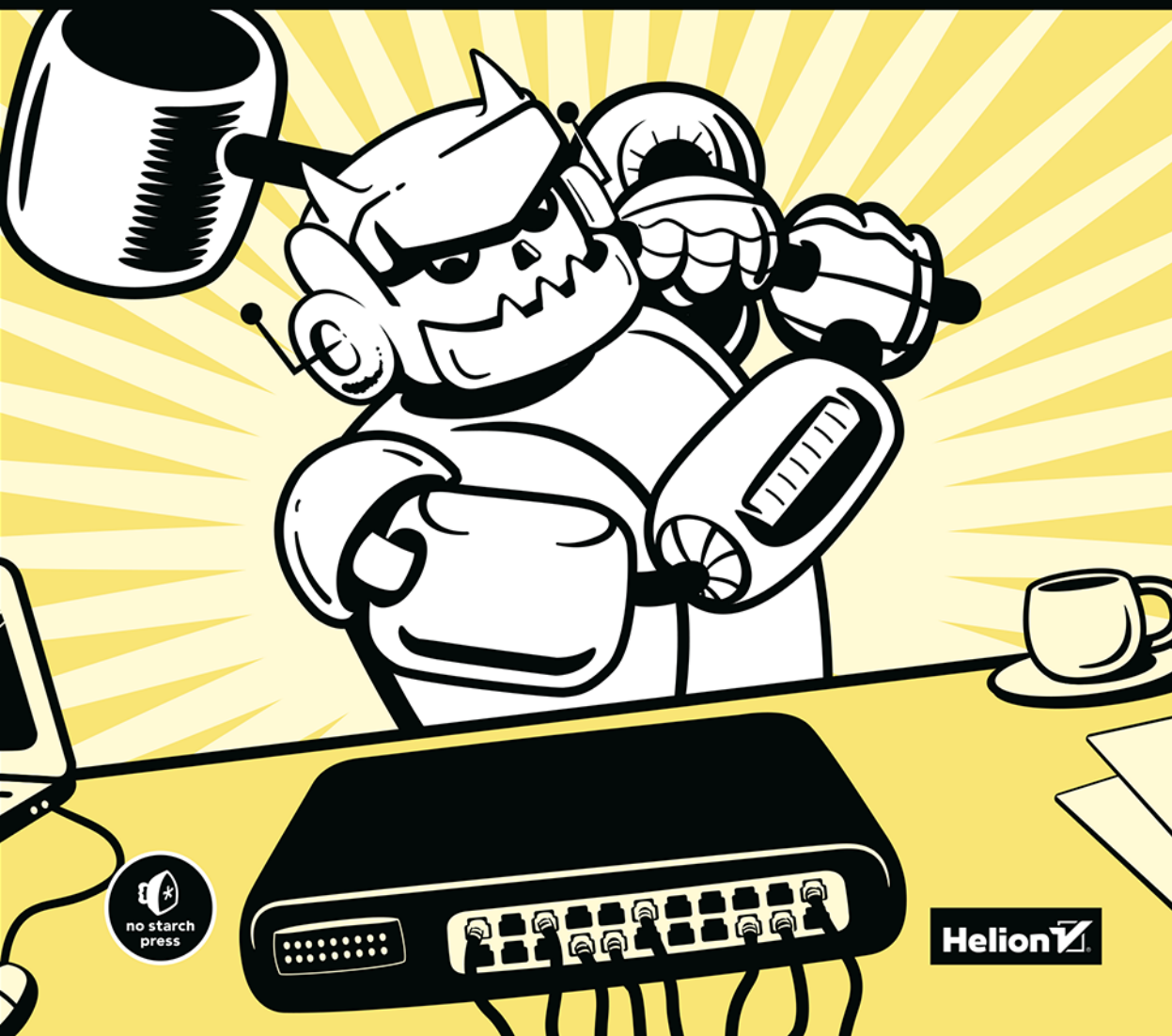
CYBERBEZPIECZEŃSTWO W MAŁYCH SIECIACH

PRAKTYCZNY PRZEWODNIK
DLA UMIARKOWANYCH PARANOIKÓW

SETH ENOKA



no starch press

Helion

Tytuł oryginału: Cybersecurity for Small Networks: A No-Nonsense Guide for the Reasonably Paranoid

Tłumaczenie: Grzegorz Werner

ISBN: 978-83-289-0428-6

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

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

3

Filtrowanie ruchu sieciowego za pomocą zapór

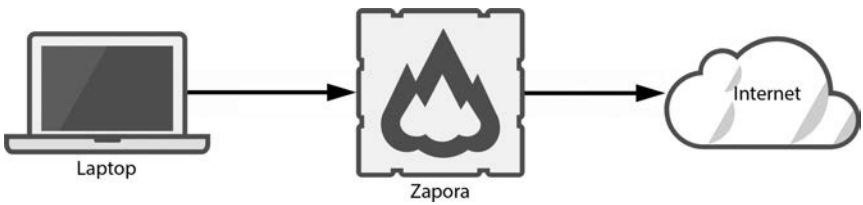
Zapora monitoruje i filtruje przychodzący i wychodzący ruch sieciowy. Powszechne jest błędne przekonanie, że zapora jest zawsze ostatnią linią obrony; w rzeczywistości graniczna zapora powinna być pierwszą przeszkodą, na którą napotykają przeciwnicy próbujący przeniknąć do dowolnej sieci, dużej czy małej. Za każdym razem, kiedy przeglądarka internetowa uzyskuje dostęp do witryny internetowej, komunikator wysyła wiadomość albo klient poczty wysyła bądź odbiera wiadomość e-mail, generowany ruch powinien przechodzić przez co najmniej jedną zaporę.

W tym rozdziale poznasz dwa rozwiązania zaporowe: iptables i pfSense. W Linuksie popularne jest rozwiązanie iptables, często używane jako **zapora hosta** (to znaczy zapora, która przepuszcza lub odrzuca ruch w określonym punkcie końcowym). Rozwiązanie pfSense, które można zaimplementować programowo z wykorzystaniem oprogramowania open source lub sprzętowo za pomocą urządzeń sprzedawanych przez Netgate, wykorzystuje się jako zaporę obwodową lub graniczną odpowiedzialną za filtrowanie ruchu w całych sieciach lub segmentach sieci.

Typy zapór

Zaporę sprzętową można umieścić w sieci fizycznie i logicznie. **Zapora programowa**, instalowana jako aplikacja w punkcie końcowym, wymaga więcej konfiguracji samej zapory i podłączonych do niej urządzeń, jeśli ma efektywnie filtrować ruch. Używając jednej lub obu tych zapór, możesz skutecznie ograniczyć **powierzchnię ataku** złożoną z punktów, w których przeciwnik może przeniknąć do Twojej sieci, naruszyć jej zabezpieczenia albo wykorzystać jej zasoby. Należy dążyć do tego, aby powierzchnia ataku była jak najmniejsza.

Zapora obwodowa lub **graniczna**, instalowana między Twoją siecią prywatną a innymi sieciami, takimi jak internet, może być albo programowa, albo sprzętowa. Zapory obwodowe umieszcza się na fizycznych i logicznych granicach sieci, więc są pierwszym punktem, do którego trafia ruch z publicznego internetu do Twojej sieci wewnętrznej, i ostatnim punktem w Twojej sieci, przez który przechodzi ruch zmierzający do internetu, jak pokazano na rysunku 3.1.



Rysunek 3.1. Zapora obwodowa

Zapory przepuszczają lub odrzucają (blokują) ruch na podstawie skonfigurowanej **listy reguł**. Sposób, w jaki reguły te są stosowane do ruchu, zależy od typu zapory. Najczęściej spotykany typ, **zapora filtrująca pakiety**, porównuje ze swoim zestawem reguł każdy pakiet danych, który próbuje dostać się do sieci wewnętrznej (lub z niej wydostać). Jeśli zawartość pakietu pasuje do jakiejś reguły, zapora przepuści lub zablokuje ruch, w zależności od tego, co wskazuje reguła.

Istnieją też zapory stanowe i bezstanowe. **Zapora stanowa** śledzi wszystkie połączenia przychodzące oraz wychodzące i monitoruje każde z nich jako unikatową „konwersację" między dwoma punktami końcowymi. Zapewnia to zaporze informacje kontekstowe o danym połączeniu i umożliwia precyzyjniejszą kontrolę ruchu. Natomiast **zapory bezstanowe** nie przechowują informacji o poszczególnych połączeniach. Zarówno iptables, jak i pfSense to zapory stanowe.

Niemal wszystkie systemy operacyjne mają wbudowaną zaporę programową, znaną jako **zapora oparta na hoście**, która filtruje ruch specyficzny dla tego hosta. Większość urządzeń Windows i Mac jest sprzedawanych z gotową zaporą opartą na hoście, której zestaw reguł jest adekwatny, choć nie wyczerpujący. Zapora ta sprawdza się w zwykłym, codziennym użytkowaniu — użytkownicy nie muszą sami konfigurować zapory, co oszczędza im kłopotów i ogranicza wsparcie techniczne, jakie muszą zapewniać producenci komputerów. W urządzeniach linuksowych trzeba samodzielnie skonfigurować zaporę — dowiesz się, jak to zrobić, w następnym podrozdziale.

Najlepiej używać zarówno zapory w hoście, jak i zapory obwodowej i prawidłowo je skonfigurować, żeby dodać do sieci wiele warstw obrony.

iptables

Linuksowe narzędzie iptables oferuje niezwykłą elastyczność filtrowania ruchu, który wchodzi do sieci, przechodzi przez nią lub opuszcza ją. Zapora organizuje swoje reguły w **łańcuchach zasad**, listach reguł, które analizują i dopasowują pakiety na podstawie ich zawartości. Każda reguła określa, co ma zrobić zapora z pakietem pasującym do definicji — może przepuścić pakiet, odrzucić go lub porzucić. Kiedy pakiet jest przepuszczany, przechodzi przez zaporę bez przeszkód. Kiedy jest porzucany, zapora blokuje go i nie wysyła żadnej odpowiedzi do nadawcy. Jeśli pakiet jest odrzucany, zapora blokuje go i odsyła do nadawcy komunikat o odrzuceniu, zapewniając informacje kontekstowe o Twojej sieci i używanej zaporze.

Istnieją trzy główne typy łańcuchów zasad: **łańcuchy wejściowe**, **łańcuchy wyjściowe** i **łańcuchy przekazywania**. Łańcuchy wejściowe określają, czy należy przepuszczać do sieci ruch z zewnętrznego źródła, na przykład połączenie **wirtualnej sieci prywatnej** (ang. *virtual private network*, VPN) ze zdalnej lokalizacji. VPN to sposób na logiczne — w przeciwieństwie do fizycznego — łączenie odrębnych sieci, zwykle w celu zapewnienia zdalnego dostępu z jednej sieci do drugiej. Sieci VPN omówimy dokładniej w rozdziale 5.

Łańcuchy wyjściowe określają, czy zapora powinna przepuszczać określony ruch wychodzący do sieci zewnętrznej. Weźmy na przykład protokół **Internet Control Message Protocol** (**ICMP**), którego używa się głównie do diagnozowania problemów z komunikacją sieciową. Pakiety „ping" protokołu ICMP to ruch wychodzący, który przechodzi przez łańcuch wyjściowy. Są to zapytania wysyłane przez jedno urządzenie do drugiego, zwykle w celu sprawdzenia, czy możliwe jest nawiązanie połączenia między nimi. Pakiety „ping" podróżują z Twojego urządzenia przez zaporę i kilka innych urządzeń w publicznym internecie, zanim ostatecznie trafią do celu. Jeśli Twój łańcuch wyjściowy blokuje ruch ICMP, Twoje urządzenie nie będzie mogło niczego „pingować", ponieważ zapora odrzuci lub porzuci te pakiety.

W większości przypadków reguły zapory stanowej powinny zezwalać zarówno na nowe, jak i ustanowione połączenia. Jeśli na przykład utworzysz łańcuch wyjściowy, który pozwala Twojemu urządzeniu „pingować" serwery Google'a, musisz poinstruować zaporę, aby zezwalała na ruch przychodzący związany z ustanowionymi połączeniami. W przeciwnym razie Twoje urządzenie wyśle „ping" do Google'a, który przejdzie przez Twoją zaporę, ale odpowiedź Google'a zostanie przez nią zablokowana.

Łańcuchy przekazywania przekazują ruch odbierany przez zaporę do innej sieci. W małej sieci biurowej lub domowej zapory oparte na hoście rzadko używają łańcuchów przekazywania, chyba że zapora jest skonfigurowana jako router. Zapora obwodowa używałaby łańcucha przekazywania do przekierowywania ruchu z sieci wewnętrznej do zewnętrznej albo z jednego segmentu sieci do drugiego, zapewne

z wykorzystaniem translacji adresów sieciowych (NAT) omówionej w rozdziale 1. Jednakże konfiguracja tego typu jest zbyt skomplikowana dla małych sieci i pasuje bardziej do sieci korporacyjnej.

Wykorzystując te łańcuchy zasad, można bardzo precyzyjnie sterować ruchem przechodzącym przez sieć. W kolejnych rozdziałach skonfigurujesz kilka serwerów linuksowych, a każdemu z nich przydałaby się zapora oparta na hoście. Zalecam skonfigurować iptables w każdym z tych serwerów według poniższej instrukcji.

UWAGA *Narzędzie iptables nie pozwala zabezpieczyć sieci i ruchu IPv6. Jeśli planujesz używać IPv6 w swojej sieci, oprócz iptables będziesz musiał użyć narzędzia ip6tables. Jeśli nie masz dobrego powodu, żeby używać protokołu IPv6 w swojej sieci, radzę wyłączyć go całkowicie. Zostanie to opisane w rozdziale 4.*

(12) Instalacja iptables

Jeśli zbudowałeś już standardowy serwer Ubuntu według instrukcji z rozdziału 1., możesz przystąpić do konfigurowania zapory iptables. Po opanowaniu podstaw będziesz mógł wykorzystać nabytą wiedzę do skonfigurowania iptables we wszystkich linuksowych punktach końcowych. Jeśli jeszcze tego nie zrobiłeś, utwórz teraz system Ubuntu.

W nowszych wersjach Ubuntu zapora iptables powinna być zainstalowana domyślnie, więc zaloguj się przez SSH jako zwykły użytkownik i sprawdź, czy tak jest w istocie, wydając polecenie:

```
$ sudo iptables -V
[sudo] password for użytkownik:
iptables v1.8.7 (nf_tables)
```

Jeśli zapora iptables jest zainstalowana, serwer powinien wyświetlić informacje o wersji, jak pokazano powyżej. Twoja wersja może być inna.

Jeśli zapora iptables nie jest zainstalowana, otrzymasz komunikat o błędzie. W takim przypadku zainstaluj iptables:

```
$ sudo apt install iptables
```

Po instalacji sprawdź wersję w pokazany wyżej sposób, aby upewnić się, że zapora została zainstalowana pomyślnie.

Następnie zainstaluj `iptables-persistent`, narzędzie, które pozwala zapisywać konfiguracje zapory i automatycznie wczytywać je po ponownym uruchomieniu serwera:

```
$ sudo apt install iptables-persistent
```

W oknie terminala powinien się pojawić kreator instalacji. Zobaczysz nazwę pliku, w którym Twój serwer zapisze reguły zapory (domyślnie jest to */etc/iptables/rules.v4*), i zostaniesz poinformowany, że reguły z tego pliku będą wczytywane automatycznie podczas rozruchu systemu. Zauważ, że po zakończeniu procesu instalacji przyszłe zmiany reguł zapory trzeba będzie zapisywać ręcznie. Wybierz opcję *Yes*, aby zapisać bieżące reguły zapory. Jeśli nie zainstalujesz tego komponentu, będziesz musiał od nowa konfigurować zaporę po każdym ponownym jej uruchomieniu.

Możesz teraz sprawdzić bieżące łańcuchy zasad w następujący sposób:

```
$ sudo iptables -L
[sudo] password for użytkownik:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Fraza `policy ACCEPT` w wynikach programu wskazuje, że domyślnie zapora akceptuje cały ruch wejściowy, wyjściowy i przekazywany. Te domyślne ustawienia są pożądane, ponieważ oznaczają, że system będzie mieć dostęp do sieci bez konieczności konfigurowania zapory przez użytkownika. Rozwiązanie to nie jest jednak bezpieczne, więc zmodyfikujmy je.

Reguły zapory iptables

Podczas tworzenia reguł iptables pamiętaj, że kolejność ma znaczenie. Kiedy ruch dociera do Twojej zapory, sprawdza ona reguły jedną po drugiej *w kolejności, w jakiej występują*. W razie dopasowania ruchu do reguły zapora nie sprawdza kolejnych — jeśli pierwsza reguła na Twojej liczącej 50 pozycji liście odrzuca cały ruch, zapora zinterpretuje tę regułę, odrzuci ruch i zakończy przetwarzanie, co będzie oznaczać faktyczną izolację urządzenia. I odwrotnie, jeśli masz tę samą listę 50 reguł, ale pierwsza reguła przepuszcza cały ruch, to cały ruch będzie przechodził przez zaporę. Powinieneś unikać obu tych sytuacji.

Aby zrozumieć, jak konstruuje się reguły zapory iptables, spójrz na poniższy przykład:

```
$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack \
--ctstate NEW,ESTABLISHED -j ACCEPT
```

Zaraz po poleceniu `sudo` wywoływany jest program `iptables` w celu zdefiniowania reguły. Następny argument określa, czy reguła zostanie dołączona na końcu określonego łańcucha zasad (`-A` od ang. *append*), usunięta z niego (`-D` od ang. *delete*) czy też wstawiona do niego (`-I` od ang. *insert*). Możesz też użyć w tym miejscu

opcji `-R` (od ang. *replace*), aby zastąpić lub zaktualizować istniejącą regułę. Słowo `INPUT` wskazuje, że reguła dotyczy łańcucha wejściowego. Możesz również wskazać łańcuch wyjściowy (`OUTPUT`), łańcuch przekazywania (`FORWARD`) lub inne łańcuchy zasad.

W większości przypadków trzeba poinformować iptables, jakiego protokołu i portu dotyczy reguła. W powyższym przykładzie `-p tcp` wskazuje, że reguła ma być stosowana tylko do ruchu TCP, a `--dport 22` informuje, że reguła dotyczy pakietów z portem docelowym numer 22. Oba te ustawienia są opcjonalne. Możesz określić wiele portów za pomocą następującej składni: `--match multiport --dports` *`port1,port2,port3`*.

UWAGA ***Transmission Control Protocol (TCP)*** *to niezawodny protokół transmisji, zaprojektowany tak, aby zapewnić pomyślne dostarczenie pakietu przez sieć. Jeśli dojdzie do utraty pakietu podczas komunikacji TCP, utracone pakiety zostaną przesłane ponownie, dzięki czemu wszystkie wysłane dane zostaną ostatecznie odebrane przez docelowy host.* ***User Datagram Protocol (UDP)*** *to protokół zawodny, który nie gwarantuje pomyślnego dostarczenia danych i nie przesyła ponownie pakietów. Protokołu TCP używa się, kiedy niezawodność ma znaczenie i konieczne jest dostarczenie każdego pakietu.*

Zapora iptables oferuje wiele modułów dopasowujących, a żądany moduł można określić za pomocą argumentu `-m`. W tym przykładzie użyto modułu `conntrack`, który umożliwia stanową inspekcję pakietów (to ustawienie również jest opcjonalne). Niektóre inne moduły to `connbytes` do tworzenia reguł na podstawie ilości przesłanych danych oraz `connrate` do dopasowywania ruchu na podstawie tempa transmisji danych. Więcej informacji można znaleźć na stronie `man` programu iptables: *https://linux.die.net/man/8/iptables/*.

Opcja `--ctstate` nakazuje zaporze iptables śledzić ruch związany z określonymi typami połączeń — w tym przypadku nowymi (`NEW`) i ustanowionymi (`ESTABLISHED`). Dostępnych jest wiele wartości stanu połączenia, ale najczęściej używane to `NEW`, `ESTABLISHED`, `RELATED` i `INVALID`. Pierwsze dwie wartości są oczywiste; pakiety są częścią nowego albo ustanowionego przepływu ruchu. Pakiety powiązane (`RELATED`) niekoniecznie pasują do ustawionego połączenia, ale są oczekiwane przez zaporę, ponieważ wymaga ich istniejące połączenie (tzn. są oczekiwane ze względu na istniejący kontekst zapory). Pakiety nieprawidłowe (`INVALID`) to takie, które nie spełniają kryteriów żadnego innego stanu.

Wreszcie opcja `-j` (od ang. *jump*) i wszystko, co po niej następuje, określa działanie, do którego należy przeskoczyć (czyli które należy podjąć) w razie dopasowania tej reguły. Najczęściej jest to działanie `ACCEPT`, które przepuszcza ruch pasujący do tej reguły; `DROP` lub `REJECT`, które porzuca lub odrzuca ruch; albo `LOG`, które rejestruje ruch w pliku dziennika (więcej na ten temat powiemy później).

Teraz, kiedy znasz podstawy reguł `iptables`, możesz skonfigurować swoją zaporę tak, aby przepuszczała i blokowała odpowiednie rodzaje ruchu.

Konfigurowanie iptables

Podczas konfigurowania `iptables` najpierw dodaj reguły, które porzucają nieprawidłowy ruch:

```
$ sudo iptables -A OUTPUT -m state --state INVALID -j DROP
$ sudo iptables -A INPUT -m state --state INVALID -j DROP
```

Następnie dodaj reguły, które akceptują połączenia powiązane i ustanowione, a także połączenia z adresem pętli zwrotnej, aby uniknąć ewentualnych późniejszych problemów (**adres pętli zwrotnej**, ang. *loopback*, to wewnętrzny adres, którego komputery używają do testowania i diagnozowania problemów z siecią):

```
$ sudo iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
$ sudo iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
$ sudo iptables -A INPUT -i lo -j ACCEPT
```

Dzięki temu zapora będzie akceptować ruch dopasowany do znanego połączenia albo powiązany z istniejącym połączeniem i porzucać nieoczekiwane pakiety (co może ochronić Twoją sieć przed nieproszonym lub złośliwym skanowaniem).

Kiedy wydasz te polecenia, aby wprowadzić reguły do łańcuchów zasad, ponownie wydaj polecenie listowania, aby upewnić się, że reguły zostały zaakceptowane:

```
$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  -- anywhere             anywhere            state INVALID
ACCEPT     all  -- anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     all  -- anywhere             anywhere
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere            anywhere            state INVALID
ACCEPT     all  --  anywhere            anywhere            state RELATED,ESTABLISHED
```

Zauważ, że reguły zostały dodane do łańcuchów `INPUT` i `OUTPUT`. Łańcuch `FORWARD` pozostaje pusty.

Następnie upewnij się, że Twoja zapora będzie przepuszczać ruch SSH. Możesz to zrobić na dwa sposoby: ogólnie zezwalając na ruch SSH albo akceptując SSH tylko od określonego podzbioru urządzeń w Twojej sieci. Aby przepuszczać do swojej sieci ruch SSH pochodzący od wszystkich urządzeń w Twojej sieci, użyj poniższego polecenia:

```
$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -j \
ACCEPT
```

Tworzenie takich ogólnych reguł może być pomocne, kiedy nawiązujesz połączenia SSH między wieloma urządzeniami w swojej sieci. Jednakże zezwalanie na nieskrępowane użycie programów i pozostawianie całkowicie otwartych portów nie jest najbezpieczniejszym rozwiązaniem. Usługi takie jak SSH powinny być dostępne tylko dla określonych adresów lub zakresów adresów IP, ponieważ zezwolenie na zdalny dostęp lub transfery plików między Twoimi punktami końcowymi a dowolnymi innymi urządzeniami jest ryzykowne.

Możesz zmniejszyć swoją powierzchnię ataku, określając za pomocą opcji `-s` *`źródło`* źródłowy adres IP lub zakres adresów (na przykład 192.168.1.25) w swoim łańcuchu wejściowym, więc jeśli konfigurujesz iptables w maszynie wirtualnej, możesz zezwolić na połączenia administracyjne tylko z jednego hosta i odmawiać dostępu wszystkim innym punktom końcowym w swojej sieci:

```
$ sudo iptables -A INPUT -p tcp -s 192.168.1.25 --dport 22 -m \
conntrack --ctstate NEW -j ACCEPT
```

Dołączamy tę regułę do łańcucha zasad `INPUT`, używając opcji `-A`, portu docelowego 22 i protokołu TCP. W przypadku nowych połączeń (`NEW`) iptables zaakceptuje (`ACCEPT`) ruch pasujący do tej reguły. Port możesz wybrać sam; upewnij się tylko, że Twoja konfiguracja SSH odpowiada regule zapory. Jeśli reguła zezwala na ruch SSH do portu 22, a Twoja konfiguracja SSH przyjmuje połączenia w porcie 2222, zapora zablokuje Twoje połączenie SSH.

Jeśli się pomylisz, usuń regułę, wydając to samo polecenie, ale z opcją `-D` w miejscu `-A`:

```
$ sudo iptables -D INPUT -p tcp -s 192.168.1.25 --dport 22 -m conntrack \
--ctstate NEW,ESTABLISHED -j ACCEPT
```

Możesz też usunąć wszystkie reguły określone dla danego łańcucha zasad, korzystając z opcji `-F` *`łańcuch`* albo `--flush` *`łańcuch`*:

```
$ sudo iptables -F INPUT
```

Dysponując tym podstawowym zestawem reguł, możesz teraz poinstruować iptables, co należy robić z całym pozostałym ruchem (który nie powinien wchodzić do Twojego serwera lub sieci ani ich opuszczać). Kiedy utworzysz reguły zezwalające na konkretny ruch, który zapora powinna akceptować, możesz prawdopodobnie blokować lub odrzucać wszystkie pozostałe pakiety. Zrób to po skonfigurowaniu reguł zapory; w przeciwnym razie może się zdarzyć, że przerwiesz swoje połączenie z serwerem i nie będziesz mógł połączyć się ponownie za pośrednictwem SSH. Argument `-P` ustawia domyślne działanie łańcuchów zasad i informuje `iptables`, co należy robić z całym ruchem niepasującym do zdefiniowanych reguł. Ustaw zatem łańcuchy zasad tak, aby domyślnie porzucały (`DROP`) taki ruch:

```
$ sudo iptables -P INPUT DROP
$ sudo iptables -P FORWARD DROP
$ sudo iptables -P OUTPUT DROP
```

Opcja `-P` wykorzystana w taki sposób różni się od poprzednio używanych opcji `-A` i `-I`, ponieważ nie wpływa na same reguły zapory; zamiast tego dotyczy ogólnych zasad, które regulują ruch w Twojej sieci. Podczas gdy opcje `-A` i `-I` odpowiednio dołączają lub wstawiają reguły zapory, opcja `-P` konfiguruje działanie zapory o jeden poziom wyżej.

W tym momencie sprawdzenie łańcuchów iptables powinno dać następujący wynik:

```
Chain INPUT (policy DROP)
target     prot opt source           destination
DROP       all  --  anywhere         anywhere          state INVALID
ACCEPT     all  --  anywhere         anywhere          state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere         anywhere
ACCEPT     tcp  --  192.168.1.25     anywhere          tcp dpt:22 ctstate NEW
Chain FORWARD (policy DROP)
target     prot opt source           destination
Chain OUTPUT (policy DROP)
target     prot opt source           destination
DROP       all  --  anywhere         anywhere          state INVALID
ACCEPT     all  --  anywhere         anywhere          state RELATED,ESTABLISHED
```

Zauważ, że zasady dla wszystkich trzech łańcuchów zmieniły się z `ACCEPT` na `DROP`, co wskazuje, że domyślnym działaniem każdego łańcucha jest porzucanie ruchu niepasującego do żadnej z utworzonych reguł. Powinieneś również móc zidentyfikować reguły, które dodałeś do łańcuchów, porównując powyższe wyniki do poprzedniego listingu reguł iptables. Możesz otrzymać komunikat o błędzie DNS, ponieważ obecnie zapora blokuje wszystko, co nie jest jawnie dozwolone, w tym usługi DNS (które działają w porcie 53). Rozwiąż ten problem poprzez dodanie dwóch poniższych reguł:

```
$ sudo iptables -A OUTPUT -p udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
$ sudo iptables -A OUTPUT -p tcp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
```

Polecenia te dołączają reguły do łańcucha wyjściowego, pozwalając temu serwerowi zgłaszać wychodzące żądania rozwikłania nazwy poprzez port 53 protokołów UDP i TCP. Po dodaniu tych reguł serwer może rozwikływać nazwy domenowe.

Przetestuj zaporę, „pingując" serwer z innego urządzenia w Twojej sieci; powinieneś otrzymać komunikat o błędzie, ponieważ zapora nie przepuszcza ruchu ICMP. Jeśli wydasz polecenie `ping` na samym serwerze, powinien się pojawić podobny błąd:

```
$ ping google.com -c 5
PING google.com (<adres_ip >): 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
--- google.com ping statistics --
5 packets transmitted, 0 received, 100% packet loss, time 4000ms
```

ICMP bywa tak przydatnym narzędziem diagnostycznym, że możesz postanowić przepuszczać pakiety ping przez swoją zaporę iptables. W tym celu dodaj następujące reguły:

```
$ sudo iptables -A INPUT -p icmp -j ACCEPT
$ sudo iptables -A OUTPUT -p icmp -j ACCEPT
```

Może się okazać, że konieczne jest otworzenie dodatkowych portów w zaporze. Jeśli masz na przykład zainstalowany serwer proxy albo jeśli zbudujesz go po przeczytaniu rozdziału 6., będziesz musiał otworzyć port proxy (3128) w zaporze:

```
$ sudo iptables -A OUTPUT -p tcp --dport 3128 -m conntrack --ctstate NEW -j ACCEPT
```

W większości przypadków powinieneś blokować przeglądanie stron internetowych z serwerów — niewiele jest uzasadnionych powodów dla takiej aktywności. Zarówno z punktu zarządzania, jak i bezpieczeństwa serwery powinny być systemami jednozadaniowymi. Zezwolenie na jakąkolwiek dodatkową usługę w serwerze — zwłaszcza przeglądanie stron internetowych — powiększa powierzchnię ataku i stwarza potencjalne słabe punkty w Twojej sieci.

Jeśli postanowisz zezwolić na taki ruch z Twojego serwera, aby mógł on na przykład pobierać aktualizacje oprogramowania, utwórz reguły wyjściowe dla portów 80 i 443, domyślnych portów protokołów (odpowiednio) HTTP i HTTPS:

```
$ sudo iptables -A OUTPUT -p tcp --dport 80 -m conntrack \
--ctstate NEW,ESTABLISHED -j ACCEPT
$ sudo iptables -A OUTPUT -p tcp --dport 443 -m conntrack \
--ctstate NEW,ESTABLISHED -j ACCEPT
```

Reguły dla HTTP i HTTPS różnią się tylko numerem portu.

Za każdym razem, kiedy dodajesz regułę, powinieneś ją przetestować. W tym przypadku najprościej byłoby sprawdzić, czy możesz otwierać strony internetowe w przeglądarce w serwerze (jeśli masz zainstalowany interfejs graficzny) albo w programie `curl` w terminalu `bash`. Najpierw zainstaluj `curl`:

```
$ sudo apt install curl
```

Jeśli nie masz reguł zezwalających na ruch HTTP i HTTPS, polecenie instalacji zawiedzie, ponieważ aktualizacje są zwykle przeprowadzane za pośrednictwem HTTP. Jeśli masz te reguły, `curl` powinien zainstalować się pomyślnie, więc sprawdź, czy masz otwarte porty 80 i 433:

```
$ curl http://icanhazip.com
adres_ip
```

Pod adresem *http://icanhazip.com/* znajduje się internetowa usługa, która zwraca Twój bieżący publiczny adres IP. Jeśli na ekranie wyświetli się Twój publiczny adres IP, oznacza to, że Twoja zapora jest skonfigurowana poprawnie.

Jeśli otrzymasz komunikat o błędzie, problemem może być jedna z reguł. Sprawdź, czy nie zrobiłeś literówki, a jeśli to nie pomoże, usuń reguły za pomocą opcji `-D` lub `-F`, jak wyjaśniono wcześniej, i zacznij od nowa. Po poprawnym skonfigurowaniu zapory możesz dodać dalsze reguły, jeśli uznasz to za stosowne.

Szczególnym przypadkiem są reguły, które blokują ruch do konkretnych adresów IP. Ponieważ jednak większość publicznych witryn może mieć wiele adresów IP, blokowanie witryny z wykorzystaniem iptables nie jest najlepszym rozwiązaniem, ponieważ musiałbyś utworzyć oddzielną regułę dla każdego unikatowego adresu IP. W większości przypadków lepiej użyć serwera proxy, co omówimy w rozdziale 6.

Gdybyś chciał użyć iptables do blokowania witryn — na przykład do blokowania ruchu do witryny i z witryny *https://www.squirreldirectory.com/*, która obecnie ma adres IP 206.189.69.35 — dodałbyś dwie poniższe reguły do łańcuchów `INPUT` i `OUTPUT`:

```
$ sudo iptables -A INPUT -s 206.189.69.35 -j DROP
$ sudo iptables -A OUTPUT -s 206.189.69.35 -j DROP
```

Zwykle regułę tego rodzaju dodaje się w celu przepuszczania lub blokowania ruchu ze statycznego, prywatnego adresu IP, który nie będzie się zmieniać, i używa serwera proxy dla publicznych adresów IP lub URL.

Rejestrowanie działania iptables

Zainstalowałeś i skonfigurowałeś zaporę iptables, ale nie nakazałeś jej niczego rejestrować, więc nie dokumentuje ona swoich działań, co może utrudniać diagnozowanie problemów albo ustalanie, czy zablokowany ruch rzeczywiście powinien być blokowany.

Najpierw utwórz nowy, niestandardowy łańcuch zasad. Zauważ, że w poniższej konfiguracji kolejność reguł ma kluczowe znaczenie. Możesz nadać łańcuchowi dowolną nazwę; w tym przykładzie nazwiemy go `LOGGING`:

```
$ sudo iptables -N LOGGING
```

Parametr `-N` służy do tworzenia nowych łańcuchów.

Następnie dodaj na końcu łańcuchów `INPUT` i `OUTPUT` regułę, która nakazuje iptables wysyłać niedopasowany dotychczas ruch do łańcucha `LOGGING`:

```
$ sudo iptables -A INPUT -j LOGGING
$ sudo iptables -A OUTPUT -j LOGGING
```

W dalszej kolejności nakaż iptables rejestrować każdy typ porzuconego pakietu tylko raz na minutę:

```
$ sudo iptables -A LOGGING -m limit --limit 1/minute -j LOG \
--log-prefix "FW-Dropped: " --log-level 4
```

Limit ten jest opcjonalny i możesz ustawić go na dowolny okres, na przykład `1/second`, `1/minute`, `1/hour` lub `1/day`. Ograniczenie liczby wpisów w dzienniku zmniejsza zarówno „szum", jak i rozmiar plików dziennika. Dodaj przedrostek (`"FW-Dropped: "`) do rejestrowanych informacji, aby móc łatwo identyfikować wpisy zapory w dzienniku. Ustawienie poziomu rejestrowania (`log-level`) na 4 spowoduje rejestrowanie zdarzeń do poziomu „ostrzeżenia", czyli takich, które mogą mieć istotny wpływ na serwer lub zaporę. Zwiększenie tej liczby sprawi, że rejestrowanych będzie więcej zdarzeń o niższych poziomach krytyczności, co bywa przydatne podczas rozwiązywania problemów. Poziomy rejestrowania od 1 do 3 powodują rejestrowanie tylko zdarzeń lub błędów o krytyczności wyższej niż „ostrzeżenie".

Wreszcie, poniższe polecenie instruuje zaporę, że po zarejestrowaniu pakiety powinny zostać porzucone:

```
$ sudo iptables -A LOGGING -j DROP
```

Twoja zapora będzie teraz rejestrować porzucone pakiety zarówno przychodzące, jak i wychodzące z serwera. Domyślnie dziennik będzie przechowywany w pliku */var/log/messages*.

Ostatnim etapem jest zapisanie konfiguracji zapory. Pamiętaj, że konfiguracje iptables domyślnie są tymczasowe i znikają po ponownym uruchomieniu komputera; właśnie dlatego zainstalowaliśmy pakiet `iptables-persistent` w projekcie 12. Aby zapisać konfigurację, wydaj poniższe polecenie (`netfilter` to polecenie używane przez `iptables-persistent` do tego celu):

```
$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
```

Teraz zapora jest gotowa do pracy.

Możesz rozważyć dodanie tymczasowych reguł do swojej zapory, ale pamiętaj o przysłowiu: „Nie ma nic trwalszego niż tymczasowa reguła zapory" (Austin Scott). Jeśli na przykład dodajesz tymczasową regułę, aby umożliwić użytkownikowi pobranie pliku z internetu, lepiej byłoby znaleźć inne rozwiązanie, na przykład użyć innego hosta. Jeśli dodasz taką regułę i pozostawisz ją w konfiguracji zapory, to stworzysz słaby punkt i zmniejszysz stopień bezpieczeństwa zapewniany przez zaporę. Unikaj tymczasowych reguł, jeśli to tylko możliwe.

pfSense

Oprócz zabezpieczenia każdego punktu końcowego w sieci za pomocą iptables powinieneś użyć zapory takiej jak pfSense, aby zabezpieczyć całą sieć na jej granicy. Każda z tych zapór dodaje kolejną warstwę do Twojej strategii dogłębnej obrony, utrudniając zadanie ewentualnym przeciwnikom w miarę rosnącego poziomu złożoności. Zaporę obwodową powinieneś umieścić na fizycznej krawędzi sieci — to znaczy tak blisko internetu, jak to możliwe, względem innych punktów końcowych w Twojej sieci. W większości sieci będzie to punkt bezpośrednio za modemem-routerem albo punkt graniczny, w którym Twoja sieć łączy się z systemami dostawcy usług internetowych. Ten sam efekt można osiągnąć logicznie, z wykorzystaniem maszyn wirtualnych i odpowiedniej konfiguracji routingu. Jednakże najlepszym i najbezpieczniejszym sposobem tworzenia zapory obwodowej jest użycie fizycznego urządzenia.

Podobnie jak iptables, pfSense jest zaporą stanową. Jednak w przeciwieństwie do zapory iptables, którą instaluje się w podstawowym systemie operacyjnym, takim jak Ubuntu, zapora pfSense jest w pełni funkcjonalnym systemem operacyjnym. Wywodzi się z FreeBSD, wersji Uniksa z otwartym kodem źródłowym (systemu podobnego do Linuksa, który ma własne jądro), ma przyjazne dla użytkownika funkcje, takie jak przeglądarkowy interfejs administracyjny, i można ją wdrożyć albo jako maszynę wirtualną, albo jako fizyczne urządzenie.

Fizyczną zaporę można utworzyć na kilka sposobów. Pierwszym jest zbudowanie jej z miniaturowego komputera, takiego jak Intel Next Unit of Computing (NUC). Jednakże za tę samą, a nawet niższą cenę firma Netgate sprzedaje urządzenia pfSense, które łatwo się konfiguruje i które są gotowe do pracy właściwie zaraz po wyjęciu z pudełka.

Dla prostoty (i bezpieczeństwa) omówimy korzystanie z gotowego urządzenia. W tej książce nie będziemy opisywać budowania urządzenia od podstaw, ponieważ ryzyko błędnej konfiguracji jest zbyt wysokie, zwłaszcza kiedy dostępne jest niedrogie, bezpieczne rozwiązanie. Kiedy pisałem tę książkę, model Netgate 2100 Base pfSense+ kosztował około 400 dolarów. Urządzenie to powinno poradzić sobie z każdym zadaniem, jakie mu powierzysz, wyjąwszy obsługę rozbudowanej sieci korporacyjnej. Model SG-3100 jest krokiem naprzód w stosunku do podstawowego modelu 2100 i oferuje więcej funkcji. Jest również wydajniejszy i osiąga większą przepustowość, więc jest idealnym wyborem dla mniejszych sieci.

(13) Instalacja zapory pfSense

Kiedy otrzymasz swoje urządzenie pfSense, wyjmij je z pudełka i podłącz zasilanie. Podłącz kabel Ethernet z portu WAN urządzenia do dowolnego portu na modemie kablowym, modemie DSL albo innym granicznym urządzeniu sieciowym. Podłącz drugi kabel Ethernet z portu LAN1 do portu Ethernet w swoim komputerze.

Aby uzyskać dostęp do strony konfiguracyjnej pfSense ze swojego komputera, przejdź pod adres 192.168.1.1, domyślny adres IP urządzenia SG-3100. Jeśli to nie zadziała, może będziesz musiał odłączyć komputer od swojej zwykłej sieci i ręcznie ustawić jego adres IP na 192.168.1.2 (albo dowolny inny adres z zakresu 192.168.1.*x*, z wyjątkiem 192.168.1.1, adresu zapory pfSense) według poniższych instrukcji. Jest to konieczne tylko podczas wstępnej konfiguracji urządzenia i wystarczy zrobić to raz na komputerze, którego używasz, aby przygotować urządzenie pfSense do pracy.

macOS

1. Otwórz *Preferencje systemowe*.
2. Kliknij *Sieć*.
3. Wybierz połączenie ethernetowe między urządzeniem pfSense a Twoim komputerem, a następnie ustaw listę rozwijaną *Konfiguruj IPv4* na pozycję *Ręcznie*.
4. Wprowadź 192.168.1.2 w polu *Adres IP*, ustaw pole *Maska podsieci* na 255.255.255.0 i wprowadź 192.168.1.1 w polu *Router*.
5. Kliknij *Zastosuj*.
6. Otwórz przeglądarkę internetową i przejdź pod adres 192.168.1.1. Powinieneś zobaczyć stronę logowania pfSense.

Windows

1. Otwórz *Ustawienia/Sieć i Internet*.
2. Kliknij *Zmień opcje karty*.
3. Otwórz połączenie ethernetowe między urządzeniem pfSense a Twoim komputerem, a następnie kliknij *Właściwości/Internet Protocol Version 4 (TCP/IP)/Właściwości*.
4. Zaznacz opcję *Użyj następującego adresu IP*.
5. Wprowadź 192.168.1.2 w polu *Adres IP*, ustaw pole *Maska podsieci* na 255.255.255.0 i wprowadź 192.168.1.1 w polu *Brama domyślna*.
6. Kliknij *OK* i zamknij pozostałe okna.
7. Otwórz przeglądarkę internetową i przejdź pod adres 192.168.1.1. Powinieneś zobaczyć stronę logowania pfSense.

Linux

1. Otwórz *Ustawienia*.
2. Kliknij *Sieć*.
3. Na połączeniu ethernetowym między urządzeniem pfSense a Twoim komputerem kliknij ikonę kółka zębatego.
4. Przejdź na kartę *IPv4*.
5. Zaznacz opcję *Ręcznie*.
6. Wprowadź 192.168.1.2 w polu *Adres*, ustaw pole *Maska sieci* na 255.255.255.0 i wprowadź 192.168.1.1 w polu *Brama*.
7. Kliknij *Zastosuj* i zamknij okno *Ustawienia*.
8. Otwórz przeglądarkę internetową i przejdź pod adres 192.168.1.1. Powinieneś zobaczyć stronę logowania pfSense.

UWAGA *Jeśli otrzymasz ostrzeżenie, że witryna nie jest prywatna albo bezpieczna, zamknij je, aby przejść do strony logowania. Ostrzeżenie pojawia się dlatego, że nie skonfigurowałeś jeszcze certyfikatu SSL, i na razie można je zignorować. Strzeż się jednak błędów tego rodzaju w innych miejscach; ogólnie rzecz biorąc, błąd certyfikatu SSL (zwłaszcza w internecie) to poważny sygnał, że strona, do której próbujesz uzyskać dostęp, nie jest bezpieczna.*

Na stronie logowania pfSense zaloguj się z wykorzystaniem poświadczeń otrzymanych wraz z urządzeniem. Po zalogowaniu się zaakceptuj umowę licencyjną użytkownika końcowego (EULA). Poświęć chwilę na przejrzenie informacji o systemie, a następnie kliknij menu *System* na górze strony i uruchom kreator instalacji — *Setup Wizard*. Wykonaj poniższe czynności, aby dokończyć konfigurowanie zapory pfSense:

1. Na ekranie powitalnym kliknij *Next*.
2. Jeśli pojawi się ekran *Support*, kliknij *Next*.
3. Na ekranie *General Information* wybierz nazwę hosta dla urządzenia albo pozostaw domyślną nazwę pfSense.
4. Jeśli w swoim środowisku masz skonfigurowaną domenę, wprowadź ją w polu *Domain*.
5. Na razie zignoruj ustawienia DNS i kliknij *Next*.
6. Na ekranie *Time Server Information* zaakceptuj domyślną nazwę serwera czasu, chyba że masz serwer czasu w swoim środowisku; w takim przypadku wprowadź tutaj informacje na jego temat.
7. Wybierz właściwą strefę czasową i kliknij *Next*.

Powinieneś teraz zobaczyć stronę *Configure WAN Interface*. Na tej stronie możesz skonfigurować urządzenie pfSense tak, aby łączyło się z dostawcą usług internetowych. Omówimy tu najbardziej typową konfigurację, nazywaną PPPoE;

prawdopodobnie będzie ona odpowiadać ustawieniom w Twoim modemie-routerze. Jeśli nie, skontaktuj się z dostawcą usług internetowych i zapytaj o szczegóły konfiguracyjne swojego połączenia.

1. W polu *SelectedType* zaznacz *PPPoE*.
2. Pomiń opcje *General configuration*, aby zaakceptować ustawienia domyślne.
3. Ustawienia statycznego adresu IP i konfiguracji klienta DHCP powinny być nieaktywne, więc przejdź do konfiguracji PPPoE.
4. Wprowadź nazwę użytkownika i hasło przypisane Ci przez dostawcę usług internetowych.
5. Zaakceptuj wszystkie pozostałe ustawienia i kliknij *Next*.
6. Ustaw adres IP urządzenia pfSense w sieci LAN. Możesz zachować schemat adresowania, który zidentyfikowałeś w rozdziale 1., nadając urządzeniu pierwszy adres IP w zakresie (192.168.1.1 w przypadku schematu adresowania 192.168.0.0/16), albo możesz zmienić go, określając inny adres LAN IP na tej stronie. Jeśli chciałbyś używać adresów z zakresu 10.0.0.0/8, podaj adres 10.0.0.1 itd. Następnie kliknij *Next*.
7. Zmień hasło administratora. Użyj mocnego hasła, liczącego co najmniej 12 znaków, i zapisz je w sejfie haseł (omówimy to dokładniej w rozdziale 11.). Kiedy to zrobisz, kliknij *Next/Reload/Finish*.

Wstępna konfiguracja jest teraz gotowa. Jeśli urządzenie zdołało połączyć się z dostawcą usług internetowych z wykorzystaniem Twoich poświadczeń, powinieneś móc otwierać strony internetowe. W przeciwnym przypadku trzeba będzie zdiagnozować problem. Najlepszym miejscem do diagnozowania problemów jest strona *System Logs* w menu *Status* na górze interfejsu przeglądarkowego. Przy odrobinie szczęścia problem okaże się oczywisty, kiedy spojrzysz na dzienniki. Jeśli jesteś pewien, że poprawnie wprowadziłeś wszystkie szczegóły konfiguracyjne, skontaktuj się z dostawcą usług internetowych i upewnij się, że podał Ci właściwe ustawienia.

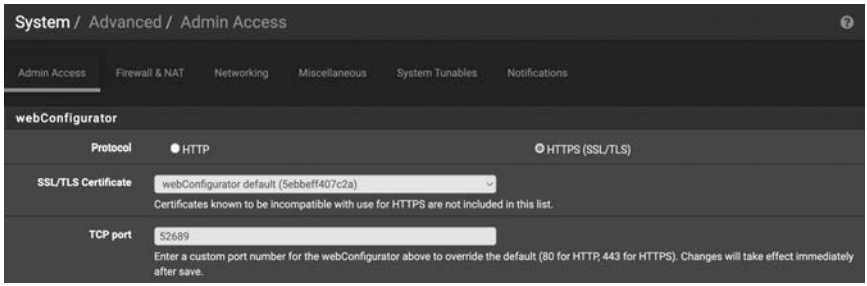
Hartowanie zapory pfSense

Zapora jest już skonfigurowana i nawet teraz powinna skutecznie blokować niepożądany ruch próbujący przeniknąć do Twojej sieci. Możesz jednak podjąć dodatkowe działania, żeby jeszcze lepiej zabezpieczyć swoje urządzenie i sieć.

Po zalogowaniu się w urządzeniu pfSense kliknij *System/Advanced*.

Na kartach *Advanced* możesz zmienić m.in. protokoły, porty i ustawienia serwera proxy używane przez pfSense. Zanim opuścisz którąś z kart, kliknij *Save*, aby zapisać zmiany.

Na karcie *Admin Access* pokazanej na rysunku 3.2 ustaw *Protocol* na *HTTPS*, aby komunikować się z urządzeniem przez zabezpieczone, szyfrowane połączenie. Zawsze lepiej używać protokołu HTTPS zamiast nieszyfrowanego HTTP, ponieważ szyfrowanie gwarantuje, że nawet jeśli przeciwnik przechwyci Twój ruch sieciowy, to nie będzie mógł go odszyfrować.



Rysunek 3.2. Zaawansowane menu pfSense

W następnej sekcji strony *Admin Access* (niepokazanej na rysunku 3.2) możesz zmienić opcje SSH. Lepiej nie zezwalać na dostęp SSH do urządzenia przez cały czas — przypominałoby to niezamykanie drzwi na noc. Jeśli zezwalasz na dostęp SSH tylko wtedy, kiedy go aktywnie używasz, przeciwnicy będą mogli spróbować wykorzystać go do włamania się do Twojej sieci wyłącznie w okresach, w których usługa jest aktywna. Jeśli usługa jest wyłączona przez 99 procent czasu, pozostawia to napastnikom tylko 1 procent czasu na ewentualny atak. Wyłącz tę opcję, chyba że aktywnie łączysz się z urządzeniem przez SSH. Po zaktualizowaniu odpowiednich ustawień kliknij przycisk *Save*.

Na karcie *Networking* możesz włączyć lub wyłączyć ruch IPv6. Jeśli nie używasz aktywnie IPv6, wyłącz jego obsługę, aby zmniejszyć swoją powierzchnię ataku. Jeśli to zrobisz, pozostałe ustawienia na tej stronie nie będą miały znaczenia.

Jeśli używasz serwera proxy do obsługi ruchu WWW, wprowadź informacje o proxy na karcie *Miscellaneous*. Jeśli planujesz zbudować własny serwer proxy według instrukcji z rozdziału 6., wróć do niniejszego rozdziału i wprowadź odpowiednie informacji, kiedy już to zrobisz.

Reguły zapory pfSense

Domyślne reguły zapory pfSense blokują połączenia z sieci prywatnych RFC1918 oraz **sieci bogonowych**. Adresy RFC1918, omówione w rozdziale 1., to zakresy zarezerwowane do wyłącznego użytku w prywatnych, wewnętrznych sieciach, co oznacza, że nie powinny się one pojawiać w publicznym internecie. Są to następujące zakresy: 192.168.0.0/16, 10.0.0.0/8 oraz 172.16.0.0/12. Jeśli taki adres pojawi się w internecie, Twoja zapora powinna uznać ruch za podejrzany i zablokować go. Podobnie, sieci bogonowe to adresy wprawdzie publiczne, ale nieprzydzielone nikomu przez IANA. Jeśli ktoś wysyła do Ciebie ruch z jeszcze nieprzydzielonego adresu, również jest to podejrzane, a zapora powinna odrzucić połączenie.

Choć domyślne reguły zapory są dobrym punktem wyjścia, powinieneś ręcznie dodać kilka reguł, aby zapewnić wyższy poziom bezpieczeństwa. Na przykład nie powinieneś zezwalać takim usługom jak **Server Message Block** (**SMB**), która umożliwia komputerom Windows współdzielenie plików przez sieć, wysyłać ruchu wychodzącego do internetu ani odbierać ruchu przychodzącego z internetu.

UWAGA *Ransomware WannaCry z maja 2017 roku wykorzystywał lukę w zabezpieczeniach protokołu SMB znaną jako EternalBlue; zablokowanie SMB w zaporze obwodowej znacznie ogranicza podatność na to zagrożenie oraz prawdopodobieństwo, że inne podobne usterki zostaną wykorzystane do naruszenia ochrony Twojej sieci.*

Aby dodać regułę, która blokuje ruch SMB, wykonaj poniższe czynności:

1. W pfSense, na górze strony, kliknij *Firewall/Rules*.
2. Kliknij *LAN/Add*, aby zacząć dodawać regułę.
3. Ustaw działanie na *block* (porzucanie pakietów) albo *reject* (odrzucanie pakietów z powiadomieniem nadawcy).
4. Ustaw pole *Address Family* na *IPv4*, a *Protocol* na *TCP*.
5. Ustaw *Source* na *Any*, *Destination* na *Any*, a *Destination Port Range* (*to* oraz *from*) na *(other) 445*.
6. Upewnij się, że zaznaczone jest pole *Log* w celu rejestrowania porzuconych pakietów, a następnie kliknij przycisk *Save*.

Kiedy to zrobisz, zapora nie będzie już zezwalać ruchowi SMB na przekraczanie granic Twojej sieci. Przeprowadź tę samą procedurę dla portów 137, 138 i 139, ponieważ działające w nich usługi (NetBIOS Name Resolution, NetBIOS Datagram Service i NetBIOS Session Service) również nigdy nie powinny przekraczać granic sieci, bo wszystkie te protokoły są używane w procesach wewnętrznych w sieci lokalnej.

PROTOKOŁY, KTÓRE NALEŻY BLOKOWAĆ

Istnieje kilka protokołów, które nigdy nie powinny przekraczać granicy (obwodu) sieci, na przykład:

- NetBIOS Name Resolution, TCP i UDP port 137: prekursor DNS, tłumaczy nazwy hostów na adresy IP.
- NetBIOS Datagram Service, UDP port 138: umożliwia przesyłanie komunikatów indywidualnych, grupowych i rozgłoszeniowych w sieci.
- NetBIOS Session Service, TCP port 139: umożliwia komunikację między dwoma komputerami w sieci.
- MS RPC, TCP i UDP port 135: umożliwia komunikację między aplikacjami klienta i serwera.
- Telnet, TCP port 23: niezabezpieczony protokół, który przesyła dane tekstem jawnym, używany do zdalnego dostępu i konserwowania systemów.
- SMB, TCP port 445: umożliwia komputerom Windows współdzielenie plików w sieci.
- SNMP, UDP porty 161 i 162: używany do zdalnego zarządzania i monitorowania systemów.
- TFTP, TCP i UDP port 69: umożliwia przesyłanie plików między komputerami w sieci.

(14) Testowanie zapory

Po dodaniu jednej lub wielu powyższych reguł przetestuj zaporę, aby się upewnić, że odpowiedni ruch jest rzeczywiście blokowany. Najlepszym narzędziem do tego celu jest **Nmap**, program do skanowania i mapowania sieci. Jest dostępny w wersjach z graficznym interfejsem użytkownika dla Windowsa, Linuksa i Maca (pod nazwą **Zenmap**), a także jako narzędzie wiersza poleceń. Zainstalowanie wersji z interfejsem graficznym instaluje również program wiersza poleceń, więc pobierz najnowszą wersję z witryny *https://www.nmap.org/* i zainstaluj ją.

Możesz też zainstalować Nmap z poziomu wiersza poleceń Ubuntu:

```
$ sudo apt install nmap
```

Po zainstalowaniu Nmap użyj poniższego polecenia, aby przeskanować port 445, który nakazaliśmy blokować:

```
$ sudo nmap -p 445 -A scanme.nmap.org
--wycięte--
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.20s latency).
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
PORT    STATE    SERVICE      VERSION
445/tcp filtered microsoft-ds
Service detection performed. Please report any incorrect results at https://
nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 2.54 seconds
```

Możesz wykonać to samo polecenie w interfejsie graficznym programu Zenmap — tylko pomiń `sudo`. Polecenie to przeprowadza skanowanie portów z Twojego urządzenia, które znajduje się za zaporą, na adresie *http://scanme.nmap.org/*, publicznej witrynie udostępnianej do celów testowych przez twórców Nmap.

Polecenie składa się z następujących elementów: `nmap` to nazwa programu. Argument `-p 445` określa port, który ma zostać przeskanowany; może to być lista rozdzielona przecinkami (na przykład `-p 445,137,138,22`), konkretny port, jak w powyższym przykładzie, albo zakres portów, taki jak `-p1-1024`. Argument `-A` nakazuje programowi Nmap spróbować zidentyfikować usługę i system operacyjny w każdym skanowanym porcie, a `scanme.nmap.org` to witryna albo system, który należy przeskanować. Jeśli w wyświetlonych wynikach stan (*STATE*) jest oznaczony jako *filtered*, oznacza to, że zapora zablokowała ruch, a nowo dodana reguła działa poprawnie. Jeśli **stan** jest oznaczony jako *closed*, oznacza to, że zapora przepuszcza ruch i sama witryna, a nie zapora, zwróciła komunikat, że port jest zamknięty. Jeśli otrzymasz taki wynik, to reguła zapory albo nie jest skonfigurowana, albo nie działa.

Kiedy upewnisz się, że reguły działają, przejdź do dzienników zapory, aby obejrzeć zablokowane pakiety. W pfSense, na górze strony, kliknij *Status*/*System Logs*/*Firewall*, aby wyświetlić ostatnie 500 wpisów dziennika zapory, jak pokazano na rysunku 3.3.

Last 500 Firewall Log Entries. (Maximum 500)

Action	Time	Interface	Rule	Source	Destination	Protocol
✕	May 31 10:09:07	LAN2	Block all IPv6 (1000000003)	[fe80::ea6f:38ff:fe33:b5dd]:5353	[ff02::fb]:5353	UDP
✕	May 31 10:09:07	bridge0	Block all IPv6 (1000000003)	[fe80::ea6f:38ff:fe33:b5dd]:5353	[ff02::fb]:5353	UDP
✕	May 31 10:09:07	LAN2	Block all IPv6 (1000000003)	[fe80::ea6f:38ff:fe33:b5dd]:5353	[ff02::fb]:5353	UDP
✕	May 31 10:09:15	LAN2	Block all IPv6 (1000000003)	[fe80::ea6f:38ff:fe33:b5dd]:5353	[ff02::fb]:5353	UDP
✕	May 31 10:09:15	bridge0	Block all IPv6 (1000000003)	[fe80::ea6f:38ff:fe33:b5dd]:5353	[ff02::fb]:5353	UDP
✕	May 31 10:09:15	LAN2	Block all IPv6 (1000000003)	[fe80::ea6f:38ff:fe33:b5dd]:5353	[ff02::fb]:5353	UDP
✕	May 31 10:09:21	WAN	Default deny rule IPv4 (1000000103)	193.46.255.123:34064	60.242.70.144:5060	UDP
✕	May 31 10:09:22	WAN	Default deny rule IPv4 (1000000103)	92.63.197.97:41735	60.242.70.144:6733	TCP:S

Rysunek 3.3. Dziennik zapory pfSense

Prawdopodobnie zobaczysz mnóstwo zablokowanego ruchu. W tym momencie trudno powiedzieć, co może reprezentować ten zablokowany ruch. Na przykład jeden z wpisów na górze mojego dziennika pokazuje zablokowane połączenie z adresu 80.82.77.245 do portu 46732.

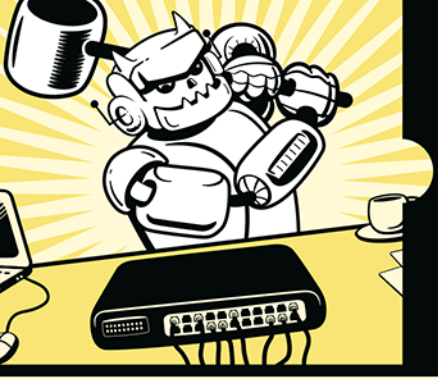
Po bliższym zbadaniu okazuje się, że jest to usługa, która rzekomo przeprowadza regularne skanowanie publicznych adresów IP „w celach badawczych". Może to jednak być cokolwiek; skąd mam wiedzieć, czy te „badania" są uzasadnione, czy może napastnik próbuje znaleźć dziury w zaporze, aby przedostać się do mojej sieci? W większości przypadków nie da się tego stwierdzić, ale przynajmniej moja zapora blokuje tę aktywność, a ja mogę znaleźć ją w dziennikach, jeśli zechcę przyjrzeć się jej bliżej i podjąć jakieś działania. W rozdziale 10., który jest poświęcony monitorowaniu bezpieczeństwa sieci, wyjaśnimy, do czego mogą się przydać te informacje.

Podsumowanie

Po zainstalowaniu zapory w hoście i zapory obwodowej Twoja sieć i hosty są znacznie bezpieczniejsze. W projektach omówionych w tym rozdziale utworzyłeś reguły i zestawy reguł, które znacznie utrudniają przeciwnikom infiltrację Twojej sieci, zwłaszcza w sposób niezauważony.

Choć w niniejszym rozdziale poznałeś podstawy działania zapór, w Twoim najlepszym interesie jest bliżej zbadać porty i protokoły, które chciałbyś blokować albo przepuszczać do wnętrza i na zewnątrz swojej sieci. Każda sieć jest inna i ma inne wymagania.


PROGRAM
PARTNERSKI
GRUPY HELION
1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ
Zmień swoją stronę WWW w działający bankomat!
Dowiedz się więcej i dołącz już dzisiaj!
http://program-partnerski.helion.pl
GRUPA
Helion



ZAHARTOWANA MAŁA SIEĆ? ZROBISZ TO SAMODZIELNIE!

Małe sieci, stosowane przez rodziny lub niewielkie firmy, są łakomym kąskiem dla różnego rodzaju przestępców. Polują oni na dane osobowe, wrażliwe dane medyczne i identyfikacyjne, a także na własność intelektualną. Wielu właścicieli małych sieci nie zabezpiecza ich wystarczająco, tymczasem konsekwencje kradzieży poufnych czy wrażliwych informacji mogą być śmiertelnie poważne.

Rozwiązanie problemu znajdziesz w tej książce. Została ona napisana specjalnie z myślą o osobach, które administrują małymi sieciami, dysponują niewielkim budżetem i ograniczonym wsparciem profesjonalistów. Dzięki niej zrozumiesz podstawy zabezpieczania łączności sieciowej i poznasz sposoby zabezpieczania sieci przy niewielkim nakładzie pieniędzy i czasu. Opanujesz uznane techniki hartowania systemów, takie jak mapowanie sieci, śledzenie stanu urządzeń i rozpoznawanie nietypowej aktywności, która może sygnalizować atak. Zagłębisz się w sposoby eliminowania luk w zabezpieczeniach i zapobiegania dostępowi do urządzeń mobilnych i stacjonarnych, a nawet punktów końcowych IoT. Dowiesz się też, jak wdrażać własne strategie backupu, a także wykrywać i blokować złośliwe oprogramowanie i ransomware.

W książce między innymi:

- użycie zapór do filtrowania ruchu sieciowego
- tworzenie planu segmentacji sieci i zarządzanie dostępem użytkowników
- szyfrowanie i ochrona komunikacji sieciowej
- ukrywanie wrażliwych danych
- przechwytywanie i analizowanie ruchu sieciowego
- korzystanie z Security Onion i alarmy o podejrzanej aktywności

Seth Enoka jest weteranem w branży cyberbezpieczeństwa. Uczestniczył w dochodzeniach związanych z naruszeniami bezpieczeństwa na całym świecie, jest uznanym mentorem w dziedzinie informatyki śledczej i reakcji na incydenty. W wolnym czasie zdobywa stopnie naukowe i certyfikaty albo przygotowuje się do konkursu podnoszenia ciężarów.

Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej! ▶

ISBN 978-83-289-0428-6
9 788328 904286

Cena: 67,00 zł

no starch press