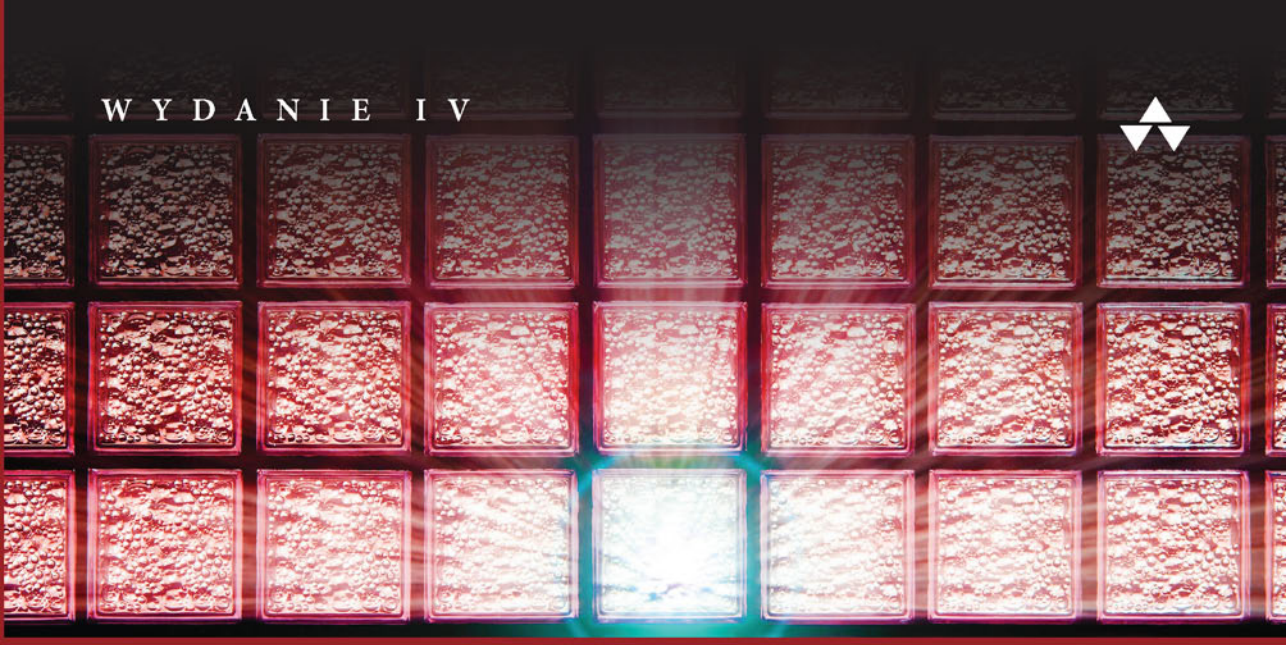


WYDANIE IV

ZAPORY SIECIOWE W SYSTEMIE LINUX®

Kompendium wiedzy o nftables

STEVE SUEHRING

Helion

Tytuł oryginału: Linux® Firewalls Enhancing Security with nftables and Beyond, Fourth Edition

Tłumaczenie: Lech Lachowski

ISBN: 978-83-283-1297-5

Authorized translation from the English edition, entitled: LINUX FIREWALLS: ENHANCING SECURITY WITH NFTABLES AND BEYOND, Fourth Edition; ISBN 0134000021; by Steve Suehring; published by Pearson Education, Inc, publishing as Addison-Wesley Professional.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
ftp://ftp.helion.pl/przyklady/zasili.zip

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/zasili
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

4

nftables: program do administrowania zaporą sieciową systemu Linux

W rozdziale 3. przedstawiłem `iptables`, wykorzystywany od dawna program administracyjny dla zapór sieciowych systemu Linux. Omówiliśmy składnię oraz różne opcje dostępne w `iptables`. W tym rozdziale przyjrzymy się nowym tabelom Netfilter, czyli programowi `nftables`. Program `nftables` jest dostępny w ramach jądra mainline systemu Linux, począwszy od wersji 3.13.

Różnice pomiędzy iptables i nftables

W jądrze program `nftables` stanowi znaczące odejście od systemu filtrowania `iptables`. Program `nftables` zastępuje funkcjonalność nie tylko `iptables`, ale także `ip6tables` dla IPv6, `arptables` dla filtrowania ARP oraz `ebtables` dla filtrowania pakietów na mostkach Ethernet. Składnie poleceń dla `nftables` i `iptables` różnią się od siebie, a `nftables` umożliwia korzystanie z dodatkowych funkcji skryptowych. Program administracyjny dla `nftables` nazywa się `nft` i za pomocą tego polecenia budowane są zapory sieciowe.

W przeciwieństwie do `iptables`, program `nftables` nie zawiera żadnych wbudowanych tabel. To do administratora należy określenie, które tabele są potrzebne, i dodanie tych tabel wraz z regułami dla przetwarzania. W dalszej części tego rozdziału przyjrzymy się składni programu `nftables` i jego wykorzystaniu do tworzenia zapory sieciowej.

Podstawowa składnia nftables

Polecenie `nft` stanowi program administracyjny, który jest używany do budowy zapory sieciowej. Podstawowa składnia polecenia `nftables` zaczyna się od samego programu `nft`, po którym następują polecenia i podpolecenia oraz różne argumenty i wyrażenia. Oto przykład:

```
nft <polecenie> <podpolecenie> <łańcuch> <definicja_reguły>
```

Do typowych poleceń należą:

- `add`;
- `list`;
- `insert`;
- `delete`;
- `flush`.

Typowe podpolecenia to:

- `table`;
- `chain`;
- `rule`.

Funkcje programu nftables

Program `nftables` zawiera kilka charakterystycznych dla języków programowania wyższego poziomu funkcjonalności, takich jak możliwość definiowania zmiennych i załączania zewnętrznych plików. Program `nftables` może być również stosowany do filtrowania i przetwarzania różnych rodzin adresów. Te rodziny adresów to:

- **ip** — adresy IPv4;
- **ip6** — adresy IPv6;
- **inet** — adresy IPv4 i IPv6;
- **arp** — adresy protokołu ARP (ang. *Address Resolution Protocol*);
- **bridge** — przetwarzanie pakietów, które przechodzą przez interfejsy zmostkowane.

Jeśli nie zostało określone inaczej, domyślną rodziną adresów są adresy IP. Zdolność do przetwarzania różnych rodzin adresów oznacza, że `nftables` ma zastąpić inne mechanizmy filtrowania, takie jak `ebtables` i `arptables`.

Ogólna architektura przetwarzania dla `nftables` polega na określeniu rodziny adresów, do której dana reguła będzie stosowana. Następnie `nftables` wykorzystuje jedną tablicę lub kilka tablic, które zawierają jeden łańcuch lub kilka łańcuchów, a te z kolei zawierają reguły przetwarzania. Reguły przetwarzania dla `nftables` składają się z wyrażeń, takich jak adres, interfejs, porty (lub inne dane zawarte w aktualnie przetwarzanym pakiecie), oraz instrukcji, takich jak `drop`, `queue` oraz `continue`.

> **Wskazówka**
> Tabele zawierają łańcuchy. Łańcuchy zawierają reguły.

Niektóre rodziny adresów zawierają zaczepy (ang. *hooks*) umożliwiające programowi `nftables` uzyskanie dostępu do pakietów, gdy trawersują one przez stos sieciowy w systemie

Linux. Oznacza to, że można wykonać operację na pakiecie, zanim zostanie on przekazany do routowania lub po jego przetworzeniu. Dla rodzin adresów ip, ip6 oraz inet zastosowanie mają następujące zaczepy:

- `prerouting` — pakiety, które właśnie przybyły i nie zostały jeszcze przeroutowane lub przetworzone przez inne części `nftables`.
- `input` — pakiety przychodzące, które zostały odebrane i wysłane poprzez zaczep `prerouting`.
- `forward` — jeśli pakiet będzie wysłany do innego urządzenia, będzie dostępny poprzez zaczep `forward`.
- `output` — pakiety wychodzące z procesów w systemie lokalnym.
- `postrouting` — tuż przed opuszczeniem systemu zaczep `postrouting` udostępnia pakiet do dalszego przetwarzania.

Rodzina adresów ARP wykorzystuje tylko zaczepy `input` i `output`.

Składnia nftables

Samo polecenie `nft` ma kilka opcji, które są dostępne z poziomu wiersza poleceń i nie są związane bezpośrednio z definiowaniem reguł filtrowania. Do tych opcji wiersza poleceń należą:

- `--debug <poziom, [poziom]>` — dodanie debugowania na poziomie <*poziom*>, takim jak `scanner`, `parser`, `eval`, `netlink`, `mnl`, `segtree`, `proto-ctx` lub na wszystkich poziomach.
- `-h | --help` — wyświetla podstawową pomoc.
- `-v | --version` — wyświetla numer wersji `nft`.
- `-n | --numeric` — wyświetla informacje o adresie i porcie w postaci numerycznej, zamiast przeprowadzać rozwiązywanie nazw.
- `-a | --handle` — wyświetla uchwyty reguł.
- `-I | --includepath <katalog>` — dodaje <*katalog*> do ścieżki wyszukiwania dla załączonych plików.
- `-f | --file <nazwa_pliku>` — załącza zawartość pliku <*nazwa_pliku*>.
- `-i | --interactive` — odczytuje dane wejściowe z wiersza poleceń.

Jak już wspomniano, w `nftables` nie ma predefiniowanych tabel. Od Ciebie zależy zdefiniowanie tabel, których chcesz użyć w systemie `nftables`. Polecenia dostępne do definiowania danej reguły zależą od tego, czy pracujesz z tabelą, łańcuchem lub regułą.

Składnia tabeli

Podczas pracy z tabelą dostępne są cztery polecenia:

- `add` — dodaje tabelę;
- `delete` — usuwa tabelę;
- `list` — wyświetla wszystkie łańcuchy i reguły dla tabeli;
- `flush` — czyści wszystkie łańcuchy i reguły w tabeli.

Listę dostępnych tabel możesz wyświetlić za pomocą następującego polecenia (uruchomionego z poziomu użytkownika *root*):

```
nft list tables
```

Pamiętaj, że nie ma żadnych domyślnych tabel dla `nftables`, tak jak miało to miejsce w przypadku `iptables`. Dlatego polecenie `list tables` może nic nie zwrócić, jeśli nie zostały zdefiniowane żadne tabele. Jest to normalne zachowanie, jeżeli dopiero skonfigurowałeś `nftables` i nie zdefiniowałeś jeszcze żadnej zapory sieciowej za pomocą tego programu. Tabelę, która będzie przechowywać standardowe łańcuchy i reguły zapory sieciowej, można zdefiniować w następujący sposób:

```
nft add table filter
```

Po dodaniu tabeli zapory sieciowej polecenie `list tables` zwróci jej nazwę:

```
table filter
```

Dalsze informacje na temat tabeli można uzyskać za pomocą następującego polecenia:

```
nft list table filter
```

Spowoduje to wyświetlanie informacji na temat tabeli, łącznie ze wszystkimi zdefiniowanymi w niej łańcuchami:

```
table ip filter{
}
```

Jak widać na przykładzie, tabela `filter` używa rodziny adresów IP i aktualnie jest pusta.

W tym przykładzie tabela została nazwana `filter`, ale można użyć dowolnej innej nazwy, na przykład `firewall`. Zazwyczaj jednak dla tej tabeli stosuje się właśnie nazwę `filter` i taka nazwa została również użyta w przykładach zawartych w dokumentacji dla programu `nftables`.

W przypadku wyświetlania listy reguł całkiem pomocne jest dodanie opcji `-a`, która pozwala sprawdzić numery uchwytów. Uchwyt może być wykorzystywany do zmodyfikowania lub usunięcia reguły w prosty sposób. To zastosowanie zostanie przedstawione w dalszej części rozdziału, podczas dodawania reguł do zapory sieciowej.

W trakcie wyświetlania listy reguł zapory sieciowej `nftables` przeprowadza rozwiązywanie adresów i portów. To zachowanie można zmodyfikować za pomocą opcji `-n`. Można dodać dwie opcje `-n`, aby zapobiec rozwiązywaniu zarówno adresów, jak i portów:

```
nft list table filter -nn
```

Składnia łańcucha

Podczas pracy z łańcuchem dostępnych jest sześć poleceń:

- `add` — dodanie łańcucha do tabeli;
- `create` — tworzenie łańcucha w tabeli, chyba że łańcuch o tej samej nazwie już istnieje;
- `delete` — usunięcie łańcucha;
- `flush` — wyczyszczenie wszystkich reguł w łańcuchu;
- `list` — wyświetlenie wszystkich reguł w łańcuchu;
- `rename` — zmiana nazwy łańcucha.

Podczas dodawania łańcucha można zdefiniować wspomniany wcześniej zaczep. Ponadto do definicji łańcucha można dodać opcjonalny priorytet.

Istnieją trzy podstawowe rodzaje łańcuchów, które mogą zawierać reguły i mieć również podłączone zaczepy. Rodzaje łańcucha i zaczepu należy zdefiniować podczas tworzenia łańcucha i są one niezbędne do funkcjonowania łańcucha podczas normalnej pracy zapory sieciowej. Jeśli rodzaje łańcucha i zaczepu nie zostaną zdefiniowane, pakiety nie będą routowane do tego łańcucha.

Trzy podstawowe rodzaje łańcuchów są następujące:

- `filter` — używany do filtrowania pakietów;
- `route` — używany do routowania pakietów;
- `nat` — używany do translacji NAT.

Aby pogrupować podobne do siebie reguły, można również dodać inne łańcuchy. Gdy pakiety trawersują przez podstawowy łańcuch, mogą być kierowane do łańcucha zdefiniowanego przez użytkownika lub do kilku takich łańcuchów w celu dodatkowego przetwarzania.

Podczas dodawania łańcucha należy określić tabelę, w której dany łańcuch ma być zdefiniowany. Poniższe polecenie dodaje na przykład łańcuch `input` do zdefiniowanej uprzednio tabeli `filter`:

```
nft add chain filter input { type filter hook input priority 0 \; }
```

To polecenie powoduje dodanie łańcucha o nazwie `input` do tabeli o nazwie `filter`. Rodzaj łańcucha to podstawowy łańcuch `filter`, który zostanie dołączony do zaczepu `input` z priorytetem 0. Przy wprowadzaniu tego polecenia w wierszu poleceń należy dodać pomiędzy nawiasami pojedynczą spację oraz średnik. W przypadku stosowania polecenia w natywnym skrypcie `nft` można pominąć spację i lewy ukośnik.

Dodawanie łańcucha `output` wygląda podobnie, trzeba jedynie w odpowiednich miejscach zamienić `input` na `output`:

```
nft add chain filter output { type filter hook output priority 0 \; }
```

Składnia reguły

W regułach przeprowadzane są akcje filtrowania pakietów. Podczas pracy z regułami dostępne są trzy polecenia:

- `add` — dodanie reguły;
- `insert` — wstawienie reguły do łańcucha na początku lub w określonym miejscu;
- `delete` — usunięcie reguły.

W ramach reguły określane są kryteria dopasowywania dla pakietów oraz podjęta zostaje decyzja dotycząca tego, co powinno się zdarzyć z pakietem, który pasuje do danej reguły. Program `nftables` oraz budowane w nim reguły używają różnych instrukcji i wyrażeń do tworzenia definicji.

Instrukcje `nftables` są podobne do instrukcji `iptables` i zazwyczaj wpływają na sposób, w jaki pakiet będzie przetwarzany. Może to polegać na zatrzymaniu przetwarzania, wysłaniu przetwarzania do innego łańcucha lub po prostu na zarejestrowaniu pakietu. Instrukcje i werdykty mogą być następujące:

- `accept` — zaakceptowanie pakietu i zatrzymanie przetwarzania;
- `continue` — kontynuowanie przetwarzania pakietu;
- `drop` — zatrzymanie przetwarzania i porzucenie pakietu po cichu;
- `goto` — wysłanie przetwarzania do określonego łańcucha, ale niezwracanie go do łańcucha wywołującego;
- `jump` — wysłanie przetwarzania do określonego łańcucha i zwrócenie go do łańcucha wywołującego po zakończeniu operacji lub gdy zostanie wykonana instrukcja `return`;
- `limit` — przetwarzanie pakietu zgodnie z regułą, jeśli osiągnięty zostanie limit odebranych pasujących pakietów;
- `log` — zarejestrowanie pakietu i kontynuowanie przetwarzania;
- `queue` — zatrzymanie przetwarzania i wysłanie pakietu do procesu przestrzeni użytkownika;
- `reject` — zatrzymanie przetwarzania i odrzucenie pakietu;
- `return` — wysłanie przetwarzania z powrotem do łańcucha wywołującego.

Wyrażenia `nftables` mogą być charakterystyczne dla rodziny adresów lub rodzaju przetwarzanego pakietu. Program `nftables` używa wyrażeń bloku danych i metawyrażeń. Wyrażenia bloku danych (ang. *payload expressions*) są wyrażeniami zebranymi z informacji dotyczących pakietów. Istnieją na przykład określone wyrażenia nagłówka, takie jak `sport` i `dport` (odpowiednio port źródłowy i port docelowy), które mają zastosowanie do pakietów TCP i UDP, ale nie mają sensu w warstwach IPv4 i IPv6, ponieważ te warstwy nie używają portów. Metawyrażenia (ang. *meta expressions*) mogą być wykorzystywane

do reguł, które mają szerokie zastosowanie lub są związane z typowymi właściwościami pakietów lub interfejsów.

Dostępne metawyrażenia zostały przedstawione w tabeli 4.1.

Tabela 4.1. Metawyrażenia w nftables

Wyrażenie	Opis
`iif`	Indeks interfejsu, który odebrał pakiet.
`iifname`	Nazwa interfejsu, na którym został odebrany pakiet.
`iiftype`	Typ interfejsu, na którym został odebrany pakiet.
`length`	Długość pakietu w bajtach.
`mark`	Wartość `mark` pakietu.
`oif`	Indeks interfejsu, który wyśle pakiet.
`oifname`	Nazwa interfejsu, z którego pakiet zostanie wysłany.
`oiftype`	Typ interfejsu, z którego pakiet zostanie wysłany.
`priority`	Priorytet `tc` pakietu.
`protocol`	Protokół warstwy wyższej.
`rtclassid`	Obszar routingu dla pakietu.
`skgid`	Identyfikator grupy dla gniazda pochodzenia.
`skuid`	Identyfikator użytkownika dla gniazda pochodzenia.

Wyrażenia śledzenia połączenia (ang. *conntrack expressions*) wykorzystują metadane z pakietu w celu zapewnienia informacji do dalszego przetwarzania reguł. Wyrażenia śledzenia połączenia dodaje się za pomocą słowa kluczowego `ct`, po którym następuje jedna z poniższych opcji:

- `daddr`;
- `direction`;
- `expiration`;
- `helper`;
- `l3proto`;
- `mark`;
- `protocol`;
- `proto-src`;
- `proto-dst`;
- `saddr`;
- `state`;
- `status`.

Wyrażenie stanu (`state`) jest ważnym wyrażeniem dla zapory sieciowej. Normalna inspekcja pakietów i przetwarzanie reguł są bezstanowe, co oznacza, że przetwarzanie „nie wie nic" o wcześniej przetworzonym pakiecie. Każdy pakiet jest sprawdzany zgodnie z jego unikatowymi cechami charakterystycznymi, takimi jak adresy źródłowe i docelowe, porty i inne kryteria. Wymienione poniżej wyrażenia stanu umożliwiają zapisywanie informacji o pakiecie, tak aby reguła przetwarzania miała kontekst dotyczący trwającej wymiany powiązanego ruchu.

- `new` — nowy pakiet przybywający do zapory sieciowej, na przykład pakiet TCP z ustawioną flagą `SYN`;
- `established` — pakiet będący częścią połączenia, które jest już przetwarzane lub śledzone;
- `invalid` — pakiet, który nie jest zgodny z regułami protokołów;
- `related` — pakiet związany z połączeniem dla protokołu, który nie używa żadnych innych środków do śledzenia swojego stanu, na przykład protokół ICMP lub pasywny FTP;
- `untracked` — stan administracyjny wykorzystywany do omijania śledzenia połączenia, zazwyczaj stosowany tylko w szczególnych przypadkach.

W praktyce często używane są stany `new`, `related` i `established`, a stan `invalid` w razie potrzeby. Przedstawiona poniżej reguła dopuszcza na przykład połączenia SSH ustanowione (`established`) i powiązane (`related`). Dopuszczanie połączeń powiązanych jest istotne w przypadku, gdy pamięć stanu zostanie wyczyszczona, a tym samym zanegowane zostaną wszystkie ustanowione stany połączenia.

```
nft add rule filter input tcp dport 22 ct state established,related accept
```

Mechanizm stanu został omówiony szczegółowo w rozdziale 3., w podpunkcie „Rozszerzenie dopasowywania state tabeli filter".

Wyrażenia bloku danych są wykorzystywane do budowania reguł, które dopasowują określone szczególne kryteria i są ściśle związane z rodzajem przetwarzanego pakietu.

Wyrażenia dla nagłówków IPv4 zostały przedstawione w tabeli 4.2.

Tabela 4.2. Wyrażenia bloku danych dla IPv4

Wyrażenie	Opis
`checksum`	Suma kontrolna nagłówka IP.
`daddr`	Docelowy adres IP.
`frag-off`	Przesunięcie fragmentacji.
`hdrlength`	Długość nagłówka IP razem z opcjami.
`id`	Identyfikator IP.
`length`	Całkowita długość pakietu.

Tabela 4.2. Wyrażenia bloku danych dla IPv4 — *ciąg dalszy*

Wyrażenie	Opis
`Protocol`	Protokół warstwy wyższej.
`saddr`	Źródłowy adres IP.
`tos`	Wartość typu usługi (TOS).
`ttl`	Wartość czasu życia (TTL).
`version`	Wersja nagłówka IP. W wyrażeniach IPv4 zawsze wartość 4.

Wyrażenia dla nagłówków IPv6 zostały przedstawione w tabeli 4.3.

Tabela 4.3. Wyrażenia nagłówka IPv6

Wyrażenie	Opis
`daddr`	Docelowy adres IP.
`flowlabel`	Etykieta przepływu.
`hoplimit`	Limit przeskoków.
`length`	Długość bloku danych.
`nexthdr`	Protokół następnego nagłówka.
`priority`	Wartość priorytetu.
`saddr`	Źródłowy adres IP.
`version`	Wersja nagłówka IP. W wyrażeniach IPv6 zawsze wartość 6.

Wyrażenia dla nagłówków TCP zostały przedstawione w tabeli 4.4.

Tabela 4.4. Wyrażenia nagłówka TCP

Wyrażenie	Opis
`ackseq`	Numer potwierdzenia.
`checksum`	Suma kontrolna pakietu.
`doff`	Długość nagłówka.
`dport`	Port, dla którego przeznaczony jest pakiet.
`flags`	Flagi TCP.
`sequence`	Numer sekwencyjny.
`sport`	Port, z którego pochodzi pakiet.
`urgptr`	Wartość wskaźnika priorytetu.
`window`	Szerokość okna TCP.

UDP jest zasadniczo prostszym protokołem, istnieje więc mniej wyrażeń dla nagłówków UDP. Wyrażenia te zostały przedstawione w tabeli 4.5.

Tabela 4.5. Wyrażenia nagłówka UDP

Wyrażenie	Opis
`checksum`	Suma kontrolna pakietów.
`dport`	Port, do którego przeznaczony jest pakiet.
`length`	Całkowita długość pakietu.
`sport`	Port, z którego pochodzi pakiet.

Wyrażenia nagłówka dostępne dla ARP zostały przedstawione w tabeli 4.6.

Tabela 4.6. Wyrażenia nagłówka ARP

Wyrażenie	Opis
`hlen`	Długość adresu sprzętowego.
`htype`	Typ warstwy fizycznej.
`op`	Operacja.
`plen`	Długość protokołu warstwy wyższej.
`ptype`	Typ protokołu warstwy wyższej.

Podstawowe operacje nftables

Przy dodawaniu reguły określane są: tabela i łańcuch oraz kryteria dopasowywania. Dodanie na przykład reguły przyjmowania połączeń SSH z określonego hosta będzie wyglądać tak, jak przedstawiono poniżej. Ta reguła jest dodawana do poprzednio utworzonego łańcucha `input` tabeli `filter`:

```
nft add filter input tcp dport 22 accept
```

Poszczególne instrukcje, takie jak `accept`, `drop`, `reject` czy `log` (omówione w poprzednim punkcie), w `iptables` były nazywane **rozszerzeniami** (ang. *exstensions*). Wiele z tych samych opcji i trybów operacji, które działały dla tych rozszerzeń, działa również z `nftables`, na przykład do rejestrowania połączeń przychodzących używana jest instrukcja `log`. Ta instrukcja może być łączona ze śledzeniem połączenia w taki sposób, że rejestrowane będą tylko nowe połączenia z portem 22. Ponadto można dodać ograniczenie, aby mechanizm rejestrowania nie był przeciążany.

W `nftables` rejestrowanie wymaga modułów jądra `nfnetlink_log` lub `xt_LOG` albo wsparcia jądra dla tych modułów. Dodatkowo należy w katalogu *proc* włączyć rejestrowanie poprzez nadanie parametrowi `nf_log` wartości `"ipt_LOG"`:

```
echo "ipt_LOG" > /proc/sys/net/netfilter/nf_log/2
```

Ostateczne polecenie nftables służące do rejestrowania nowych połączeń SSH (z ograniczoną prędkością) wygląda następująco:

```
nft add filter input tcp dport 22 ct state new limit rate 3/second log
```

Jako dalsze selektory w ramach reguły używane są metawyrażenia, takie jak te służące do wybierania interfejsów wejściowego i wyjściowego. Polecenie używane na przykład do rejestrowania nowych połączeń przychodzących na interfejs eth0 będzie wyglądać tak:

```
nft add filter input iif eth0 ct state new limit rate 10/minute log
```

Reguły i opcje składni dla różnych wyrażeń zostały opisane w rozdziale 3.

Składnia plików nftables

Jedną z najlepszych cech nftables jest możliwość odczytu zewnętrznych plików zawierających reguły nftables. Te pliki umożliwiają importowanie zapisanych zestawów reguł i wykorzystywanie ich bez konieczności tworzenia długich i skomplikowanych skryptów powłoki. Mimo to skrypty powłoki nadal są pomocne jako główny kontener dla plików reguł zapory sieciowej, który importuje je w odpowiednim czasie.

Pliki są importowane poprzez dodanie do nftables opcji -f. Przedstawiony poniżej plik tworzy na przykład podstawową zaporę sieciową filtrującą pakiety, która rejestruje nowe pakiety SSH (z ograniczoną prędkością):

```
table filter {
        chain input {
                type filter hook input priority 0;
                tcp dport 22 ct state new limit rate 3/second log prefix "NEW packet: "
        }

        chain output {
                type filter hook input priority 0;
        }
}
```

Jeśli przyjmiemy, że plik został zapisany pod nazwą *firewall.nft*, można go załadować za pomocą następującego polecenia:

```
nft -f firewall.nft
```

Podsumowanie

Program nftables jest podobny do iptables pod tym względem, że reguły i opcje tych programów zazwyczaj dobrze się przekładają podczas budowania zapory sieciowej. Program nftables wykorzystuje tabele zawierające łańcuchy, które z kolei zawierają reguły. Reguły wskazują nftables, co zrobić z przetwarzanymi pakietami. Podobnie jak w przypadku iptables, program nftables może akceptować, porzucać, odrzucać i rejestrować pakiety oraz wykonywać na nich inne podobne działania. Program nftables

może również obejmować przetwarzanie oparte na stanie. Zastępuje programy `arptables`, `iptables` i `ebtables`.

Ponieważ wiele reguł i operacji z `nftables` jest podobnych do tych z `iptables`, w celu uzyskania informacji o wyrażeniach, które nie zostały bezpośrednio opisane w tym rozdziale, możesz odwołać się do rozdziału 3.

Skorowidz

F

H

I

J

K

L

Ł

M

N

O

R

S

Ś

T

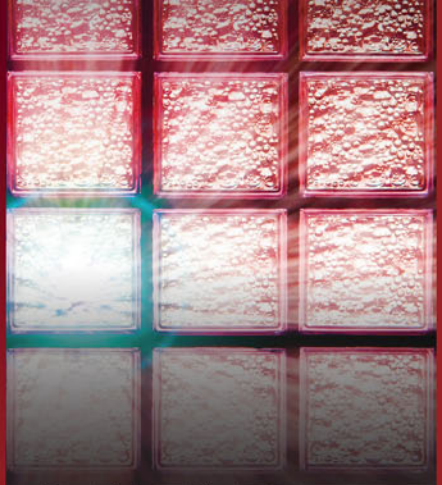
U

V

W

Z, Ż

ZAAWANSOWANE ZAPORY SIECIOWE DLA KAŻDEGO!



Zapora sieciowa (ang. *firewall*) to ostatni bastion chroniący Twój komputer przed atakiem. Dzięki odpowiedniej konfiguracji jesteś w stanie decydować, które pakiety mogą trafić do wnętrza sieci, a które nie. Możesz przesądzić o dostępie do określonych usług, zezwolić lub zabronić aplikacjom korzystać z dostępu do sieci Internet oraz ustalić limity na prędkość albo ilość przesłanych danych. Duże możliwości konfiguracji pozwalają na elastyczne podejście do tematu. Dzięki tej książce możesz wykorzystać wszystkie dostępne opcje!

Jednak na samym początku zapoznasz się z podstawowymi zasadami działania zapór sieciowych filtrujących pakiety. Przypomnisz sobie model sieciowy OSI, protokół IP, zasady routingu oraz sposób wyboru reguł filtrujących pakiety. Po tym wstępie przejdziesz do szczegółowego badania możliwości starszego narzędzia, jakim jest iptables. Składnia, omówienie dostępnych rozszerzeń dopasowywania, konfiguracja NAT — to tylko niektóre z poruszanych kwestii. Czasy się zmieniają i obecnie popularnością cieszy się narzędzie znane pod nazwą nftables. Dlatego z kolejnych rozdziałów dowiesz się, jak zbudować i uruchomić własną zaporę sieciową, korzystając właśnie z nftables. W następnych krokach uruchomisz usługi takie jak: e-mail, SSH, FTP oraz DHCP. Dalej zajmiesz się zagadnieniami związanymi z optymalizacją działania zapory oraz usługą NAT, czyli udostępnianiem łącza innym komputerom. W tej książce zostały poruszone niezwykle ważne kwestie łączące się z wykrywaniem włamań i najlepszymi sposobami reagowania na te incydenty. Ta książka jest doskonałą lekturą zarówno dla administratorów sieci, jak i dla pasjonatów systemu Linux!

STEVE SUEHRING — ma ponad 20-letnie doświadczenie w administrowaniu sieci i bezpieczeństwa systemów Linux. Redaktor do spraw bezpieczeństwa w magazynie „LinuxWorld". Konsultant, architekt technologii, autorytet w branży IT. Autor książek *JavaScript krok po kroku* oraz *MySQL Bible*.

Dzięki tej książce:

Poznasz podstawowe koncepcje związane z zaporami sieciowymi

Zaznajomisz się z narzędziem iptables

Skonfigurujesz i uruchomisz zaawansowaną zaporę dzięki nftables

Zoptymalizujesz działanie zapory sieciowej

Udostępnisz łącze innym urządzeniom w Twojej sieci

Prawidłowo zareagujesz na incydenty

W pełni wykorzystasz możliwości iptables oraz nftables!

Helion

37633 numer katalogowy

księgarnia internetowa

http://helion.pl

zamówienia telefoniczne

0 801 339900

0 601 339900

Informatyka w najlepszym wydaniu

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

Addison-Wesley

ISBN 978-83-283-1297-5

cena: 69,00 zł

sięgnij po WIĘCEJ
KOD KORZYŚCI

ALWAYS LEARNING

PEARSON