# iptables

## Historia

Prace nad projektem ***iptables*** rozpoczął w 1998 roku australijczyk Rusty Russell, autor wcześniejszego projektu ***ipchains***. ***ipchains*** jak i wcześniejszy ***ipfwadm*** (wywodzący się z BSD-owskiego ***ipfw***) zmieniały kod sieciowy, umożliwiając manipulację pakietami. Brakowało jednak ogólnego szkieletu służącego zarządzaniu pakietami. Dopiero ***iptables*** łącząc główne idee poprzedników stał się potężnym narzędziem umożliwiającym manipulację pakietami. Obecnie ***iptables*** jest częścią projektu ***netfilter*** pisanego i udostępnianego na licencji *GNU General Public License (GPL)* przez *netfilter core team.*

## Infomacje ogólne

***iptables*** jest narzędziem (tzw. *user space tool)* pozwalającym administratorowi systemu na zarządzaniu regułami filtrowania, przekazywania oraz maskowania pakietów. Za bezpośrednią manipulację pakietami odpowiada infrastruktura odpowiednich mechanizmów zawarta w jądrze systemu. ***netfilter*** został włączony do jądra Linuxa 2.3.

## Zasada działania

Podczas gdy ***ipchains*** jak i ***ipfwadm*** łączyły w sobie filtrowanie pakietów i NAT, ***netfilter/iptables*** pozwala na rozłączenie tych działań na trzy części: filtrowanie pakietów, śledzenie połączenia i translację adresów (NAT). Takie rozwiązanie pozwoliło na uzyskanie informacji o pakiecie już w warstwie połączenia. ***Iptables*** umożliwia przekierowywanie, modyfikowanie, zatrzymywanie pakietów na podstawie stanu połączenia a nie tylko adresów (źródłowego i docelowego) oraz zawartości samego pakietu, jak to było w ***ipchains.*** Zapora ogniowa zbudowana na bazie ***iptables*** nosi angielską nazwę *stateful firewall*; oparta na ***ipchains*** to *stateless firewall.*

## Reguła, łańcuch, tabela

W celu zbadania danego pakietu ***netfilter/iptables*** wykorzystuje tzw. reguły. Jest to najmniejsza jednostka filtra *netfilter*. Zawiera ona zbiór warunków, jakie musi spełniać pakiet oraz akcję, która zostanie wykonana gdy warunki zostaną spełnione. Pojedyncze reguły grupowane są w łańcuchy. Zbiór łańcuchów natomiast tworzy tabelę.

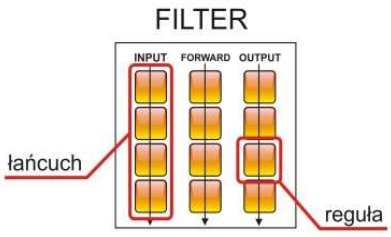
Istnieją trzy tabele:

1. ***filter*** domyślna tabela; filtrowanie pakietów
2. ***nat*** tabela używana przez pakiety nawiązujące połączenie; translacja adresów
3. ***mangle*** tabela służąca do modyfikacji przepływających pakietów

Każda z tabel zawiera kilka predefiniowanych łańcuchów oraz łańcuchy zdefiniowane przez administratora. Istnieją następujące predefiniowane łańcuchy:

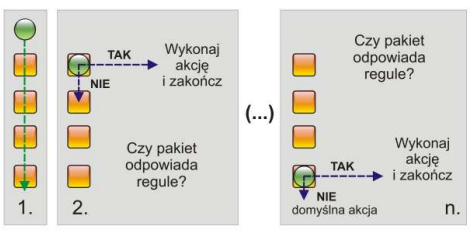
- **INPUT** - wywoływany dla pakietów przybywających z sieci przeznaczonych dla lokalnej maszyny.
- **FORWARD** - wywoływany dla pakietów routowanych przez lokalną maszynę, lecz pochodzących spoza niej i nie przeznaczonych dla niej.
- **OUTPUT** - wywoływany dla pakietów tworzonych lokalnie i wychodzących poza maszynę.
- **PREROUTING** - wywoływany dla pakietów z zewnątrz jeszcze przed ich routowaniem.
- **POSTROUTING** - wywoływany dla pakietów, które właśnie opuszczają maszynę

Różne tabele dysponują różnymi wbudowanymi łańcuchami. Tabela **filter** zawiera łańcuchy INPUT, FORWARD i OUTPUT. Waro zauważyć, że (w przeciwieństwie do występującego w kernelach 2.2 ipchains) pakiet może znaleźć się w tylko jednym z tych trzech łańcuchów. Tabela **nat** zawiera łańcuchy PREROUTING, OUTPUT i POSTROUTING. Natomiast tabela **mangle** zawiera wszystkie rodzaje wbudowanych łańcuchów.




Rys. Tabela ***filter***

Każdy pakiet rozpoczyna swoją podróż przez filtr pakietowy w jednym z predefiniowanych łańcuchów. Łańcuch skanowany jest od góry do dołu w poszukiwaniu reguł, które będą odpowiadać sprawdzanemu pakietowi. Po napotkaniu takiej reguły, wykonywana jest jej akcja. Pozostała część łańcucha nie jest już skanowana. Wyjątkiem jest przypadek, gdy wywołany zostanie łańcuch zdefiniowany przez administratora zakończony akcją RETURN. Skanowanie rozpocznie się wtedy od kolejnej reguły od tej, która wywołała przeskok do łańcucha administratora. Jeśli pakiet opuszcza łańcuch bez wykonania żadnej akcji, wykonywana jest akcja domyślna (chain policy).




Rys. Przepływ pakietu przez łańcuch

## Konfiguracja reguł

Do zarządzania regułami służy polecenie iptables. Można je wywołać na następujące sposoby:

- **iptables -t tabela -A łańcuch opis_reguły**
  Dodaje regułę na koniec wskazanego łańcucha we wskazanej tabeli. Parametr **-t** można pominąć, zostanie wtedy użyta tabela filter.
- **iptables -t tabela -D łańcuch opis_reguły**
  Usuwa zadaną regułę z łańcucha.
- **iptables -t tabela -I łańcuch numer_reguły opis_reguły**
  Dodaje regułę we wskazanym miejscu łańcucha. Jeśli pominie się numer_reguły, reguła zostanie wstawiona na początku łańcucha.
- **iptables -t tabela -R łańcuch numer_reguły opis_reguły**
  Zamienia regułę wskazaną numerem na opisaną w poleceniu.
- **iptables -t tabela -D łańcuch numer_reguły**
  Usuwa regułę o podanym numerze.
- **iptables -t tabela -L łańcuch**
  Listuje reguły we wskazanym łańcuchu. Pominięcie nazwy łańcucha spowoduje wylistowanie całej zawartości tabeli.
- **iptables -t tabela -N łańcuch**
  Tworzy łańcuch użytkownika o zadanej nazwie.
- **iptables -t tabela -X łańcuch**
  Usuwa łańcuch użytkownika. Warunkiem jest brak odwołań do wskazanego łańcucha w innych łańcuchach.
- **iptables -t tabela -P łańcuch domyślny_cel**
  Ustawia policy (domyślną akcję) zadanego łańcucha

Na definicję reguły składa się dowolna ilość warunków oraz dokładnie jedna akcja. W przypadku braku warunków akcja wykonywana jest zawsze, oczywiście tylko wtedy, gdy wykonanie łańcucha dojdzie do takiej reguły. Natomiast reguła posiadająca tylko warunki, a nie posiadająca akcji, będzie używana wyłącznie do liczenia pakietów pasujących do warunków.

Najczęściej używane warunki to:

- **-p protokół** - protokół, do którego należy pakiet. Może być to tcp, udp, icmp lub all (wszystkie). Może być również użyta wartość numeryczna. Wykrzyknik przed nazwą protokołu odwraca znaczenie warunku.
- **-s adres_źródłowy/maska** - adres źródłowy pakietu. Maska może być zapisana tradycyjnie - w postaci czterech liczb oddzielonych kropkami, np. 255.255.255.0 - bądź jako liczbę oznaczającą ilość bitów ustawionych po lewej stronie maski. W przypadku nie podania maski, przyjmowana jest wartość domyślna /32 - akceptowany jest więc tylko zadany adres. Wykrzyknik przed adresem odwraca znaczenie warunku.
- **-d adres_źródłowy/maska** - adres docelowy pakietu. Zasady zapisu adresu takie same, jak dla adresu źródłowego.
- **-i interfejs** - interfejs, z którego pakiet został przyjęty. Można użyć wykrzyknika dla odwrócenia znaczenia warunku.
- **-o interfejs** - interfejs, przez który pakiet zostanie wysłany.
- **--source-port port** - port źródłowy lub zakres portów wg. schematu port_minimalny:port_maksymalny. Użyteczne jedynie z -p tcp lub udp.
- **--destination-port port** - port docelowy lub zakres portów wg. schematu port_minimalny:port_maksymalny. Użyteczne jedynie z -p tcp lub udp.

Istnieje również zestaw warunków opcjonalnych, ładowanych przez opcję -m. Pozwalają one na uzyskiwanie ciekawych efektów, jak na przykład dopasowywanie określonej ilości pakietów na sekundę. Dokładne informacje o nich znajdują się w manualu.

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

- **ACCEPT** - przyjmij pakiet.
- **DROP** - wyrzuć pakiet 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** - oznaczenie pakietu za pomocą wartości numerycznej podanej w parametrze --set-mark. Użyteczne w tabeli mangle. Używane zwykle dla przekazywania informacji do iproute2.
- **MASQUERADE** - maskuje adres nadawcy pakietu, użyteczne wyłącznie w łańcuchu POSTROUTING tabeli nat. Powinno się używać jedynie przy dynamicznie przydzielanym adresie IP.
- **SNAT** - zmienia adres źródłowy pakietów i zapamiętuje tą zmianę dla danego połączenia. Używane głównie do podłączania sieci LAN do Internetu przez łącze ze stałym adresem IP. Użyteczne wyłącznie w łańcuchu POSTROUTING tabeli nat. Akceptuje parametr --to-source, wskazujący 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. Opcjonalny argument --to-ports pozwala na przekierowanie połączenia na dowolnie wybrany port lokalnej maszyny.
- **DNAT** - zmienia adres docelowy pakietów i zapamiętuje zmianę dla danego połączenia. Używane głównie do przekazywania połączeń z interfejsu internetowego do maszyn w sieci lokalnej. Użyteczne w łańcuchach PREROUTING i OUTPUT tabeli NAT. Akceptuje parametr --to-destination, określający adres z opcjonalnym portem docelowym (wg formatu adres:port).

## Instalacja i konfiguracja

W związku z tym, że ***netfilter*** jest częścią jądra musimy upewnić się, że nasze jądro zawiera odpowiednie elementy:

```
[root@localhost:/usr/src/linux]# make menuconfig
```

Wybieramy gałąź:
*Device drivers -> Networking support -> Networking options -> Network packet filtering*
oraz odpowiednie elementy *IP: netfilter configuration* według potrzeb. Następnie kompilujemy jądro na przykład tak:

```
[root@localhost:/usr/src/linux]# make ; make modules_install
```

Teraz wystarczy skopiować nowy obraz jądra do /boot i zrestartować komputer. Nasza maszyna jest przygotowana (prawie) do filtrowania pakietów. Ostatnim krokiem będzie przygotowanie narzędzia, za pomocą którego powiemy systemowi jak postępować z pakietami.

Pobieramy plik z: *http://www.netfilter.org/projects/iptables/files/iptables-1.3.4.tar.bz2*

Rozpakowujemy go na przykład tak:

```
[user@localhost:~/download]$ tar xjvf iptables-1.3.4.tar.bz2
```

Kompilujemy tak:

```
[user@localhost:~/download/iptables-1.3.4]$ make KERNEL_DIR=/usr/src/linux
```

Instalujemy tak:

```
[root@localhost:/home/user/download/iptables-1.3.4]# make install KERNEL_DIR=/usr/src/linux
```

Jeśli wszystko przebiegło poprawnie, to mamy system z działającym iptables. Jeśli nie, szukamy przyczyn. ;) Przydatny adres:

*http://www.netfilter.org/projects/iptables/index.html*

## Przykłady:

- ***iptables*** -P INPUT -j DROP
  *blokujemy wszystko, co do nas dochodzi*
- ***iptables*** -A INPUT -s 0.0.0.0/0 --dport 80 -j ACCEPT
  *zezwalamy na ruch przychodzący z dowonego miejsca na nasz port :80*
- ***iptables*** -A INPUT -s 192.168.0.15 --dport 22 -j ACCEPT
  *zezwalamy na połączenia ssh z maszyny o podanym adresie*
- ***iptables*** -A INPUT -s 158.75.2.7 -j ACCEPT
  *zezwalamy na połączenia od waldemara (DNS)*
- ***iptables*** -A INPUT -p TCP -s ! 10.0.0.1 --syn -j DENY
  *blokujemy dostęp do naszego komputera dla wszystkich adresów różnych od podanego, które mają ustawioną flagę SYN*
- ***iptables*** -A INPUT -p icmp -j ACCEPT
  *zaczynamy odpowiadać na ping (dotąd blokowała to pierwsza reguła)*
- ***iptables*** -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
  *przyjmujemy tylko odpowiedzi na nasze zapytania ping (usuń regułę powyżej, aby ta miała sens)*
- ***iptables*** -A INPUT -p tcp --dport 20:21 -j ACCEPT
  *umożliwiamy korzystanie z portów 20, 21 przez klientów ftp (tylko passive mode)*
- ***iptables*** -A INPUT -m state --state RELATED -j ACCEPT
  *umożliwia korzystanie z klientów ftp w trybie aktywnym (potrzebne moduły: ip_conntrack i ip_conntrack_ftp)*
- ***iptables*** -t nat -A POSTROUTING -s 10.0.0.0/255.0.0.0 -o ppp0 -j MASQUERADE
  *prosta "maskarada" czyli translacja adresów NAT umożliwiająca dostęp do ppp0 komputerom z sieci 10.0.0.0/8. UWAGA! Aby powyższe polecenie miało sens należy włączyć przekazywanie pakietów w jądrze linuxa:*
  ```
  [root@localhost:~]# 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 10.0.0.1:80
  *próba połączenia się z maszyną o wskazanym adresie i podanym porcie z maszyny w sieci lokalnej (w której działa NAT) spowoduje przekierowanie na 10.0.0.1:80*
- ***iptables*** -A OUTPUT -t nat -p tcp -d 83.156.33.14 --dport 8080 -j DNAT --to 10.0.0.1:80
  *próba połączenia się z maszyną o wskazanym adresie i podanym porcie z komputera lokalnego spowoduje przekierowanie na 10.0.0.1:80*
- ***iptables*** -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3128
  *ustawienie używane przy tzw. transparent proxy, prekierowuje ruch przychodzący z eth0 z docelowym portem 80 na port 3128, gdzie słucha transparent proxy*

## Przydatne narzędzia:

- netstat — wyświetla stan połączeń, otwarte porty i nie tylko
- lsof — **Li**s**t** **O**pen **F**iles; wyświetla otwarte pliki
- ifconfig — wyświetla/zmienia konfigurację interfejsów
- iptraf — monitor LAN w ncurses

## Literatura:

- man iptables
- Strona projektu netfilter <http://www.netfilter.org/documentation/index.html#documentation-other> i odnośniki