# Rozszerzenia Netfilter HOWTO


## Fabrice Marie `fabrice@celestix.com` lista pocztowa `netfilter-devel@lists.samba.org`



Wersja oryginalna: 1.20

Oryginał tego dokumentu znajduje się pod adresem: http://netfilter.samba.org/



## Tłumaczenie: Łukasz Bromirski, `l.bromirski@mr0vka.eu.org`



Wersja tłumaczenia: 1.1, $Date: 2002/08/22 21:29:41 $

Oryginał tego tłumaczenia znajduje się pod adresem: http://mr0vka.eu.org/tlumaczenia/netfilter-extensions.html


---


*Dokument ten opisuje jak zainstalować i używać rozszerzeń do iptables dla infrastruktury netfilter.*


---

# 1. Wprowadzenie

Witam. Mam świetną okazję by podziękować wielu ludziom za całą masę czasu spędzoną przy kodowaniu, testowaniu, zgłaszaniu błędów i używaniu netfilter. Dzięki wam wszystkim !!

Ten dokument zakłada, że przeczytałeś i zrozumiałeś dokument Rusty'ego Filtrowanie pakietów w Linuksie 2.4 HOWTO. Zakładam również, że wiesz jak prawidłowo skompilować i zainstalować kernel.

Dystrybucja `iptables` zawiera pewne rozszerzenia, które nie są używane przez zwykłych użytkowników i tak naprawdę są nadal w jakimś stopniu eksperymentalne, lub czekają na wprowadzenie do kernela. Zwykle nie kompiluje się ich, chyba, że jasno sobie tego zażyczysz.

Najnowsza wersja tego dokumentu powinna być dostępna pod adresem http://www.netfilter.org/documentation/index.html#HOWTO.

Celem tego HOWTO jest pomoc ludziom, którzy zaczynają pracę z rozszerzeniami netfilter, przez wyjaśnienia jak je zainstalować a następnie używać.



# 2. Patch-O-Matic

# 2.1 Co to jest Patch-O-Matic ?

Plik Makefile `iptables` zawiera funkcjonalność nazwaną `patch-o-matic' (lub `p-o-m'). p-o-m kieruje twoją drogą przez proces selekcji łat, które możesz chcieć zaaplikować, oraz automatycznie dodaje je do kernela.

Po pierwsze, powinieneś ściągnąć najnowsze drzewo CVS, by być pewnym że pracujesz z najnowszymi wersjami rozszerzeń. By to zrobić, wykonaj:

```
# cvs -d :pserver:cvs@pserver.samba.org:/cvsroot login
# cvs -z3 -d :pserver:cvs@pserver.samba.org:/cvsroot co netfilter
```

Stworzy to główny katalog `netfilter/' a następnie ściągnie do niego wszystkie potrzebne pliki.

Upewnij się, że źródła twojego kernela są w `/usr/src/linux/'. Jeśli z jakiś powodów kernel, który będziesz łatał nie

znajduje się w tym katalogu, musisz ustawić zmienną środowiskową KERNEL_DIR na katalog zawierający źródła kernela:

```
# export KERNEL_DIR=/ścieżka/do/linuksa
```

Upewnij się również, że wykonałeś sprawdzenie **zależności** ( ang. *dependencies* ). Jeśli nie jesteś pewien:

```
# cd /usr/src/linux/
# make dep
```

Możesz teraz przejść do katalogu netfilter, wejść do katalogu `userspace/' i wykonać p-o-m.

# 2.2 Praca z Patch-O-Matic

Uruchommy w katalogu `userspace/' p-o-m :

```
# make patch-o-matic

Welcome to Rusty's Patch-o-matic!

Each patch is a new feature: many have minimal impact, some do not.
Każda łata to nowa funkcjonalność: wiele z nich wnosi niewielkie zmiany, ale niektóre
Almost every one has bugs, so I don't recommend applying them all!
Prawie każda ma jakieś błędy, więc nie rekomenduję nakładania ich w ogóle!
-------------------------------------------------------

Already applied: 2.4.1 2.4.4
Już nałożone: 2.4.1 2.4.4
Testing... name_of_the_patch NOT APPLIED ( 2 missing files)
Testuje... nazwa_łatki NIE NAŁOŻONA ( znaleziono 2 brakujące pliki )
The name_of_the_patch patch:
nazwa_łatki:
   Here usually is the help text describing what
   Tutaj znajduje się zwykle tekst objaśniający
   the patch is for, what you can expect from it,
   do czego jest łata, czego możesz się po niej spodziewać,
   and what you should not expect from it.
   a czego nie.
Do you want to apply this patch [N/y/t/f/q/?]
Czy chcesz nałożyć łatkę
```

p-o-m przejdzie przez wszystkie łaty. Jeśli zostały już nałożone, zobaczysz napis `Already applied:' w pierwszej linii. Jeśli nie, wyświetli nazwę łaty z krótkim objaśnieniem. p-o-m poinformuje również o tym, co się dzieje: `NOT APPLIED ( n missing files)' oznacza po prostu, że łaty jeszcze nie nałożono, a `NOT APPLIED ( n rejects out of n hunks)' oznacza generalnie, że:

1. Łata nie może być nałożona, lub
2. że łata już została włączona do kernela, który próbujesz łatać. re trying to patch.

Na koniec, zapyta o twoją decyzję - łatać czy nie.

- Naciśnij po prostu Enter jeśli nie chcesz jej aplikować.
- Naciśnij `y' jeśli chcesz by p-o-m przetestował łatę i zaaplikował ją; jeśli test się nie powiedzie, poinformuje o tym i zapyta ponownie o decyzję; jeśli wszystko pójdzie dobrze, zobaczysz nazwę łatki i tekst `Already Applied'.
- Naciśnij `t' jeśli chcesz tylko przetestować łatę.
- Naciśnij `f' by p-o-m wymusił zainstalowanie łaty.
- Możesz również nacisnąć `q', jeśli chcesz opuścić p-o-m.

Dobrą zasadą jest przeczytanie opisu łaty zanim się ją nałoży. Ponieważ obecnie jest bardzo dużo oficjalnych łat dla patch-o-matic (i prawdopodobnie jeszcze więcej nieoficjalnych) absolutnie nie zalecam stosowania ich wszystkich ! Powinieneś poważnie przemyśleć swój wybór i nałożyć tylko te, których faktycznie potrzebujesz, nawet jeśli oznacza to rekompilację netfilter gdy będziesz potrzebował więcej łat.

Stworzono nową wersję patch-o-matic, która pokazuje tylko łaty o których wiadomo, że nałożą się bezproblemowo, lub

przynajmniej nie zniszczą efektu innych łat. By ją wywołać, napisz :

```
# make most-of-pom
```

Działa to tak samo jak patch-o-matic, jeśli chodzi o łatanie. Zaoszczędzisz sobie po prostu oglądania łat tylko dla programistów.

## 2.3 A co dalej ?

W momencie gdy nałożyłeś już wszystkie łaty, które chciałeś, następnym krokiem będzie rekompilacja kernela i zainstalowanie go. To HOWTO tego nie objaśni, przeczytaj Linux Kernel HOWTO.

Podczas konfiguracji swojego kernela, zauważysz nowe opcje w `Networking Options -> Netfilter Configuration'. Wybierz opcje, których potrzebujesz, przekompiluj i zainstaluj nowy kernel.

Jeśli już się to stało, możesz zainstalować paczkę `iptables' z katalogu `userspace/' w ten sposób:

```
# make all install
```

I to wszystko! Twoje nowe `iptables` zostało właśnie zainstalowane! Teraz czas na zabawę z nową funkcjonalnością.

# 3. Nowe testy netfilter

W tej sekcji postaram się omówić użycie nowych testów. Łaty wymieniane będą w kolejności alfabetycznej. Nie poruszymy łat, które powodują uszkadzanie innych łat, choć być może kiedyś tak się stanie.

Ogólnie rzecz biorąc, możesz uzyskać pomoc dla testów wpisując:

```
# iptables -m test_o_który_ci_chodzi --help
```

Wyświetli to normalną pomoc iptables, oraz dodatkowe informacje dotyczące `testu_o_który_ci_chodzi' na końcu.

## 3.1 Łata ah-esp

Przygotowana przez Yon Uriarte <yon@astaro.de> dodaje dwa nowe testy:

- `ah' : pozwala dopasowywać pakiety AH na podstawie **Indeksu Parametrów Bezpieczeństwa** ( ang. *Security Parameter Index, SPI* )
- `esp': pozwala dopasowywać pakiety ESP na podstawie ich SPI

Łata będzie użyteczna dla ludzi używających IPSEC, którzy chcą kontrolować połączenia na podstawie ich SPI.

Na przykład, by odrzucić pakiety AH które mają SPI równe 500:

```
# iptables -A INPUT -p 51 -m ah --ahspi 500 -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       ipv6-auth--  anywhere             anywhere           ah spi:500
```

Opcje dostępne dla testu `ah' to:

- `--ahspi [!] spi[:spi]`

  -> match spi (zakres)

Test `esp' działa dokładnie tak samo:

```
# iptables -A INPUT -p 50 -m esp --espspi 500 -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       ipv6-crypt--  anywhere             anywhere           esp spi:500
```

Opcje dostępne dla testu `esp' to:

- `--espspi [!] spi[:spi]`

  -> match spi (zakres)

Nie zapomnij podać właściwego protokołu, czyli `-p 50' lub `-p 51' ( dla odpowiednio `esp' i `ah' ), ponieważ w innym przypadku nie powiedzie się dopisanie reguły do łańcucha.

# 3.2 Łata iplimit

Łata przygotowana przez Gerd Knorr <kraxel@bytesex.org> dodaje nowy test, pozwalający na ograniczanie jednoczesnych połączeń TCP z konkretnej sieci lub komputera.

Na przykład, ograniczmy ilość jednoczesnych połączeń HTTP wykonywanych przez pojedynczy adres IP do 4:

```
# iptables -A INPUT -p tcp --syn --dport http -m iplimit --iplimit-above 4 -j REJECT

# iptables --list
Chain INPUT (policy ACCEPT)
target   prot opt source    destination
REJECT   tcp  --  anywhere  anywhere     tcp dpt:http flags:SYN,RST,ACK/SYN #conn/32
```

Możesz również ograniczyć liczbę jednoczesnych połączeń na przykład dla całej klasy A:

```
# iptables -A INPUT -p tcp --syn --dport http -m iplimit --iplimit-mask 8 --iplimit-a

# iptables --list
Chain INPUT (policy ACCEPT)
target   prot opt source    destination
REJECT   tcp  --  anywhere  anywhere     tcp dpt:http flags:SYN,RST,ACK/SYN #conn/8 >
```

Opcje dostępne dla testu `iplimit' to:

- `[!] --iplimit-above n`

  - pasuje jeśli ilość istniejących połączeń nie jest wyższa niż n
- `--iplimit-mask n`

  - grupuje komputery przez użycie maski

# 3.3 Łata ipv4options

Łata autorstwa Fabrice MARIE <fabrice@celestix.com> dodaje nowy test, pozwalający dopasowywać pakiety na podstawie ustawionych w nich opcjach IP.

Na przykład, odrzućmy wszystkie pakiety z ustawioną opcją **zapisz trasę** (ang.*record-route*) lub **stempel czasu** (ang.*timestamp*):

```
# iptables -A INPUT -m ipv4options --rr -j DROP
# iptables -A INPUT -m ipv4options --ts -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
```

```
target     prot opt source               destination
DROP       all  --  anywhere             anywhere            IPV4OPTS RR
DROP       all  --  anywhere             anywhere            IPV4OPTS TS
```

Opcje dostępne dla testu `ipv4options' to:

- `--ssrr`

  - pasuje do pakietów z ustawioną flagą `strict source routing'
- `--lsrr`

  - pasuje do pakietów z ustawioną flagą `loose source routing'
- `--no-srr`

  - pasuje do pakietów bez ustawionej flagi `source routing'
- `--rr`

  - pasuje do pakietów z ustawioną flagą `record route flag'
- `[!] --ts`

  - pasuje do pakietów z ustawioną flagą `timestamp'
- `[!] --ra`

  - pasuje do pakietów z ustawioną flagą `router-alert'
- `[!] --any-opt`

  - pasuje do pakietów z ustawioną przynajmniej jedną opcją (lub bez opcji IP jeśli użyje się negacji (!)).

# 3.4 Łata length

Łata autorstwa James Morris <jmorris@intercode.com.au> dodaje nowy test, pozwalający dopasowywać pakiety na podstawie ich długości.

Na przykład, odrzućmy wszystkie pingi o długości większej niż 85 bajtów:

```
# iptables -A INPUT -p icmp --icmp-type echo-request -m length --length 85:0xffff -j

# ptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere           icmp echo-request length
```

Opcje dostępne dla testu `length' to:

- `[!] --length długość[:długość]`

  - testuje pakiety o określonej długości lub mieszczące się w podanym zakresie wielkości (włącznie)

Wartości, których nie podano są dodawane automatycznie. Minimalna wartość to 0, maksymalna 65535.

# 3.5 Łata mport

Łata autorstwa Andreas Ferber <af@devcon.net> dodaje nowy test, pozwalający na podanie zestawu pojedynczych portów i ich zakresów, dla protokołów TCP i UDP.

Na przykład, chcąc zablokować ftp, ssh, telnet i http w jednej linii możesz napisać:

```
# iptables -A INPUT -p tcp -m mport --ports 20:23,80 -j DROP
```

```
# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere           mport ports ftp-data:telr
```

Opcje dostępne dla testu `mport' to:

- `--source-ports port[,port:port,port...]`

  - testuje porty źródłowe
- `--sports port[,port:port,port...]`

  - testuje porty źródłowe
- `--destination-ports port[,port:port,port...]`

  - testuje porty docelowe
- `--dports port[,port:port,port...]`

  - testuje porty docelowe
- `--ports port[,port:port,port]`

  - testuje zarówno porty źródłowe jak i docelowe

# 3.6 Łata nth

Łata autorstwa Fabrice MARIE <fabrice@celestix.com> dodaje nowy test, sprawdzający czy dany pakiet nie jest n-tym pasującym do reguły.

Na przykład, jeśli chcesz odrzucać co drugi ping, napisz:

```
# iptables -A INPUT -p icmp --icmp-type echo-request -m nth --every 2 -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere           icmp echo-request every 2
```

Rozszerzenie dodane przez Richarda Wagnera <rwagner@cloudnet.com> umożliwia prosty i szybki sposób na wykonanie rozkładania obciążenia zarówno dla połączeń wychodzących jak i wchodzących.

Na przykład, jeśli chcesz rozłożyć obciążenie pomiędzy trzy adresy: 10.0.0.5, 10.0.0.6 i 10.0.0.7, możesz napisać:

```
# iptables -t nat -A POSTROUTING -o eth0 -m nth --counter 7 --every 3 --packet 0 -j S
# iptables -t nat -A POSTROUTING -o eth0 -m nth --counter 7 --every 3 --packet 1 -j S
# iptables -t nat -A POSTROUTING -o eth0 -m nth --counter 7 --every 3 --packet 2 -j S

# iptables -t nat --list
Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  anywhere             anywhere           every 3th packet #0 to:10
SNAT       all  --  anywhere             anywhere           every 3th packet #1 to:10
SNAT       all  --  anywhere             anywhere           every 3th packet #2 to:10
```

Opcje dostępne dla celu `nth' to:

- `--every Nth`

  - pasuje do n-tego pakieto
- `[--counter]  num`

  - użyj licznika 0-15 (domyślnie:0)

- `[--start] num`

  - zainicjuj licznik wartością `num'; musi być pomiędzy 0 a n-ty - 1
- `[--packet] num`

  - pasuje do pakietu numer `num'; musi być pomiędzy 0 a n-ty - 1; jeśli `--packet' używany jest jako licznik, musi być n reguł `--packet', pokrywających wszystkie wartości pomiędzy 0 a (n-ty - 1) włącznie

# 3.7 Łata pkttype

Łata autorstwa Michala Ludviga <michal@logix.cz> dodaje nowy test, pozwalający dopasowywać pakiety na podstawie ich typu: host/broadcast/multicast.

Jeśli na przykład chcesz odrzucać wszystkie pakiety broadcast'owe:

```
# iptables -A INPUT -m pkttype --pkt-type broadcast -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere           PKTTYPE = broadcast
```

Opcje dostępne dla celu `pkttype' to:

- `--pkt-type [!] packettype`

  - pasuje do pakietu, który jest pakietem typu
    - `host`

      - do nas
    - `broadcast`

      - do wszystkich
    - `multicast`

      - do grupy

# 3.8 Łata pool

Łata autorstwa Patricka Schaafa <bof@bof.de>. Joakim Axelsson i Patrick są w trakcie przepisywania go od nowa, więc prawdopodobnie zamienią tą sekcję niedługo tym co tak naprawdę stworzyli.

# 3.9 Łata psd

Łata autorstwa Dennisa Koslowskiego <dkoslowski@astaro.de> dodaje nowy test, pozwalający wykryć skanowanie portów.

W swej najprostszej postaci, `psd' może być użyty tak:

```
# iptables -A INPUT -m psd -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target  prot opt source    destination
DROP    all  --  anywhere  anywhere    psd weight-threshold: 21 delay-threshold: 300
```

Opcje dostępne dla testu `psd' to:

- `[--psd-weight-threshold próg]`

- waga progu detekcji skanowania portów
- `[--psd-delay-threshold zwłoka]`

  - waga progu zwłoki skanowania portów
- `[--psd-lo-ports-weight lo]`

  - waga uprzywilejowanych portów
- `[--psd-hi-ports-weight hi]`

  - waga wysokich portów

# 3.10 Łata random

Łata autorstwa Fabrice MARIE <fabrice@celestix.com> dodaje nowy test, pozwalający na dopasowywanie losowych pakietów na podstawie podanego prawdopodobieństwa.

Na przykład, jeśli chcesz odrzucać połowę pingów losowo, możesz napisać:

```
# iptables -A INPUT -p icmp --icmp-type echo-request -m random --average 50 -j DROP

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source       destination
DROP       icmp --  anywhere     anywhere        icmp echo-request  random 50%
```

Opcje dostępne dla testu `random' to:

- `[--average] procent`

  - prawdopodobieństwo w procentach dopasowania. Jeśli zostanie pominięte, przyjmowane jest prawdopodobieństwo 50%. Wartość musi być w przedziale 1-99.

# 3.11 Łata realm

Łata autorstwa Sampsa Ranta <sampsa@netsonic.fi> dodaje nowy test, umożliwiający wykorzystanie kluczy sfery ( ang. *realm* ) pochodzących z routingu jako testów podobnych do tych, z klasyfikatora pakietów.

Na przykład, by logować wszystkie wychodzące pakiety do sfery 10, możesz napisać:

```
# iptables -A OUTPUT -m realm --realm 10 -j LOG

# iptables --list
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere           REALM match 0xa LOG level
```

Opcje dostępne dla testu `realm' to:

- `--realm [!] wartość[/maska]`

  - dopasuj do sfery

# 3.12 Łata record-rpc

Łata autorstwa Marcelo Barbosa Lima <marcelo.lima@dcc.unicamp.br> dodaje nowy test, umożliwiający sprawdzanie, czy źródło pakietu żądało tego portu już wcześniej przy użyciu portmapper'a, czy jest to nowe żądanie typu GET do portmapper'a; dzięki temu możliwe jest filtrowanie wywołań RPC.

By użyć śledzenia połączeń RCP, napisz:

```
# iptables -A INPUT -m record_rpc -j ACCEPT

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```

Test nie ma żadnych opcji.

Nie przejmuj się, że informacje o teście nie są drukowanie. Po prostu funkcja print() tego testu jest pusta:

```
/* Prints out the union ipt_matchinfo. */
static void
print(const struct ipt_ip *ip,
      const struct ipt_entry_match *match,
      int numeric)
{
}
```

# 3.13 Łata string

Łata autorstwa Emmanuela Rogera <winfield@freegates.be> dodaje nowy test, umożliwiający dopasowywanie ciągów znaków w dowolnym miejscu pakietu.

Jeśli na przykład chcesz wyłapywać ciąg znaków `cmd.exe' i kolejkować je do systemu IDS w przestrzeni użytkownika, napisz:

```
# iptables -A INPUT -m string --string 'cmd.exe' -j QUEUE

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
QUEUE      all  --  anywhere             anywhere           STRING match cmd.exe
```

Proszę jednak używać tego testu z uwagą. Wiele ludzi chce użyć tego celu do zatrzymania robaków, używając celu DROP. Jest to poważny błąd i zostanie ominięty przez każdy sposób unikania wykrycia przez IDS.

Wielu ludzi używało tej łaty do powstrzymania pewnych metod HTTP, takich jak POST czy GET, przez odrzucanie każdego pakietu HTTP zawierającego ciąg znaków POST. Proszę zrozumieć, że lepiej tą pracę wykona proxy filtrujące. Co więcej, dokument HTML zawierający słowo POST również zostanie odrzucony. Łatę stworzono aby można było kolejkować ciekawe pakiety do przestrzeni użytkownika, nic więcej.

Opcje dostępne dla testu `string' to:

- `--string [!] ciąg znaków`

  - dopasuj ciąg znaków w pakiecie

# 3.14 Łata time

Łata autorstwa Fabrice MARIE <fabrice@celestix.com> dodaje nowy test, pozwalający sprawdzać pakiety na podstawie czasu ich przyjścia lub opuszczania maszyny.

Na przykład, by zaakceptować pakiety które przybywają pomiędzy ósmą rano a osiemnastą wieczorem od poniedziałku do piątku, możesz napisać:

```
# iptables -A INPUT -m time --timestart 8:00 --timestop 18:00 --days Mon,Tue,Wed,Thu,

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
ACCEPT     all  --  anywhere        anywhere       TIME from 8:0 to 18:0 on Mon,Tue
```

Opcje dostępne dla testu `time' to: Supported options for the time match are :

- `--timestart value`

  - minimalny czas w formacie HH:MM
- `--timestop value`

  - maksymalny czas w formacie HH:MM
- `--days listofdays`

  - lista dni (ważna wielkość liter)
    - Mon
    - Tue
    - Wed
    - Thu
    - Fri
    - Sat
    - Sun

# 3.15 Łata ttl

Łata autorstwa Haralda Welte <laforge@gnumonks.org> dodaje nowy test, pozwalający dopasowywać pakiet na podstawie jego TTL.

Jeśli na przykład chcesz logować pakiety z TTL mniejszym niż 5, napisz:

```
# iptables -A INPUT -m ttl --ttl-lt 5 -j LOG

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere           TTL match TTL < 5 LOG lev
```

Opcje dostępne dla testu `ttl' to:

- `--ttl-eq wartość`

  - TTL równy wartości
- `--ttl-lt wartość`

  - TTL mniejszy niż wartość
- `--ttl-gt wartość`

  - TTL większy niż wartość

# 4. Now cele netfilter

W tej sekcji postaram się wyjaśnić sposób użycia nowych celów dla netfilter. Łaty przedstawione zostaną w kolejności alfabetycznej. Nie poruszymy łat, które powodują uszkadzanie innych łat, choć być może kiedyś tak się stanie.

Ogólnie rzecz biorąc, możesz uzyskać pomoc dla celów wpisując:

```
# iptables -m cel_o_który_ci_chodzi --help
```

Wyświetli to normalną pomoc iptables, oraz dodatkowe informacje dotyczące `celu_o_który_ci_chodzi' na końcu.

# 4.1 Łata ftos

Łata autorstwa Matthew G. Marsh <mgm@paktronix.com> dodaje nowy cel, umożliwiający ustawianie wartości TOS pakietu na określoną wartość.

Na przykład, jeśli chcesz ustawiać wartość TOS wszystkich wychodzących pakietów na wartość 15, napisz:

```
# iptables -t mangle -A OUTPUT -j FTOS --set-ftos 15

# iptables -t mangle --list
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
FTOS       all  --  anywhere             anywhere           TOS set 0x0f
```

Opcje dostępne dla celu `FTOS' to:

- `--set-ftos wartość`

  - ustaw pole TOS na wartość. Wartość może być w postaci decymalnej ( np.: `32` ) lub heksadecymalnej ( np.: `0x20` )

# 4.2 Łata IPV4OPTSSTRIP

Łata autorstwa Fabrice MARIE <fabrice@celestix.com> dodaje nowy cel, umożliwiający obranie pakietu IPv4 z jego opcji.

Używa się jej po prostu tak:

```
# iptables -t mangle -A PREROUTING -j IPV4OPTSSTRIP

# iptables -t mangle --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
IPV4OPTSSTRIP  all  --  anywhere             anywhere
```

Ten cel nie udostępnia żadnych opcji.

# 4.3 Łata NETLINK

Łata autorstwa Gianni Tedesco <gianni@ecsc.co.uk> dodaje nowy cel, który umożliwia wysyłanie odrzucanych pakietów do przestrzeni użytkownika przez gniazdo netlink.

Na przykład, by odrzucając wszystkie pingi wysyłać je jednocześnie do wspomnianego gniazda, napisz:

```
# iptables -A INPUT -p icmp --icmp-type echo-request -j NETLINK --nldrop

# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
NETLINK    icmp --  anywhere             anywhere           icmp echo-request nldrop
```

Opcje dostępne dla celu `NETLINK' to:

- `--nldrop`

  - po przesłaniu odrzuć pakiet
- `--nlmark <numer>`

  - zaznacz pakiet
- `--nlsize <bajtów>`

  - ogranicz rozmiar pakietu

Po więcej informacji dotyczących gniazd netlink, odsyłam pod adres http://www.skyfree.org/linux/kernel_network/netlink.html.

# 4.4 Łata NETMAP

Łata autorstwa Svenning Soerensen <svenning@post5.tele.dk> dodaje nowy cel, pozwalający na mapowanie adresów sieci 1:1, przy zachowaniu adresu hosta.

Na przykład, jeśli chciałbyś zmienić adres docelowy połączeń przychodzących z 1.2.3.0/24 na 5.6.7.0/24, napisz:

```
# iptables -t nat -A PREROUTING -d 1.2.3.0/24 -j NETMAP --to 5.6.7.0/24

# iptables -t nat --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
NETMAP     all  --  anywhere             1.2.3.0/24          5.6.7.0/24
```

Opcje dostępne dla celu `NETMAP' to:

- `--to adres[/maska]`

  - adres sieciowy do mapowania

# 4.5 Łata SAME

Łata autorstwa Martina Josefssona <gandalf@wlug.westbo.se> dodaje nowy cel, podobny do SNAT, dający danemu klientowi ten sam adres dla każdego połączenia.

Na przykład, jeśli chcesz modyfikować adres źródłowy połączeń na 1.2.3.4-1.2.3.7 możesz napisać:

```
# iptables -t nat -A POSTROUTING -j SAME --to 1.2.3.4-1.2.3.7

# iptables -t nat --list
Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SAME       all  --  anywhere             anywhere            same:1.2.3.4-1.2.3.7
```

Opcje dostępne dla celu `SAME' to:

- `--to <ipaddr>-<ipaddr>`

  - adresy na które mapować źródło. Mogą być podane wielokrotnie, dla wielu zakresów.
- `--nodst`

  - nie używaj adresu docelowego przy wybieraniu źródłowego

# 4.6 Łata tcp-MSS

Łata autorstwa Marca Bouchera <marc+nf@mbsi.ca> dodaje nowy cel, który pozwala badać i zmieniać wartość MSS pakietów TCP SYN, co pozwala na kontrolowanie maksymalnej wartości dla danego połączenia.

Tak jak tłumaczy sam Marc, TO JEST HACK, używany w przypadku napotkania zbrodniczo debilnych dostawców internetowych, lub serwerów, które blokują pakiety ICMP Fragmentation Needed.

Typowe użycie wygląda tak:

```
# iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu

# iptables --list
```

```
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
TCPMSS     tcp  --  anywhere             anywhere           tcp flags:SYN,RST/SYN TCF
```

Opcje dostępne dla celu `tcp-MSS' to (można podawać tylko pojedynczo):

- `--set-mss wartość`

  ustaw wartość MSS na podaną liczbę
- `--clamp-mss-to-pmtu`

  automatycznie dopasuj wartość MSS do (MTU_ścieżki - 40)

# 4.7 Łata TTL

Łata autorstwa Haralda Welte <laforge@gnumonks.org>, dodaje nowy cel umożliwiający użytkownikowi ustawienie wartości TTL dla pakietu IP, lub zwiększanie/zmniejszanie jej o określoną wartość.

Na przykład, jeśli chcesz ustawić TTL wszystkich wychodzących połączeń na wartość 126, napisz:

```
# iptables -t mangle -A OUTPUT -j TTL --ttl-set 126

# iptables -t mangle --list
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
TTL        all  --  anywhere             anywhere           TTL set to 126
```

Opcje dostępne dla celu `TTL' to:

- `--ttl-set wartość`

  - ustaw wartość TTL na <wartość>
- `--ttl-dec wartość`

  - zmniejsz wartość TTL o <wartość>
- `--ttl-inc wartość`

  - zwiększ wartość TTL o <wartość>

# 4.8 Łata ulog

Łata autorstwa Haralda Welte <laforge@gnumonks.org>, dodaje nowy cel, udostępniający trochę bardziej zaawansowany mechanizm logowania niż standardowy cel `LOG'. Katalog `libipulog/' zawiera bibliotekę obsługującą odbieranie wiadomości `ULOG'.

Harald utrzymuje stronę http://www.gnumonks.org/projects/ulogd zawierającą odpowiednią dokumentację dla celu, więc nie ma sensu cytować tego tutaj.

# 5. Nowe łaty dla śledzenia połączeń

W tej sekcji przedstawimy dostępne łaty śledzenia połączeń i NAT. By ich użyć, załaduj odpowiednie moduły (z opcjami, jeśli są wymagane).

# 5.1 Łata eggdrop-conntrack

Łata autorstwa Magnusa Sandini <magnus@sandin.cx> dodaje obsługę dla połączeń botów eggdrop.

## 5.2 Łata ftp-fxp

Łata autorstwa Magnusa Sandini <magnus@sandin.cx> dodaje obsługę FXP do śledzenia sesji FTP. Nie działa jeszcze użycie FXP do maszyn z demonem ftp za NAT'em. By włączyć moduł, napisz:

```
# modprobe ip_conntrack_ftp.o fxp=1
```

Łata ostrzega informacją o bezpieczeństwie: UWAGA, użycie tej łaty i włączenie jej `OBNIŻY` poziom bezpieczeństwa oferowany przez moduł śledzenia połączeń FTP. Używaj go z dużą ostrożnością (i tylko wtedy, gdy wiesz co robisz).

## 5.3 Łata irc-conntrack-nat

Łata autorstwa Haralda Welte <laforge@gnumonks.org> dodaje obsługę DCC przez NAT.

## 5.4 Łata record-rpc

Łata autorstwa Marcelo Barbosa Lima <marcelo.lima@dcc.unicamp.br> umożliwia netfilter śledzenie żądań portmapper'a wykonywanych przez TCP i UDP.

## 5.5 Łata snmp-nat

Łata autorstwa Jamesa Morrisa <jmorris@intercode.com.au> dodaje do NAT możliwość operowania na połączeniach SNMP. Jest to wersja protokołu SNMP-ALG opisana w RFC 2962: http://www.faqs.org/rfcs/rfc2962.html i działa przez modyfikację adresów IP w pakietach SNMP tak, by pasowały do mapowań NAT w warstwie IP.

## 5.6 Łata talk-conntrack-nat

Łata autorstwa Jozsefa Kadlecsika <kadlec@blackhole.kfki.hu> dodaje możliwość śledzenia przez netfilter połaczeń protokołu `talk', jak również obsługi ich NATowania. Domyślnie, obsługiwane są zarówno `otalk' ( port UDP 517 ) jak i `talk' ( port UDP 518 ). Obsługę obu programów można wyłączać i włączać selektywnie, za pomocą parametrów modułów `ip_conntrack_talk' i `ip_nat_talk modules'. Dostępne opcje to:

- otalk = 0 | 1
- talk = 0 | 1

gdzie `0' oznacza `nie obsługuj' a `1' oznacza `obsługuj' dany protokół.

## 5.7 Łata tcp-window-tracking

Łata autorstwa Jozsefa Kadlecsika <kadlec@blackhole.kfki.hu> dodaje możliwość śledzenia połączeń TCP na zasadach opisanych w artykule Real Stateful TCP Packet Filtering in IP Filter autorstwa Guido van Rooij. Oznacza to obsługę skalowanych okien i obsługiwanie już wcześniej ustanowionych połączeń.

Łata wymaga innej łaty - `ftp-fixes'. Powinna być ona już w standardowym kernelu.

# 6. Nowe testy dla IPv6

W tej sekcji postaram się omówić użycie nowych testów. Łaty wymieniane będą w kolejności alfabetycznej. Nie poruszymy łat, które powodują uszkadzanie innych łat, choć być może kiedyś tak się stanie.

Ogólnie rzecz biorąc, możesz uzyskać pomoc dla testów wpisując:

```
# iptables -m test_o_który_ci_chodzi --help
```

Wyświetli to normalną pomoc iptables, oraz dodatkowe informacje dotyczące `testu_o_który_ci_chodzi' na końcu.

# 6.1 Łata agr

Łata autorstwa Andras Kis-Szabo <kisza@sch.bme.hu> dodaje jeden nowy test:

- `agr' : pozwala dopasować pakiet IPv6 na podstawie jego parametrów adresowych

Może ona być pomocna dla ludzi używających schematu adresowania EUI-64, którzy chcieliby sprawdzić pakiety pod kątem dostarczonego adresu w sieci LAN.

Na przykład, przekierujemy wszystkie pakiety posiadające prawidłowe adresy EUI-64:

```
# ip6tables -N ipv6ok
# ip6tables -A INPUT -m agr -j ipv6ok
# ip6tables -A INPUT -s ! 3FFE:2F00:A0::/64 -j ipv6ok
# ip6tables -A INPUT -j LOG
# ip6tables -A ipv6ok -j ACCEPT

# ip6tables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ipv6ok     all      anywhere             anywhere           AGR
ipv6ok     all     !3ffe:2f00:a0::/64    anywhere
LOG        all      anywhere             anywhere           LOG level warning

Chain ipv6ok (2 references)
target     prot opt source               destination
ACCEPT     all      anywhere             anywhere
```

Test nie ma żadnych opcji.

# 6.2 Łata ipv6header

Łata autorstwa Andras Kis-Szabo <kisza@sch.bme.hu> dodaje nowy test, pozwalający sprawdzać pakiety na podstawie ich rozszerzonych nagłówków.

Na przykład, odrzucajmy pakiety z dodanymi hop-by-hop, nagłówkami ipv6-route i zawartością protokołu:

```
# ip6tables -A INPUT -m ipv6header --header hop-by-hop,ipv6-route,protocol -j DROP

# ip6tables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all      anywhere             anywhere           ipv6header flags:hop-by-h
```

A teraz, odrzućmy pakiety które mają rozszerzony nagłówek ipv6-route:

```
# ip6tables -A INPUT -m ipv6header --header ipv6-route --soft -j DROP

# ip6ptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all      anywhere             anywhere           ipv6header flags:ipv6-rou
```

Dostępne opcje dla celu to:

- `--header [!] nagłówki`

  - możesz podać interesujące cię nagłówki z jego opcjami. Akceptowane formaty to:
    - hop,dst,route,frag,auth,esp,none,proto
    - hop-by-hop,ipv6-opts,ipv6-route,ipv6-frag,ah,esp,ipv6-nonxt,protocol
    - 0,60,43,44,51,50,59
- `--soft`

- możesz określić tryb `miękki' - sprawdzana jest tylko obecność nagłówka, nie cały test!

# 6.3 Łata ipv6-ports

Łata Jana Rekorajskiego <baggins@pld.org.pl> dodaje cztery nowe testy:

- `limit' : pozwala ograniczyć ilość jednoczesnych połączeń TCP od określonego komputera lub sieci
- `mac' : pozwala dopasować pakiet na podstawie jego adresu MAC
- `multiport' : pozwala dopasować pakiet na podstawie zestawu portów i zakresów portów dla protokołów TCP i UDP
- `owner' : pozwala dopasować pakiet na podstawie identyfikatora procesu nadającego

Są to wersje łat dla protokołu IPv4, sprawdź wyżej szczegóły.

# 6.4 Łata length

Łata autorstwa Imrana Patela <ipatel@crosswinds.net> dodaje nowy test umożliwiający dopasowywanie pakietu na podstawie jego długości. (bezwstydna adaptacja łaty dla IPv4 autorstwa Jamesa Morrisa <jmorris@intercode.com.au>)

Na przykład, odrzućmy wszystkie pingi z długością pakietu większą niż 85 bajtów:

```
# ip6tables -A INPUT -p ipv6-icmp --icmpv6-type echo-request -m length --length 85:0x

# ip6ptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       ipv6-icmp --  anywhere             anywhere           ipv6-icmp echo-reque
```

Opcje dostępne dla testu to:

- `[!] --length długość[:długość]`

  - testuje pakiety o określonej długości lub mieszczące się w podanym zakresie wielkości (włącznie)

Wartości, których nie podano są dodawane automatycznie. Minimalna wartość to 0, maksymalna 65535.

# 7. Now cele dla IPv6

W tej sekcji postaram się omówić użycie nowych celów. Łaty wymieniane będą w kolejności alfabetycznej. Nie poruszymy łat, które powodują uszkadzanie innych łat, choć być może kiedyś tak się stanie.

Ogólnie rzecz biorąc, możesz uzyskać pomoc dla celu wpisując:

```
# iptables -m cel_o_który_ci_chodzi --help
```

Wyświetli to normalną pomoc iptables, oraz dodatkowe informacje dotyczące `celu_o_który_ci_chodzi' na końcu.

# 7.1 Łata LOG

Łata autorstwa Jana Rekorajskiego <baggins@pld.org.pl> dodaje nowy cel umożliwiający logowanie pakietów dokładnie tak, jak dla IPv4.

Przykłady są takie same jak dla iptables, zajrzyj na stronę podręcznikową po szczegóły!

# 7.2 Łata REJECT

Łata autorstwa Haralda Welte <laforge@gnumonks.org> dodaje nowy cel, umożliwiający odrzucenie pakietu dokładnie

tak, jak dla IPv4.

Przykłady są takie same jak dla iptables, zajrzyj na stronę podręcznikową po szczegóły!

# 8. Nowe łaty dla śledzenia połączeń IPv6

Śledzenie połączeń nie jest jeszcze obsługiwane.

# 9. Współpraca

## 9.1 Dodawanie nowego rozszerzenia

Podstawowy skład netfilter zawsze mile widzi nowe rozszerzenia i poprawki. W tej sekcji nie opowiem o robieniu paczek z nowych rozszerzeń, by ułatwić ich integrację z patch-o-matic (jeszcze nie). Ale być może, taka informacja znajdzie się tutaj w którejś z nowych wersji dokumentu.

Po pierwsze, powinieneś zapoznać się z Netfilter Hacking HOWTO.

Rusty napisał już przewodnik dla piszących łaty do netfilter, znajduje się on w

```
/ścieżka/do/netfiltercvs/netfilter/userspace/patch-o-matic/NEWPATCHES
```

Możesz też przeczytać najnowszą wersję pod adresem: http://www.samba.org/cgi-bin/cvsweb/~checkout~/netfilter/userspace/patch-o-matic/NEWPATCHES.

Na koniec, dobrym pomysłem jest zapisanie się na listę `netfilter-devel'. Więcej informacji o tym jak to zrobić, znajdziesz na stronie domowej netfilter.

## 9.2 Dodatki do tego HOWTO

Byłoby miło, gdybyś dodał coś do tego HOWTO. By to zrobić, najlepiej wysłać łatę z różnicami pomiędzy głównym dokumentem SGML a twoim, na listę `netfilter-devel'.

# 10. Uwagi od tłumacza

Chciałbym podziękować następującym osobom za uwagi co do tłumaczenia, poprawki i zasugerowanie poprawek:

**Tomasz Sedyka, cindy (at) zaba.tbg.net.pl**
- literówki, literówki i o dziwo, uparty błąd w słowie **pojedyncze**