Sieci komputerowe

Tadeusz Kobus, Maciej Kokociński
Instytut Informatyki, Politechnika Poznańska

Filtracja pakietów w Linuksie

Netfilter – część jądra systemu operacyjnego Linux pozwalająca na kontrolę ruchu sieciowego.

Do kontroli mechanizmów netfilter służą następujące programy:

- `iptables`
- `ip6tables`
- `ebtables`
- `arptables`

Projekty związane z mechanizmem filtracji pakietów w Linuksie funkcjonują w ramach *Netfilter project*.

Proces filtracji pakietu (1)

Na różnych etapach przechodzenia pakietu przez jądro Linuksa, aplikowane są kolejne filtry.

Filtry pogrupowane są w tabele (tables) grupujące różne funkcjonalności:

- `filter` – odrzuca niechciane pakiety,
- `nat` – zmienia adresy źródłowe lub docelowe pakietu
- `mangle` – modyfikuje pakiety,
- `raw` – udostępnia pakiety przed ich przetworzeniem przez część kernela (np. w trybie śledzenia połączeń (conntrack)),
- `security` – używana przy zaawansowanych modelach bezpieczeństwa w Linuksie (np. SELinux).

Proces filtracji pakietu (2)

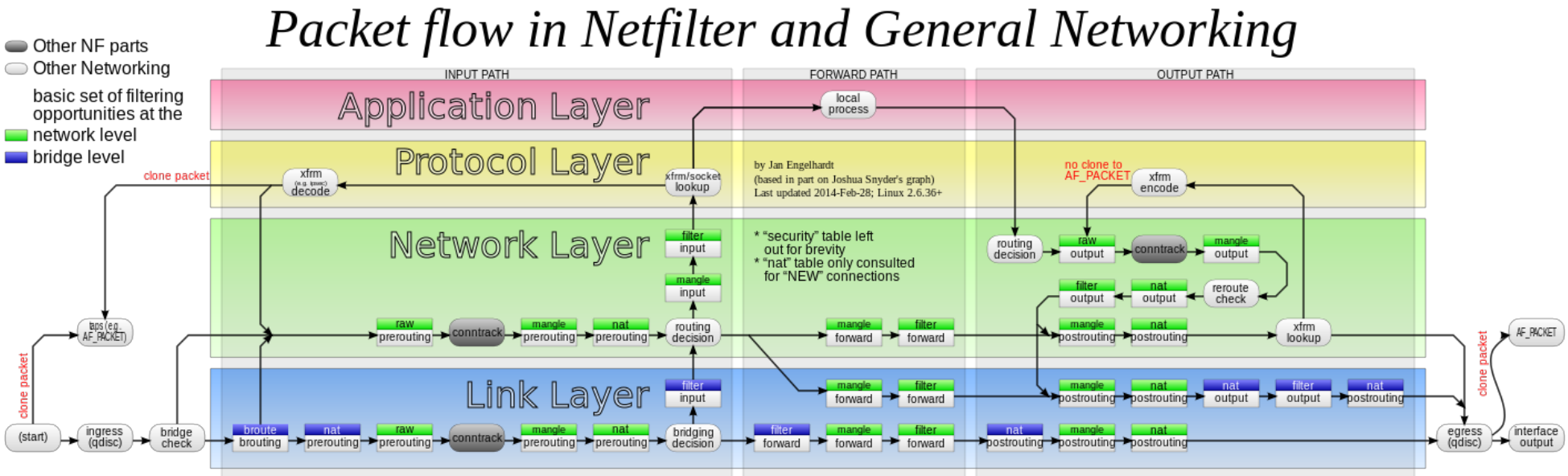
Każda z tablic ma wbudowany zestaw łańcuchów reguł odpowiadających etapom ich aktywacji:

- `filter`: `INPUT`, `FORWARD`, `OUTPUT`,
- `nat`: `PREROUTING`, `OUTPUT`, `POSTROUTING`,
- `mangle`: `PREROUTING`, `INPUT`, `FORWARD`, `OUTPUT`, `POSTROUTING`,
- `raw`: `PREROUTING`, `OUTPUT`.

Można tworzyć własne łańcuchy, ale mogą one być uruchomione tylko przez dołączenie ich do już istniejących.

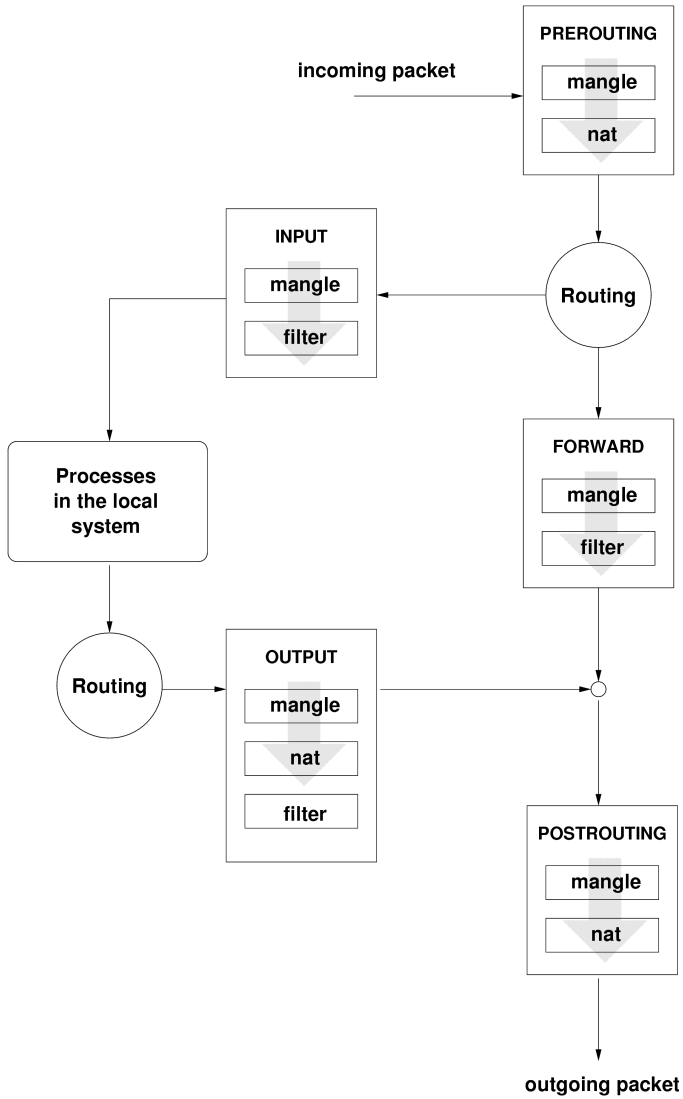
Jądro Linuksa jest zaopatrzone w mechanizm śledzenia połączeń (conntrack). Korzystanie z tego mechanizmu często znacznie ułatwia konfigurację filtracji pakietów.

Proces filtracji pakietu (3)



JK: I tak ten rysunek zawiera nieścisłości.

Proces filtracji pakietu (4)



Konfiguracja filtrów (1)

Reguły są przetwarzane w kolejności – jeśli któraś reguła zadecyduje o losie pakietu, późniejsze nie są brane pod uwagę.

Komenda `iptables` wszędzie rozróżnia wielkie i małe litery. Kolejność argumentów jest istotna!

Typowe wywołanie komendy:

```
iptables [-t tablica] [opcja + łańcuch
+ parametry pakietu] [-j akcja]
```

Operacje w ramach akcji:

- `ACCEPT` – przepuszczenie pakietu,
- `DROP` – ignorowanie pakietu,
- `REJECT` – odrzucenie pakietu (symuluje zamknięte gniazdo),
- inne akcje, np. `LOG`, `MARK`, `SET`.

Konfiguracja filtrów (2)

Wypisywanie reguł:

- `iptables [-t filter] -L`
- `iptables -t <tablica> -L`
- `iptables [-t <tablica>] -L -n -v -x --line-numbers`

Zmiana domyślnego zachowania (tj. jeśli żadna inna reguła nie określi działania):

- `iptables [-t filter] -P <łańcuch> <polityka>`
- zwykle stosowane polityki to ACCEPT lub DROP

```
# iptables -P INPUT DROP
```

Konfiguracja filtrów (4)

Modyfikacja reguł:

- `iptables -A <łańcuch> <reguła>`
  dodawanie (`--append`) reguły do łańcucha,
- `iptables -I <łańcuch> [pozycja] <reguła>`
  wstawianie (`--insert`) reguły do łańcucha na daną pozycję,
- `iptables -D <łańcuch> <reguła>/<nr>`
  usuwanie (`--delete`) reguły z łańcucha z podanej pozycji,
- `iptables -F [łańcuch]`
  usuwanie wszystkich reguł [z łańcucha] (`--flush`),
- `iptables -Z [łańcuch]`
  resetowanie (`--zero`) liczników łańcucha.

```
# iptables -A INPUT -s 1.2.3.4 -j ACCEPT
# iptables -I INPUT 1 -s 1.2.3.4 -j ACCEPT
# iptables -D INPUT 2
# iptables -D INPUT -s 1.2.3.4 -j ACCEPT
# iptables -F INPUT
```

Konfiguracja filtrów (5)

Rdzeń iptables zawiera niewiele filtrów, m.in. źródłowy i docelowy adres (`-s`, `-d`), interface (`-i`, `-o`) i protokół (`-p`). Wiecej: `man iptables`.

Filtry można negować przez użycie znaku wykrzyknika '!'.

Komenda iptables ma rozbudowaną pomoc wewnętrzną – po napotkaniu `--help` przerwie dodawanie reguły i pokaże listę opcji.

```
# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
# iptables -A INPUT -i wlan0 -s 192.168.55.0/24 -j REJECT
# iptables -I INPUT -p tcp --help
```

Konfiguracja filtrów (6)

Wiele funkcji dostępnych jest poprzez rozszerzenia, ładowane przez `-m <nazwa>` (patrz też `man iptables-extensions`):

- `-m tcp/udp` – jest automatycznie ładowany razem z `-p udp/tcp`; pozwala ustalić m.in. port źródłowy i docelowy `--sport/--dport`,
- `-m conntrack` – wybiera stan połączenia `--ctstate`, m.in.: `INVALID`, `NEW`, `ESTABLISHED`, `RELATED`,
- `-m comment` – pozwala na dowolny komentarz `--comment`,
- `-m limit` – dzięki `--limit` ogranicza liczbę pakietów na jednostkę czasu,
- `-m time` – pozwala włączyć regułę o `--datestart` i wyłączyć o `--datestop`,
- `-m connlimit` – używając `--connlimit-above` pozwala ograniczyć liczbę połączeń z jednego adresu.

Konfiguracja filtrów (7)

```
(ruch wychodzacy dla grupy o gid 1006 tylko do sieci 192.168.1.0/24)
# iptables -A OUTPUT -m owner --gid-owner 1006 ! -d 192.168.1.0/24 -j DROP

(ignoruj przychodzace polaczenia tcp)
# iptables -A INPUT --protocol tcp --syn -j DROP
# iptables -A INPUT --protocol tcp --tcp-flags SYN,RST,ACK,FIN SYN -j DROP

(tylko 2 polaczenia telnetowe per klient)
# iptables -A INPUT -p tcp --syn --dport 23 -m connlimit \
                                          --connlimit-above 2 -j REJECT

(dopusc ruch tylko z okreslonego adresu mac)
# iptables -A INPUT -m mac ! --mac-source 00:12:34:56:78:ab -j DROP

(umozliwienie przyjmowania komunikatow icmp)
# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

Zadanie 1

1. Zablokuj cały ruch wchodzący. Użyj do tego polityki, nie reguły.
2. Pozwól wchodzić połączeniom już ustanowionym.
3. Pozwól wchodzić połączenim ssh.
4. Zabroń wchodzenia ruchu ssh z wybranego adresu nie kasując żadnej z powyższych reguł.
5. Wyświetl liczniki, sprawdź ile razy reguła z poprzedniego zadania została uruchomiona.
6. Zbadaj (używając wiresharka) czym różni się cel DROP od REJECT.
7. Zablokuj ruch wychodzący do `www.cs.put.poznan.pl`
8. Ogranicz liczbę połączeń ssh od każdego adresu IP z osobna.
9. Ogranicz globalną liczbę połączeń ssh.