

Pakiet Iptables

Mgr inż. Łukasz Jopek
Katedra Informatyki Stosowanej Politechniki Łódzkiej
ljopek@kis.p.lodz.pl

„Filtrowanie pakietów i filtrowanie stanowe”

Filtrowanie pakietów oraz filtrowania stanowe są jedną z podstawowych funkcjonalności firewalli, pozwalają bowiem na decydowanie, które pakiety zostaną przyjęte / przekazane dalej, lub odrzucone. Jednak pojęcie filtrowania pakietów jest starsze od samych firewalli, wcześniej bowiem routery z takim filtrowaniem były pierwszymi urządzeniami stworzonymi z myślą zabezpieczenia naszej sieci przed intruzami. Mechanizm filtrowania sprawdza każdy pakiet sieciowy przesyłany za pośrednictwem danej maszyny, na której zainstalowane jest oprogramowanie służące do filtracji pakietów i zgodnie ze zdefiniowanymi wcześniej regułami – albo go odrzuca, albo przyjmuje, albo przekazuje dalej. Operuje on głównie na podstawie informacji zawartych w nagłówkach pakietów TCP/IP.

1. Czym jest i jak działa filtrowanie pakietów ?

Filtrowanie adresów IP opiera się na sprawdzaniu adresu źródłowego i docelowego w nagłówku pakietu IP i podejmowaniu odpowiednich działań w wypadku odnalezienia pakietu spełniającego pewne zdefiniowane reguły firewalla. Filtrować można również w oparciu o protokoły, można np. blokować pakiety protokołu ICMP odpowiadające za działanie popularnego polecenia *ping*. Choć przydatne w przypadku testowania sieci intruzowi pozwolić może na sprawdzenie, jakie adresy wykorzystywane są w danej sieci. W przypadku protokołów TCP i UDP można blokować odpowiednie porty. Pozwala to np. zablokować użytkownikom sieci na korzystanie z pewnych usług (np. WWW, ftp czy innej aplikacji).

2. Czym jest i jak działa filtrowanie stanów ?

iptables wykorzystuje także filtrowanie stanowe, które jest podobnie do filtrowania pakietów, ale dodatkowo śledzi i analizuje kontekst komunikacji. Firewall tego typu utrzymują tablice z informacjami na temat aktualnych połączeń. Filtrowanie pakietów odbywa się w warstwie

sieci, a filtrowanie stanowe następuje w warstwie wyższej. Dlatego też jest w stanie wykryć i powstrzymać bardziej formy wyszukane ataków za pomocą protokołów wyższego poziomu, np. TCP czy UDP.

Filtrowanie pakietów nie jest jednak tylko funkcjonalnością firewalli, pełni 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 sieć używa NAT. W niej ustalamy maskaradę adresów sieciowych i przekierowujemy pakiety.

Tabela **filter** służy do właściwego filtrowania pakietów. Tutaj definiowane są podstawowe reguły firewalla, ustalone są reguły, Dzięki za wszystko którym firewall decyduje, czy dany pakiet zaakceptować czy odrzucić.

Łańcuchy :

INPUT - wywoływany dla pakietów przybywających z sieci przeznaczonych dla lokalnej

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

FORWARD - wywoływany dla pakietów routowanych przez lokalną maszynę, lecz pochodzących dla niej.

PREROUTING - wywoływany dla pakietów z zewnątrz jeszcze przed ich routowaniem.

POSTROUTING - wywoływany dla pakietów, które właśnie opuszczają maszynę

Różne tabele iptables dysponują różnymi wbudowanymi łańcuchami. Np. tabela **filter** zawiera łańcuchy `INPUT`, `FORWARD` i `OUTPUT`. Tabela **nat** zawiera łańcuchy

PREROUTING, OUTPUT i POSTROUTING, ale już tabela **mangle** zawiera wszystkie rodzaje łańcuchów.

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

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

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. Jednak każdy następny pakiet już tak potraktowany nie zostanie.

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

RELATED (powiązany) - Służą do obsługi innego obecnie istniejącego połączenia, np. takiego, które są 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, nie jest jednak automatycznie odrzucany, ale jest ciągle aktywny. Można go wykorzystać używając innych zasad, np. dynamicznie ustawić taktykę łańcuchową.

Schemat budowy reguły iptables wygląda następująco:

iptables [-t table] command [match] [-j jump/target]

gdzie:

Table – umożliwia wybór tablicy, dla której obowiązywać będzie dana reguła, brak tej informacji w regule powoduje zapisanie jej do tabeli *filter*.

Command - (wybieramy jedną; z opcją modprobe ładujemy regułę do jądra):

- **iptables -t tabela -A łańcuch opis_reguły**

Dodaje regułę na koniec wskazanego łańcucha

- **iptables -t tabela -D łańcuch opis_reguły**

Usuwa zadaną regułę z łańcucha (trzeba podać cały łańcuch, który chcemy usunąć).

- **iptables -t tabela -D łańcuch numer_reguły**

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

- **iptables -t tabela -I łańcuch numer_reguły opis_reguły**

Dodaje regułę we wskazanym miejscu łańcucha. Jeśli łańcucha (reguły numerowane są w zbiorach względem wierszy – od góry do dołu, począwszy od 1).

- **iptables -t tabela -R łańcuch numer_reguły opis_reguły**

Zamienia regułę wskazaną numerem na opisaną w poleceniu.

- **iptables -t tabela -L łańcuch**

Listuje reguły we wskazanym łańcuchu. Pominięcie nazwy

- **iptables -t tabela -N łańcuch**

Tworzy łańcuch użytkownika o zadanej nazwie.

- **iptables -t tabela -X łańcuch**

Usuwa łańcuch użytkownika. Warunkiem jest brak odwołań

- **iptables -t tabela -P łańcuch domyślny_cel**

Ustawia policy (domyślną akcję) zadanego łańcucha

match (pominięcie opcji ustalającej “X” oznacza ustalenie reguły dla wszystkich “X”):

- `-p [!] protokół` lub `-- protocol [!] protokół`

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

- `-S [!] adresIP , src [!] adresIP , source [!] adresIP`

ustala regułę dla pakietów o adresie źródłowym adresIP, 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 [!] adresIP ,dst [!] adresIP ,destination [!]adresIP`

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

- `-i [!] interfejs ,ininterface [!] interfejs`

ustala regułę dla danego interfejsu (np. eth0,eth1,ppp0) do którego pakiet przychodzi, 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 ,outinterface [!] interfejs`

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

- `-Sport [!] nr ,sourceport [!] nr` (tylko dla protokołów TCP, UDP)

ustala regułę dla pakietów o numerze źródłowym portu nr. Możemy zamiast nr 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)

- `dport [!] nr ,destinationport [!] nr` (dla protokołów TCP, UDP)

ustala regułę dla pakietów o numerze docelowym portu nr. Zasady jak wyżej.

jump/target

Akcję definiuje się, używając parametru **-j**. Najczęściej używane akcje to:

- **ACCEPT** - przyjmij pakiet.
- **DROP** - wyrzuć 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 użycie 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

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! Jednak aby polecenia maskarady działały poprawnie należy włączyć przekazywanie pakietów w jądrze 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 się 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

```
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 eth1, który jest interfejsem lokalnym.)

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 polecenie *ping* dla dowolnego adresu kończyło się komunikatem się powodzeniem z wyjątkiem interfejsu *lo*.
2. Zbuduj firewall, który będzie blokował połączenia protokołu TCP na porcie 80 (czyli dla usługi *WWW*), sprawdź dowolną metodą, czy reguła działa. Jaki będzie rezultat działania polecenia *ping* w takim przypadku dla adresu [WWW.wp.pl](http://www.wp.pl) ?
3. Zbuduj firewall, który będzie odrzucał wszystkie pakiety protokołów TCP, UDP oraz ICMP z włączeniem :
 - 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.

- 3a. Zmodyfikuj firewall z punktu 3, tak aby blokował także połączenia ICMP.

Połączenia na protokole TCP można testować za pomocą polecenia *wget*

Przykładowy adres do pliku (zawierającego rpm 'iptables') :

<ftp://ftp.pbone.net/mirror/ftp.pld-linux.org/dists/2.0/test/i586/iptables-1.4.0-5.i586.rpm>

(tylko dla trybu aktywnego).

lub za pomocą programu *lynx* (tekstowa przeglądarka stron www).

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>