V, ale podajesz nazwę urządzenia zamiast adresu.

Parametry polecenia IPFWADM.

W powyższej tabeli w parametrach używamy opcji:

- *adres* może być nazwą hosta, nazwą sieciową lub numerem IP;
- *maska* adresowa, jeżeli nie poda się maski sieci, zostanie przyjęta maska „/32”;
- *port* może być nazwą lub numerem z pliku /etc/services; zakres portów może być określony jako port_pierwszy:port_ostatni; jeśli wartość port nie jest określona, reguła dotyczy wszystkich portów; musisz podać protokół za pomocą argumentu -P, aby ta opcja zadziałała.
- *protokół* może przyjmować wartości numeryczne (takie jak w pliku /etc/protocols) lub może występować jako słowo kluczowe, np.: tcp, udp, icmp lub all. [1]

4.5 Argumenty opcjonalne IPFWADM.

Tabela 4.4.

Opcje	Znaczenie
-b	Jest używany dla trybu dwu kierunkowego. Do tej opcji pasuje ruch w obie strony pomiędzy zadanymi adresami źródłowymi i docelowymi. Opcja ta zaoszczędza tworzenie dwóch reguł: jednej do wysyłania i drugiej do odbierania;
-o	Pozwala na zapisywanie pasujących datagramów do logu jądra. wszelkie datagramy pasują do reguły będą zapisywane jako komunikaty jądra. Jest to użyteczna opcja do wykrywania nieautoryzowanego dostępu.
-y	Ta opcja jest używana do filtrowania połączeniowych datagramów TCP. Dzięki niej reguła filtruje tylko datagramy podejmujące próbę zestawienia połączeń TCP. Pasować będą jedynie datagramy posiadające ustawiony bit SYN i wyzerowany bit ACK. Jest to użyteczna opcja do filtrowania prób połączeń TCP i ignorowania innych protokołów.
-k	Jest używany do filtrowania datagramów – potwierdzeń TCP. Ta opcja powoduje, że do reguły pasują tylko datagramy będące potwierdzeniem odbioru pakietów próbujących zestawić połączenie TCP. Będą pasować jedynie datagramy, które mają ustawiony bit ACK. Opcja ta jest użyteczna do filtrowania prób połączeń TCP i ignorowania wszystkich pozostałych protokołów.

Argumenty opcjonalne IPFWADM. [1]

LITERATURA

[1] C. Kirch, T. Dawson „LINUX podręcznik administratora sieci“ wydawnictwo RM Warszawa 2000

[2] M. Canou, J. Georzen, A. Van Couwenberghe „Debian Linux – Księga eksperta“ wydawnictwo Helion Gliwice 2001

[3] Craig Hunt „Serwery sieciowe linuksa” wydawnictwo MIKOM Warszawa 2000

[4] Bruce Hallberg „Sieci komputerowe, kurs podstawowy” wydawnictwo ”Edition2000” Kraków 2001

[5] HOWTO: firewall, iptables, ipchains

[6] Zasoby internetowe:
http://elektron.elka.pw.edu.pl/~kmadej
http://www.ziolek.piotrkow.pl/linux/iptablesi.htm
http://linuxpub.gnu.pl/