

# Instalacja i konfiguracja pakietu iptables

Tomasz Nowocień

Zespół Bezpieczeństwa PCSS

[security@man.poznan.pl](mailto:security@man.poznan.pl)

# Zawartość

Czyli o czym to będzie...

- Podstawy wiedzy...
- Co to jest iptables?  
Skąd się bierze iptables?
- Podstawy konfiguracji iptables.
- Przykłady konfiguracji iptables.

# Podstawy wiedzy

Czyli co już wiemy...

- Co to jest sieć komputerowa?
  - adres IP? maska sieci? porty?
  - firewall? DNS?
  - *routing*?
  - warstwowy model sieci?
  - Protokoły? TCP? IP? UDP? ICMP?

# Co to jest iptables?

- filtr pakietów działający na warstwie TCP/IP
- zaimplementowany w jądrze Linuxa 2.4.X oraz 2.6.X
- aktualna wersja 1.3.5 (26.06.2006r.)

# Skąd pobrać?

- <http://netfilter.filewatcher.org/>
- <http://netfilter.samba.org/>
- <http://netfilter.gnumonks.org/>

# Podstawy konfiguracji

- Łańcuchy iptables
- Reguły iptables
- Polityki iptables

Podstawy konfiguracji

# Łańcuchy iptables

- Predefiniowane:
  - INPUT
  - OUTPUT
  - FORWARD
  - (PREROUTING)
  - (POSTROUTING)
- Łańcuchy definiowane przez użytkownika

# Zasady tworzenia reguł

(1)

- Zasady tworzenia reguł są proste  
:]

```
iptables -A FORWARD -i eth1  
-s 0/0 -d 0/0 -p TCP --sport  
! 80 --dport ANY -j ACCEPT
```

# Zasady tworzenia reguł

(2)

- Oznaczenia:
  - s – adres źródłowy
  - d – adres docelowy
  - i – interfejs wejściowy
  - o – interfejs wyjściowy
  - j – wykonywana akcja
  - p – protokół

# Zasady tworzenia reguł

(3)

- Operacje na pojedynczych regułach:
  - A – dodawanie
  - D - kasowanie
  - I - wstawianie
  - R – zamiana

# Zasady tworzenia reguł

(4)

- Operacje na łańcuchach
  - N –tworzenie nowego łańcucha
  - F –opróżnianie łańcucha
  - Z –zerowanie liczników dla łańcucha
  - P –tworzenie polityki dla łańcucha
  - X –kasowanie łańcucha
  - L –przeglądanie reguł łańcucha

# Proste reguły iptables

(1)

- Najprostszy firewall:

```
iptables -P INPUT DROP
```

**Czy zadziała?**

# Proste reguły iptables

(2)

„Rozbudowany” firewall:

```
iptables -P INPUT DROP
```

```
iptables -A INPUT -p tcp --sport 80
```

```
-j ACCEPT
```

# Rozszerzenia iptables

- używanie rozszerzeń „jawne”:
  - m xxxx
- „niejawne”

## Rozszerzenia iptables

# Rozszerzenia tcp

- „niejawne”
  - ładowane podczas pojawienia się
    - p tcp
- opcje:
  - tcp-flags
  - syn
  - source-port (--sport)
  - destination-port (--dport)

Rozszerzenia iptables

# Rozszerzenia udp

- „niejawne”
  - ładowane podczas pojawienia się
    - p udp
- opcje:
  - source-port (--sport)
  - destination-port (--dport)

Rozszerzenia iptables

# Testy stanów

– m state

**Testowane stany:**

- **NEW**
- **ESTABLISHED**
- **RELATED**
- **INVALID**

# Testy stanów

przykład

Ulepszenie „naszego” firewalla:

```
iptables -P INPUT DROP
```

```
iptables -A INPUT -m -state
```

```
ESTABLISHED,RELATED -j ACCEPT
```

# Testy stanów

## przykład

Nie pozwalamy żeby w sieci podłączonej do interfejsu eth1 stawiano serwery:

```
iptables -A FORWARD -i eth1  
-p tcp -m state ! --state NEW  
-j DROP
```

Rozszerzenia iptables

# Filtrowanie po adresie MAC

- m mac

```
iptables -A INPUT -i eth1 -m mac --mac-  
source 00:00:00:00:00:00 -j DROP
```

Uwagi: Działa w łańcuchach:

- INPUT
- PREROUTING

# Informacje o użytkowniku

- m owner
  - uid-owner
  - gid-owner
  - pid-owner
  - sid-owner

- Uwagi: Działa tylko lokalnie w łańcuchu OUTPUT

Rozszerzenia iptables

# Limity

- Limity połączeń określane są modułem:
  - m limit
  - limit - ilość pozytywnych testów w jednostce czasu
  - limit-burst - maksymalna seria, po której określony limit się włącza

## Rozszerzenia iptables

# Limity - przykłady

```
iptables -A FORWARD -p tcp -syn  
-m limit --limit 1/s -j ACCEPT
```

```
iptables -A FORWARD -p icmp  
--icmp-type echo-request  
-m limit --limit 1/s -j ACCEPT
```

# Cele reguł

- Predefiniowane:
  - DROP
  - REJECT
  - ACCEPT
  - RETURN
  - QUEUE
  - LOG
- Łańcuch użytkownika

Cele reguł

# Odrzucanie połączeń

(1)

- Odrzucanie połączeń:
  - REJECT
  - DROP
  
- Przykłady:
  - `iptables -P INPUT DROP`
  - `iptables -P INPUT REJECT`

Cele reguł

# Odrzucanie połączeń

(2)

- Reject:

```
iptables -A INPUT -p tcp -i eth1  
-j REJECT --reject-with tcp-reset
```

```
iptables -A INPUT -p udp -i eth1  
-j REJECT --reject-with icmp-port-  
unreachable
```

Cele reguł

# Łańcuchy użytkownika

## (1)

```
iptables -N test1
```

```
iptables -A test1 -m state
```

```
  --state ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -A test1 -m state
```

```
  --state NEW -i ! ppp0 -j ACCEPT
```

```
iptables -A test1 -j DROP
```

Cele reguł

# Łańcuchy użytkownika

(2)

```
iptables -A INPUT -j test1
```

```
iptables -A FORWARD -j test1
```

# Przykłady reguł FORWARD

(1)

```
iptables -A FORWARD -p tcp -i eth0  
-o eth1 -s any -d 150.254.xxx.xxx  
-dport ! 80 -j REJECT --reject-with  
tcp-reset
```

# Przykłady reguł FORWARD

(2)

```
iptables -A FORWARD -p tcp -i eth0  
-o eth1 -s ! 1.2.3.0/24  
-d 150.254.xxx.xxx  
-j REJECT --reject-with tcp-reset
```

Cele reguł

# Logowanie

```
iptables -A FORWARD -p tcp -j LOG  
--log-level X --log-prefix 'my_log'
```

# Tworzenie firewalla

- Stosuj zasadę: „co nie jest dozwolone, jest zabronione”
  - `iptables -P INPUT DROP`
- Rozpoczynaj budowę firewalla od początku
- Otwieraj dostęp tylko do tych usług, które są konieczne
- Zapisz najczęściej występujące dopasowania na początku

# Przykładowy łańcuch

(1)

```
IIP="150.254.170.3"
```

```
INET="150.254.170.0/20"
```

```
INET_IFACE="eth1"
```

```
MY_NET="10.0.0.254"
```

```
MY_NET="10.0.0.0/24"
```

```
MY_IFACE="eth0"
```

```
# CZYSZCZENIE
```

```
iptables -F INPUT
```

```
iptables -F OUTPUT
```

```
iptables -F FORWARD
```

# Przykładowy łańcuch

## (2)

```
# POLITYKI DOMYSLNE
```

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD DROP
```

```
# Zapobieganie spoofowaniu
```

```
iptables -A OUTPUT -o $INET_IFACE -d $MY_NET  
-j DROP
```

```
iptables -A OUTPUT -o $INET_IFACE -s $MY_NET  
-j DROP
```

# Przykładowy łańcuch

## (3)

```
#Pozbycie się pakietow blednych
```

```
iptables -A INPUT -p tcp ! --syn -m state  
  --state NEW -j DROP
```

```
iptables -A OUTPUT -p tcp ! --syn -m state  
  --state NEW -j DROP
```

```
#Lokalny ruch odblokowany
```

```
iptables -A INPUT -i lo -j ACCEPT
```

```
#Pozwalamy na ruch ICMP, poza „timestamp-request”
```

```
iptables -A INPUT -p icmp --icmp-type timestamp-request  
  -j DROP
```

```
iptables -A INPUT -p icmp -j ACCEPT
```

# Przykładowy łańcuch

## (4)

#Pozbywamy się ruchu na określonych portach

```
iptables -A INPUT -p tcp  
--destination-port 6000:6010 -j DROP
```

```
iptables -A INPUT -p udp  
--destination-port 6000:6010 -j DROP
```

#Pozwalamy na połączenia z ssh

```
iptables -A INPUT -p tcp --destination-port ssh  
-j ACCEPT
```

# Przykładowy łańcuch

## (5)

```
#Pozwalamy na wchodzące polaczenia smtp
iptables -A INPUT -p tcp --destination-port smtp
-j ACCEPT

#Pozwalamy na ruch http/https
iptables -A INPUT -p tcp --destination-port www -
j ACCEPT
iptables -A INPUT -p udp --destination-port www -
j ACCEPT
iptables -A INPUT -p tcp --destination-port https
-j ACCEPT
iptables -A INPUT -p udp --destination-port https
-j ACCEPT
```

# Przykładowy łańcuch

## (6)

#Akceptujemy połączenia zainicjowane przez nas

```
iptables -A INPUT -p TCP -m state --state  
ESTABLISHED,RELATED -j ACCEPT
```

# Dziękuję za uwagę

Komentarze i uwagi proszę  
kierować na:

[security@man.poznan.pl](mailto:security@man.poznan.pl)