# Pakiet Iptables

mgr inż. Rafał Jachowicz
Instytut Informatyki Stosowanej Politechniki Łódzkiej
rjachowicz@kis.p.lodz.pl
opracowane na podstawie materiałów mgr inż. Łukasza Jopka (ljopek.kis.p.lodz.pl)

## Filtrowanie pakietów i filtrowanie stanowe

Filtrowanie pakietów oraz filtrowanie stanowe są jedną z podstawowych funkcjonalności firewalli, pozwalają bowiem na decydowanie, które pakiety zostaną przyjęte, które przekazane dalej, a które będą odrzucone. Pojęcie filtrowania pakietów jest jednak starsze od samych firewalli, wcześniej bowiem routery posiadały możliwość filtrowania i były pierwszymi urządzeniami stworzonymi z myślą o zabezpieczeniu naszej sieci przed intruzami. Mechanizm filtrowania ma za zadanie sprawdzenie każdego pakietu sieciowego przesyłanego za pośrednictwem maszyny, na której zainstalowane jest oprogramowanie służące do filtracji pakietów. Odbywa się to zgodnie ze zdefiniowanymi wcześniej regułami – albo go odrzuca, albo przyjmuje, albo przekazuje dalej. Mechanizm filtracji operuje głównie na podstawie informacji zawartych w nagłówkach pakietów TCP/IP.

### 1. Czym jest i jak działa filtrowanie pakietów?

Filtrowanie pakietów opiera się o warstwę 3 i 4 modelu OSI. Oznacza to, że dla warstwy 3 analizowane będą adresy IPv4 (w przypadku korzystania z protokołu IP, czyli najczęściej), a dla warstwy 4 będą to nagłówki pakietów protokołów warstwy 4, np. TCP, UDP, ICMP itp.

Mechanizm filtracji w warstwie 3 polega na sprawdzaniu adresu źródłowego i docelowego w nagłówku pakietu IP i podejmowaniu odpowiednich działań w wypadku wychwycenia pakietu spełniającego pewne zdefiniowane reguły firewalla. Jak to było wspomniane wcześniej, filtrować można również w oparciu o protokoły warstwy 4, można np. blokować pakiety protokółu ICMP odpowiadające za działanie popularnego polecenia *ping*. Choć przydatne w przypadku testowania sieci intruzowi pozwolić mogą na sprawdzenie, jakie adresy wykorzystywane są w danej sieci. W przypadku innych protokołów warstwy 4: TCP i UDP można blokować odpowiednie porty. Pozwala to np. zablokować użytkownikom sieci korzystanie z pewnych usług (np. WWW, FTP czy innej aplikacji).

### 2. Czym jest i jak działa filtrowanie stanów?

Filtrowanie stanowe jest podobnie do filtrowania pakietów, ale dodatkowo śledzi i analizuje kontekst komunikacji. Firewalle tego typu utrzymują tablice z informacjami na temat aktualnych połączeń. Ponieważ filtrowanie stanowe, w odróżnieniu od zwykłego filtrowania pakietów, wiąże się z analizą danych warstw wyższych, następuje ono głównie w warstwie 4 (analizuje np. takie dane jak numery sekwencyjne TCP). Na podstawie danych z

poprzednich sesji i obecnej można wykryć sesję o nienaturalnym przebiegu. Dlatego też filtrowanie stanowe jest w stanie wykryć i powstrzymać bardziej złożone i wyszukane formy ataków za pomocą protokołów wyższego poziomu, np. TCP czy UDP.

Filtrowanie pakietów i filtrowanie stanowe nie jest jednak tylko funkcjonalnością firewalli służącą do powstrzymywania ataków. Pełni ona także ważną rolę w sterowaniu ruchem w sieci, przekazywaniem pakietów itp.

**3. Filtrowanie pakietów za pomocą iptables**

Pakiet iptables operuje na tabelach:

- `mangle`
- `nat`
- `filter`

oraz łańcuchach:

- `PREROUTING`
- `POSTROUTING`
- `INPUT`
- `OUTPUT`
- `FORWARD`

Tabela **mangle** używana jest do zmiany nagłówków pakietów, tzn. takich pól, jak TOS (Type Of Service), TTL (Time To Live), jak i do filtrowania stanowego. Tabela **nat** służy do translacji adresów sieciowych, gdy siec używa NAT (Network Address Translation). W niej ustalamy „maskaradę” adresów sieciowych i przekierowujemy pakiety. Tabela **filter** służy do właściwego filtrowania pakietów. Tutaj definiowane są podstawowe reguły firewalla, według których decyduje on, czy dany pakiet zaakceptować czy odrzucić.

Łańcuchy :

**INPUT** - wywoływany dla pakietów przybywających z sieci, przeznaczonych dla lokalnej maszyny.

**OUTPUT** - wywoływany dla pakietów tworzonych lokalnie i wychodzących poza maszynę.

**FORWARD** - wywoływany dla pakietów trasowanych przez lokalną maszynę, lecz nie przeznaczonych dla niej.

**PREROUTING** - wywoływany dla pakietów z zewnątrz jeszcze przed ich trasowaniem.

**POSTROUTING** - wywoływany dla pakietów, które właśnie opuszczają maszynę po trasowaniu.

Różne tabele iptables dysponują różnymi wbudowanymi łańcuchami. Np. tabela **filter** zawiera/obsługuje łańcuchy INPUT, FORWARD i OUTPUT. Tabela **nat** zawiera/obsługuje łańcuchy PREROUTING, OUTPUT i POSTROUTING, ale już tabela **mangle** zawiera/obsługuje wszystkie rodzaje łańcuchów. Wymienione tabele zawierają wszystkie reguły, korzystające ze wspomnianych łańcuchów, potrzebne do zarządzania pakietami na maszynie lokalnej.

Jeżeli chodzi o połączenia: każde połączenie rejestrowane jest w maszynie stanów iptables (filtrowanie stanów). Wyróżnić można następujące stany połączeń: NEW, ESTABLISHED, RELATED, INVALID. Aktualną tablicę stanów połączeń znaleźć można w pliku */proc/net/ip_conntrack* w systemie Linux, można także użyć poniższego polecenia, aby wyświetlić aktualną tablicę stanów połączeń:

```
cat /proc/net/ip_conntrack
```

Opis stanów połączeń:

**NEW (nowy)** - Jeśli pakiet przychodzi ze zdalnej maszyny lub też zostanie do niej wysłany z naszej maszyny w celu nawiązania nowego połączenia to zostanie on potraktowany jako NEW, czyli nowy. Każdy następny pakiet skojarzony z tym połączeniem już tak potraktowany nie zostanie.

**ESTABLISHED (połączony)** - Pakiety zostaną potraktowane jako ESTABLISHED jeśli nawiązane zostanie połączenie w dwie strony ze zdalną maszyną.

**RELATED (powiązany)** – Pakiety tak oznaczone służą do obsługi innego obecnie istniejącego połączenia, np. takiego, które jest częścią multipołączenia w protokołach typu FTP, albo jako błędne pakiety związane z istniejącymi połączeniami (np. pakiet błędu ICMP związany z obecnie występującymi połączeniami).

**INVALID (niepoprawny)** - Pakiety, które nie mogą być zaklasyfikowane jako jedna z powyższych trzech kategorii traktowane są jako INVALID. Pakiety te nie są jednak automatycznie odrzucane, ale pozostaną ciągle aktywne. Można je wykorzystać używając innych zasad, np. dynamicznie ustawić taktykę łańcuchową.

Podstawowy schemat budowy komendy iptables wygląda następująco:

```
iptables [-t table] -action chain rule_specification
```

gdzie:

*table* - umożliwia wybór tablicy, dla której obowiązywać będzie dana reguła, brak tej informacji w regule powoduje zapisane jej do tabeli *filter*.

*action* - rodzaj akcji, jaka ma zostać wykonana na podanej tabeli oraz łańcuchu. Do najprostszych należą dodawanie, usuwanie itp.

*chain* - rodzaj łańcucha, wobec którego mamy zamiar wykonać akcję we wskazanej tabeli.

*rule_specification* - dalszy opis reguły (by dowiedzieć się więcej w terminalu proszę wpisać *iptables --help*).

Opis akcji:

- **iptables -t TABELA -A ŁAŃCUCH OPIS_REGUŁY**

Dodaje regułę na koniec wskazanego łańcucha we wskazanej tabeli.

- **iptables -t TABELA -D ŁAŃCUCH OPIS_REGUŁY**

Usuwa zadaną regułę z łańcucha (trzeba podać całą regułę, którą chcemy usunąć).

- **iptables -t TABELA -D ŁAŃCUCH NUMER_REGUŁY**

Usuwa regułę o podanym numerze (reguły numerowane są w zbiorach pod łańcuchami względem wierszy - od góry do dołu, począwszy od 1).

- **iptables -t TABELA -I ŁAŃCUCH NUMER_REGUŁY OPIS_REGUŁY**

Dodaje regułę we wskazanym miejscu zbioru pod wskazanym łańcuchem (reguły numerowane są w zbiorach pod łańcuchami względem wierszy - od góry do dołu, począwszy od 1).

- **iptables -t TABELA -R ŁAŃCUCH NUMER_REGUŁY OPIS_REGUŁY**

Zamienia regułę wskazaną numerem na regułę opisaną w poleceniu (OPIS_REGUŁY).

- **iptables -t TABELA -L ŁAŃCUCH**

Listuje reguły we wskazanym łańcuchu.

- **iptables -t TABELA -N ŁAŃCUCH**

Tworzy łańcuch użytkownika o zadanej nazwie.

- **iptables -t TABELA -X ŁAŃCUCH**

Usuwa łańcuch użytkownika.

- **iptables -t TABELA -P ŁAŃCUCH DOMYŚLNY_CEL**

Ustawia policy (domyślną akcję) zadanego łańcucha.

Ważniejsze opcje dotyczące dalszego opisu reguły (podane ciągi znaków są częścią fragmentu OPIS_REGUŁY, podanej w podstawowym schemacie komendy iptables):

`-p[!] **PROTOKÓŁ**`

lub

`--protocol [!] **PROTOKÓŁ**`

ustala regułę dla danego protokołu: np. TCP (możemy wpisać 6 w miejsce protokołu), UDP (17), ICMP (1) lub dla wszystkich protokołów: all (0). Jeśli użyjemy znak wykrzyknika ! to spowoduje to inwersje znaczenia, zatem *-p! tcp* będzie oznaczać regułę dla protokołów różnych od TCP.

`-s[!] **ADRES_IP**`

lub

`--source[!] **ADRES_IP**`

ustala regułę dla pakietów o adresie źródłowym ADRES_IP, w który możemy wpisać pojedynczy adres (np. 192.168.0.1) lub zakres adresów (np. 192.168.0.0/24 lub 192.168.0.0/255.255.255.0). Podobnie jak wcześniej, znak ! dokonuje inwersji znaczenia.

`-d[!] **ADRES_IP**`

lub

`--destination[!] **ADRES_IP**`

ustala regułę dla pakietów o adresie docelowym ADRES_IP, zasady podobne jak wyżej.

`-i[!] **INTERFEJS**`

lub

`**--in-interface[!] INTERFEJS**`

ustala regułę dla danego interfejsu (np. eth0, eth1, ppp0) do którego pakiet się odnosi. Tak jak poprzednio znak ! oznacza inwersję znaczenia. Stosowane wyłącznie dla łańcuchów

INPUT, FORWARD i PREROUTING. Znak + oznacza wszystkie interfejsy podobne do danego, np. eth+ będzie oznaczać interfejs eth0, eth1, eth2, itp.

`-o[!] INTERFEJS`

lub

`--out-interface[!] INTERFEJS`

ustala regułę dla interfejsu z którego pakiet wychodzi, stosowane wyłącznie dla łańcuchów OUTPUT, FORWARD, POSTROUTING.

`-sport[!] NUMER_PORTU`

lub

`-sourceport[!] NUMER_PORTU`

ustala regułę dla portów o podanym numerze. Możemy zamiast numeru portu wpisać nazwę usługi (zostanie przeszukany plik /etc/services) jak i zakres portów oddzielany dwukropkiem, np. *-sport 22:80* (dla portów 22, 23, ..., 80), *-sport! :80* (dla portów 81, 82, ..., 65535). Podane polecenie ma zastosowanie dla protokołów TCP oraz UDP.

`-dport[!] NUMER_PORTU`

lub

`--destinationport[!] NUMER_PORTU`

ustala reguł dla pakietów o numerze docelowym portu nr. Zasady jak wy_ej.

Ważnym elementem składni komendy jest zdefiniowanie zdarzenia jakie ma zostać wywołane, gdy wychwycony zostanie opisany pakiet. Zdarzenia te definiuje się, używając parametru **-j**. Definicję zdarzeń umieszcza się na końcu komendy. Najczęściej używane zdarzenia to:

• **ACCEPT** - przyjmij pakiet.

• **DROP** - odrzuć pakiet (tutaj pakiet zostanie odrzucony bez żadnego komunikatu)

• **RETURN** - powoduje powrót z łańcucha zdefiniowanego przez administratora do łańcucha, z którego został on wykonany.

• **REJECT** - odrzuca pakiet z odesłaniem komunikatu o błędzie, domyślnie ICMP port unreachable.

• **MARK** - pozwala na oznaczenie pakietu za pomocą wartości numerycznej podanej w parametrze *--set-mark*. Użyteczne w tabeli mangle, oraz zwykle używane do przekazywania informacji do iproute2.

• **MASQUERADE** - maskuje adres nadawcy pakietu, użyteczne wyłącznie w łańcuchu POSTROUTING tabeli NAT. Na ogół używane i zalecane do użycia w przypadku korzystania z dynamicznie przydzielanego adresu IP.

• **SNAT** - zmienia adres źródłowy pakietów i zapamiętuje zmianę dla danego połączenia. Najczęściej używane w przypadku połączenia LAN do Internetu przez łącze ze stałym adresem IP. Użyteczne wyłącznie w łańcuchu POSTROUTING tabeli NAT. Akceptuje parametr *--to-source*, który wskazuje adres IP do wstawienia jako adres źródłowy - zazwyczaj IP interfejsu internetowego.

• **REDIRECT** - przekierowuje pakiet do lokalnej maszyny, użyteczne w łańcuchach PREROUTING i OUTPUT tabeli nat. Posiada argument opcjonalny *--to-ports* pozwala na przekierowanie połączenia na dowolnie wybrany port lokalnej maszyny.

• **DNAT** - zmienia adres docelowy pakietów i zapamiętuje zmianę dla danego połączenia. Używane na ogół w celu przekazywania połączeń z interfejsu internetowego do maszyn w sieci LAN. Użyteczne w łańcuchach PREROUTING i OUTPUT tabeli NAT. Posiada także opcjonalny parametr *--to-destination*, określający adres z opcjonalnym portem docelowym (format zapisu to *adres:port*).

**4. Przykłady użycia:**

```
iptables -P INPUT DROP
```

blokujemy wszystko, co przychodzi do naszej maszyny.

```
iptables -A INPUT -s 192.168.0.15 --dport 22 -j ACCEPT
```

```
iptables -A OUTPUT -s 192.168.0.15 --dport 22 -j ACCEPT
```

zezwalamy na połączenia przychodzące ssh z maszyny o podanym adresie (192.168.0.15).

```
iptables -I INPUT 1 - p icmp d 127.0.0.1 -j DROP
```

zablokowanie pakietów ICMP dla adresu 127.0.0.1.

```
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

ustanowienie maskarady dla interfejsu ppp0.

```
iptables -t nat -A POSTROUTING -s 10.0.0.0/255.0.0.0 -o ppp0 -j MASQUERADE
```

inna wersja maskarady, czyli translacji adresów NAT umożliwiająca dostęp do ppp0 komputerom z sieci 10.0.0.0/8.

**UWAGA!** Aby polecania maskarady działały poprawnie należy włączyć przekazywanie pakietów w jadrze Linuxa:

*echo ”1” > /proc/sys/net/ipv4/ip_forward*

```
iptables -A PREROUTING -t nat -p tcp -d 83.156.33.14 --dport 8080 -j DNAT -to 192.168.0.1:80
```

próba połączenia sie z maszyną o wskazanym adresie i podanym porcie z maszyny w sieci lokalnej (w której działa NAT) spowoduje przekierowanie na 192.168.0.1:80.

```
iptables -A PREROUTING -t nat -p tcp --dport 8080 -j DNAT --to 192.168.0.1:80
```

to samo, ale dla dowolnego adresu.

```
iptables -t filter -A FORWARD -s 192.168.0.1/255.255.255.0 -d 0/0 -j ACCEPT
iptables -t filter -A FORWARD -s 0/0 -d 192.168.0.1/255.255.255.0 -j ACCEPT
iptables -t filter -A INPUT -j ACCEPT
```

Forwardowanie pakietów dla karty w sieci LAN o adresie 192.168.0.1.

```
iptables -F -t nat
iptables -X -t nat
iptables -F -t filter
```

```
iptables -X -t filter
```

Czyszczenie reguł iptables z tabel nat i filter.

```
iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP
```

Firewall “doskonały”, czyli taki ,który niczego nie wpuszcza ani nie wypuszcza.

**Zadania**. Zakładamy, że komputer posiada dwa interfejsy sieciowe – Internet (np. eth0) oraz interfejs lokalny (np. eth1).

1. Zbuduj firewall, który będzie akceptował jedynie pakiety ICMP. Sprawdź, czy rzeczywiście działa.

   1a. Zmodyfikuj firewall z punktu 1 tak, aby tym razem akceptował wszystko oprócz pakietów ICMP. Sprawdź, czy działa.

   1b. Zmodyfikuj firewall z punktu 1a, tak aby można było używać protokołu ICMP tylko dla interfejsu lo dwiema metodami (za pomocą wskazania konkretnego adresu oraz wskazania interfejsu ).

   1c. Zmodyfikuj firewall z punktu 1, tak aby plecenie ping dla dowolnego adresu kończyło się komunikatem powodzenia z wyjątkiem interfejsu lo.

2. Zbuduj firewall, który będzie blokował połączenia protokołu TCP na porcie 80 (czyli dla usługi WWW), sprawdź dowolną metoda, czy reguła działa. Jaki będzie rezultat działania polecenia ping w takim przypadku dla adresu www.wp.pl?

3. Zbuduj firewall, który będzie odrzucał wszystkie pakiety protokołów TCP, UPD oraz ICMP z wyłączeniem:

- dla protokołu TCP : porty : 80, 20 do 22

- dla protokołu UDP: porty 67 (usługa DHCP), 53 (usługa DNS)

Firewall ponadto powinien blokować wszystkie pakiety UDP z interfejsu lokalnego eth1.

Opracowano na podstawie :
1. http://www.openbsd.org/faq/pf/pl/filter.html
2. http://zsk.wsti.pl/publikacje/iptables_przystepnie.htm

3. http://iptables-tutorial.frozentux.net/iptables-tutorial.html
4. http://www-users.mat.uni.torun.pl/~minaq/iptables.pdf
5. http://grise.top100.net.pl/net/bezpieczenstwo/r3a.htm
6. http://ljopek.kis.p.lodz.pl/ZBS/IPtables1.pdf