Zapora sieciowa w systemie Linux

Zapora sieciowa w systemie Linux wbudowana jest bezpośrednio w jądro systemu. Najbardziej popularnym modułem jest na razie Netfilter. Jego zadaniem jest przechwytywanie i i zmiana ruchu wszystkich pakietów sieciowych jakie docierają do naszej maszyny. W domyślnej konfiguracji (nie modyfikowanej przez żadną dystrybucję) przepuszcza on cały ruch sieciowy w niezmienionej postaci (wchodzący i wychodzący). Odpowiednie modyfikacje pozwalają na następujące operacje:
- blokowanie ruchu wychodzącego i przychodzącego
- przekierowywanie ruchu pomiędzy sieciami (np. siecią zewnętrzną i wewnętrzną)
- zapobieganie nieautoryzowanym przejęciom sesji połączenia
- rozdzielaniu połączenia do sieci rozległej (brama do sieci Internet)
- przekierowaniu portów usług sieciowych (np. z sieci zewnętrznej do zasobów lokalnych)

Netfilter można konfigurować poprzez narzędzie **iptables**. Zawiera ono 4 tablice, do których można dodawać reguły (nazywane łańcuchami), które w odpowiedni sposób przetwarzane są przez jądro systemu.

Pierwszą, domyślą tablicą jest filter. Umieszcza się w niej reguły, które mają za zadanie filtrować wszystkie pakiety przechodzące przez system. Podstawowo zawiera 3 łańcuchy, chociaż użytkownicy mogą definiować swoje własne:
- INPUT – ten łańcuch przejmuje pakiety przychodzące do systemu (są przez niego odbierane)
- FORWARD – obsługuje pakiety, które zostaną przekazane dalej (są trasowane)
- OUTPUT – odpowiada za pakiety, które generuje używany system/sprzęt komputerowy

Aby zobaczyć wpisy w tej tablicy należy posłużyć się poleceniem:

iptables -t filter -L

lub po prostu

iptables -L

Tablica nat obsługuje pakiety, które podlegają tzw. maskaradzie (Network Address Translation – tłumaczeniu adresów sieciowych). Może służyć do tworzenia reguł przekierowywania ruchu do innych sieci (system będzie pełnił rolę routera), jak i przekierowywać ruch z jednego interfejsu na inny. Pozwala także na przekierowywanie portów (ang. port forwarding). Tablica ta, w przeciwieństwie do pozostałych, obsługuje jedynie pierwszy pakiet rozpatrywanego połączenia (dla pozostałych pakietów automatycznie stosuje te same zasady). Zawiera trzy domyślne łańcuchy:
- PREROUTING – obsługuje pakiety przychodzące; ponieważ na tę chwilę nie ma jednoznaczniej informacji czy połączenie zostanie obsłużone lokalnie, czy przekazane dalej (np. do komputera w sieci lokalnej) nie ma możliwości wskazania wyjściowego interfejsu (parametr -o <nazwa_interfejsu>)
- POSTROUTING – obsługuje pakiety wychodzące; ponieważ na tę chwilę nie ma jednoznaczniej informacji czy połączenie zostało wygenerowane przez system lokalny czy urządzenie sieci lokalnej. W związku z tym nie ma możliwości wskazania wejściowego interfejsu (parametr -i <nazwa_interfejsu>)
- OUTPUT – obsługuje pakiety generowane przez system lokalny. Pakiety obsłużone przez ten łańcuch przechodzą w następnej kolejności przez łańcuch POSTROUTING

Aby zobaczyć wpisy w tej tablicy należy posłużyć się poleceniem:

iptables -t nat -L

Tablica mangle operuje na regułach, które mogą zmieniać niektóre parametry pakietów (np. ttl – time-to-live czy MSS – Maximum Segment Size). Pakiety te nie są tłumaczone (NAT). Dostępne łańcuchy:
- PREROUTING – obsługuje pakiety przychodzące
- INPUT – obsługa pakietów odebranych przez system (po uprzedniej obróbce w łańcuchu PREROUTING)
- FORWARD – obsługuje pakiety przekazane przez system lokalny. Pakiety te pochodzą z łańcucha PREROUTING oraz, jeżeli nie zostaną zatrzymane, trafią do łańcucha POSTROUTING
- POSTROUTING – obsługuje pakiety wychodzące z lokalnego systemu
- OUTPUT – obsługuje pakiety generowane przez system lokalny (przechodzą także przez POSTROUTING)

Aby zobaczyć wpisy w tej tablicy należy posłużyć się poleceniem:

iptables -t mangle -L

Ostatnią tablicą jest raw. Pakiety obsługiwane przez nią są traktowane priorytetowo (są obsługiwane przez jakąkolwiek klasyfikacją). Zawiera dwa łańcuchy:
- PREROUTING – obsługuje pakiety wchodzące do systemu
- OUTPUT – obsługuje pakiety generowane przez system

Wyświetlenie wpisów:

iptables -t raw -L

Obsługa Netfilter poprzez iptables (wersja ogólna):

iptables <tablica> <polecenie> <dopasowania> <akcja>

Szczegóły możliwości polecenia można znaleźć w podręczniku dostępnym w systemie:

man iptables

lub na stronie WWW – np. https://pl.wikibooks.org/wiki/Sieci_w_Linuksie/Netfilter/iptables

Przykładowe polecenia iptables:

iptables -P INPUT DROP

polecenie blokuje domyślnie CAŁY ruch sieciowy do systemu operacyjnego. Żaden pakiet przychodzący się do niego nie dostanie (zostanie domyślnie odrzucony). Dotyczy to wszystkich interfejsów (w tym localhost – lo). Polecenie można stosować dla pozostałych łańcuchów.

iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

jeżeli użyliśmy wcześniejszego polecenia, powyższe polecenie pozwoli nam przepuścić pakiety wchodzące, których inicjalizacja nastąpiła na naszym komputerze (np. zażądaliśmy treści strony WWW). Odwołanie się do narzędzia conntrack pozwala mieć pewność, że dane połączenie jest tym samym, które rozpoczęło pierwotną transmisję. Stan RELATED oznacza, że pakiet rozpoczyna nowe połączenie powiązane bezpośrednio z istniejącym (wywołującym) ,ESTABLISHED natomiast, że pakiet należy do obecnego połączenia. Ostatni człon, -j ACCEPT wskazuje, że filtr ma zaakceptować pakiet spełniający regułę.

Więcej o akcjach - https://pl.wikibooks.org/wiki/Sieci_w_Linuksie/Netfilter/iptables/akcje
Więcej o dopasowaniach: https://pl.wikibooks.org/wiki/Sieci_w_Linuksie/Netfilter/iptables/dopasowania

iptables -A INPUT -i lo -j ACCEPT

jeżeli zablokowaliśmy domyślnie cały ruch INPUT powinniśmy odblokować akceptowanie wszystkich (bez wyjątku) pakietów połączenia lokalnego (localhost). Często może się bowiem zdarzyć, że system będzie inicjował połączenia na adresie 127.0.0.1 i/lub ::1%128, które zostaną odrzucone (ze względu na domyślą politykę odrzucania pakietów przychodzących), co może skutkować niestabilnym działaniem całego systemu.

Tym prostym sposobem zamkniemy nasz system na przyjmowanie pakietów z zewnątrz, jednak każdy nasz ruch (np. połączenie do poczty, serwera plików czy stron WWW) będzie jak najbardziej możliwy.

Prosta konfiguracja iptables zabezpieczająca system operacyjny przed niechcianymi pakietami i/lub częścią ataków typu DoS/DdoS:

https://wiki.archlinux.org/index.php/simple_stateful_firewall

Przykładowa konfiguracja zapory sieciowej przepuszczającej ruch zewnętrzny do usługi SSH (wraz z komentarzem).

Jeżeli na co dzień korzystamy z komputera PC, przeważnie trzymamy na nim tworzone bądź edytowane pliki. Czasami może możemy znajdować się poza domem i potrzebować dostać się do naszego komputera (np. celem pobrania bądź przejrzenia swoich plików, pobrania/wysłania czegoś przez pocztę lub skonfigurowania jakiejś usługi). W tym celu możemy wykorzystać SSH (Secure Shell). Narzędzie to wykorzystywane jest głównie przez administratorów sieciowych do konifuracji serwerów, jednak może być wykorzystane przez każdego z użytkowników do wyżej wspomnianych celów.

Zakładając, że wycinamy cały ruch przychodzący do naszego systemu (konfiguracja wyżej), musimy dokonać pewnych zmian. Poniżej pełna konfiguracja iptables wraz z komentarzami (komentarze w postaci opisów do poleceń w skrypcie iptables).

UWAGA: Poniższe polecenia można też wpisywać bezpośrednio w linii poleceń (każda rozpoczyna się od polecenia iptables).

```
#wycięcie całego ruchu przychodzącego (włącznie z tym rozpoczętym)
iptables -P INPUT DROP
#umożliwienie nawiązywanie ruchu sieciowego po pętli zwrotnej
iptables -A INPUT -i lo -j ACCEPT
#podtrzymanie ruchu pakietów rozpoczętego przez nasz komputer
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
#otworzenie portu usługi SSH
iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
#upewnienie się, że każdy przekazany pakiet oraz każdy wyjściowy pakiet (wysyłany przez nas) na
#pewno zostanie przepuszczony
iptables-A FORWARD -j ACCEPT
iptables -A OUTPUT -j ACCEPT
```

Uruchamianie usługi na domyślnym porcie nie zawsze jest dobrym pomysłem. Ponadto większość usług, jak nie wszystkie, powinny być uruchamiane jedynie po pętli zwrotnej (127.0.0.1/::1). Sam iptables nie umożliwia jednak filtrowania ruchu po pętli zwrotnej. Dodatkowo wymaga odblokowania wszystkich portów, na których dostępne są usługi.
W związku z tą sytuacją należy posłużyć się pewną sztuczką. Po pierwsze konfigurujemy naszą usługę w taki sposób, by działała jedynie na adresach pętli zwrotnej (plik /etc/ssh/sshd_config):

AddressFamily any
ListenAddress 127.0.0.1

Jak widać, jedynym adresem, na którym SSH będzie nasłuchiwać będzie 127.0.0.1. Kolejną ważną rzeczą będzie ustawienie innego (niż domyślny) portu, na którym będzie działała usługa:

Port 7800

Ta linia spowoduje, że usługa zacznie nasłuchiwać na porcie 7800.

Restartujemy usługę np. poprzez polecenie (wykonywane jako root):

systemctl restart sshd.service

Teraz gdy z naszego komputera wpiszemy

ssh nazwa@127.0.0.1 -p 7800

to system spróbuje nas zalogować zdalnie do siebie samego (otworzymy wirtualną konsolę, którą będziemy mogli zarządzać fizycznym sprzętem). Aby przywrócić właściwą funkcjonalność SSH (zarządzać komputerem z innych miejsc na globie) przekierujemy pakiet z domyślnego portu na port 7800 znajdującym się dokładnie na naszym komputerze.

Należy dodać następującą linię w konfiguracji (można również dodać drugą):

iptables -A OUTPUT -p tcp -m tcp --dport 22 -j REDIRECT --to-ports 7800

Spowoduje ona przekierowanie KAŻDEGO pakietu z portu 22 na port 7800. W tym wypadku trzeba dokonać otwarcia portu 7800 dla pozostałych:

iptables -A INPUT -p tcp -m tcp --dport 7800 -j ACCEPT

(analogicznie do 22). Na koniec można dodać następującą linię:

iptables -A PREROUTING -p tcp -m tcp --dport 22 -j DNAT --to-destination 127.0.0.1:7800

linia ta przekieruje każdy pakiet dostający się do nas na port 22 do portu 7800 dostępnego na adresie 127.0.0.1

Bez względu na sposób przekierowania powinniśmy uzyskać możliwość łączenia się przez SSH do naszego komputera bez podawania portu (przekierowanie niejawne). Dodatkowo można dołożyć dodatkową funkcjonalność, taką jak sprawdzanie czy pakiet przychodzący na port 22 na pewno przesłany jest przez klienta SSH (a nie usługi go udającej).