Zabezpieczenia w systemach Linux

Autorzy: Krzysztof Majka, Mirosław Mika IVFDS

STRESZCZENIE

Problemem tej pracy było pokazanie w jaki sposób można skonfigurować zaporę sieciową na systemie operacyjnym Linux. Pierwszy rozdział przedstawia istotę potencjalnych ataków zagrażających bezpieczeństwu komputera jak również charakteryzuje najważniejsze typy ataków. W kolejnych rozdziałach przedstawiony został schemat organizacji zapory sieciowej i proces jej budowania. Na końcu tej pracy zostały przedstawione dwa przykłady konfiguracji firewall-a dla różnych topologii sieci.

SPIS TREŚCI

1. WSTĘP

Zabezpieczanie systemu musi prowadzić do zmniejszania ryzyka nieupoważnionego zmieniania danych, czy to przesyłania przez sieć czy to składowanych w naszym systemie.

Najważniejsze metody ataku i sposoby zabezpieczania się przed nimi:

- ***Nieautoryzowany dostęp***

Oznacza po prostu, że ludzie, którzy nie powinni korzystać z usług oferowanych przez komputer, są w stanie się do niego podłączyć i z nich korzystać. Na przykład ludzie spoza firmy mogą połączyć się z komputerem obsługującym księgowość firmy lub z serwerem NFS. Istnieją różne sposoby uniknięcia tego ataku. Trzeba precyzyjnie określić, kto może mieć dostęp do danych usług. Można zabronić dostępu do sieci wszystkim poza wyznaczonymi osobami.

- ***Wykorzystanie znanych dziur w programach***

Wtedy kiedy powstały niektóre programy i usługi sieciowe, nie uwzględniano jeszcze rygorystycznych zasad bezpieczeństwa. Te właśnie są z natury bardziej podatne na zagrożenia. Usługi zdalne BSD (rlogin, rexec itp.) są tu doskonałym przykładem. Najlepszym sposobem na zabezpieczenie się przed tego typu atakiem jest wyłączenie wszelkich podatnych usług lub znalezienie alternatywy. W przypadku Open Source czasem jest możliwe załatanie dziury w programie.

- ***Odmowa obsługi***

Ataki typu odmowa obsługi powodują, że usługa lub program przestają działać lub nie pozwalają innym z siebie korzystać. Może to być spowodowane wysyłaniem w warstwie sieciowej starannie przygotowanych, złośliwych datagramów, które powodują awarie połączeń sieciowych. Ataki mogą być realizowane w warstwie aplikacji, gdzie starannie przygotowane polecenia aplikacji podane programowi powodują, że staje się on zajęty lub przestaje działać. Uniemożliwienie podejrzanym pakietom sieciowym dotarcie do hosta oraz zapobiegnięcie uruchamianiu podejrzanych poleceń i żądań są najlepszymi sposobami na zminimalizowania ryzyka ataku odmowy obsługi.

- ***Podszywanie się***

Ten typ ataku powoduje, że host lub aplikacja naśladują działanie innego. Zwykle atakujący udaje niewinny host, przesyłając sfałszowany adres IP w pakietach sieciowych. Na przykład dobrze udokumentowany sposób wykorzystania usługi rlogin BSD stosuje te metodę do udawania połączeń TCP z innego hosta. Robi to, odgadując numery kolejnych pakietów TCP. Aby zabezpieczyć się przed tego typu atakiem, należy weryfikować wiarygodność datagramów i poleceń. Wyłączyć możliwość rutowania datagramów o złym adresie źródłowym. Wprowadzić nieprzewidywalność do mechanizmów kontroli połączenia, na przykład stosowanie kolejnych numerów TCP lub alokację dynamicznych adresów portów.

- ***Podsłuchiwania***

Jest to najprostszy typ ataku. Host jest skonfigurowany na „słuchanie” i zbieranie danych nie należących do niego. Dobrze napisane programy podsłuchujące mogą odczytać z połączeń sieciowych nazwy użytkowników i hasła. Sieci rozgłoszeniowe, takie jak Ethernet, są szczególnie podatne na tego typu atak.

[2]

Firewalle IP są bardzo użyteczne; są w stanie zapobiec nieautoryzowanym dostępom, odmowom obsługi w warstwie sieciowej i atakom przez podszywanie się lub znacznie zmniejszyć ryzyko ich wystąpienia. Niezbyt dobrze zabezpieczają przed wykorzystywaniem

dziur w usługach sieciowych czy programach oraz nie zapobiegają podsłuchiwaniu.

Zapora sieciowa (ang. ***Firewall***) to konstrukcja zapewniająca kontrolowane połączenie pomiędzy siecią prywatną a Internetem (siecią publiczną). Dostarcza ona mechanizmu kontroli ilości i rodzaju ruchu sieciowego między obydwoma sieciami. Zapory sieciowe to narzędzia o dużych możliwościach, ale nie powinno się ich używać zamiast innych środków bezpieczeństwa, lecz obok nich.

Podstawowe funkcje, które powinna spełniać zapora sieciowa to:

- zapewnienia „bezpiecznego" dostępu do Internetu użytkownikom sieci prywatnej,
- zapewnienie zasobów ochrony sieci prywatnej przed atakami z zewnątrz,

Oprócz tych dwóch podstawowych funkcji można jeszcze wyszczególnić kilka dodatkowych, które z powodzeniem może realizować zapora sieciowa:

- blokowanie dostępu do określonych miejsc w Internecie, blokowanie (całkowite lub częściowe) dostępu do Internetu określonym użytkownikom,
- monitorowanie komunikacji pomiędzy siecią prywatną a Internetem,
- rejestrowanie całości lub określonej części ruchu międzysieciowego,
- tworzenie prywatnych sieci wirtualnych (VPN) pomiędzy oddziałami organizacji [1]

2. SCHEMAT ORGANIZACJI ZAPORY SIECIOWEJ

Na konstrukcję zaporę sieciowej zwykle składają się filtry pakietów oraz serwery proxy.

Podstawowym zadaniem zapory jest ograniczenie przepływu danych między sieciami. Przed postawieniem zapory trzeba określić, jakie rodzaje danych mają być przez nie przepuszczane, a jakie nie. Czyli trzeba zdefiniować **politykę zapory**. Następnie należy skonstruować mechanizmy, które umożliwią wprowadzenie tej polityki.

Filtry pakietów to urządzenia przechwytujące każdy transmitowany pakiet danych i dopuszczające lub blokujące przesłanie tego pakietu do adresata. Decyzja o przesłaniu jest podejmowana na podstawie atrybutów rozpatrywanego pakietu. Są to m.in. adres źródłowy, adres docelowy, typ protokołu, port źródłowy, port docelowy, zawartość.

W praktyce funkcjonują dwie podstawowe strategie konfiguracji filtrów pakietów -domyślne przepuszczanie oraz domyślne powstrzymanie. Pierwsza polega na blokowaniu tylko niektórych portów, protokołów czy adresów. Stosowana jest więc zasada: wszystko, co nie jest zabronione jest dozwolone. Druga strategia polega na odblokowaniu tylko niektórych portów, protokołów czy adresów. Obowiązuje więc zasada: wszystko, co nie jest dozwolone jest zabronione.

Serwer proxy to pakiety programowe służące do pośredniczenia w ruchu sieciowym pomiędzy siecią prywatną a Internetem. Użytkownik sieci prywatnej, który chciałby skorzystać z usługi udostępnianej na serwerze w Internecie, rejestruje się najpierw w aplikacji serwera proxy. Zadaniem tego serwera jest uwierzytelnienie użytkownika i po stwierdzeniu, że ma on odpowiednie prawa, zezwolenie na skorzystanie z usługi w Internecie. Przy połączeniach z sieci zewnętrznej postępowanie jest podobne. Ponieważ serwer proxy działa na poziomie aplikacji, więc każdy typ aplikacji wymaga oddzielnego serwera. Taki zestaw serwerów proxy nazywamy bramą aplikacyjną.

Przez połączenie filtrów pakietów i serwerów proxy , oraz ich odpowiednie osadzenie na platformach sprzętowych, można uzyskać różne konfiguracje zapór sieciowych. Najbardziej popularne są w tej chwili cztery konfiguracje:

a. host z dwoma portami,
b. filtr pakietów,
c. zapora z jednym filtrem pakietów i bramą aplikacyjną,
d. zapora z dwoma filtrami pakietów i bramą aplikacyjną

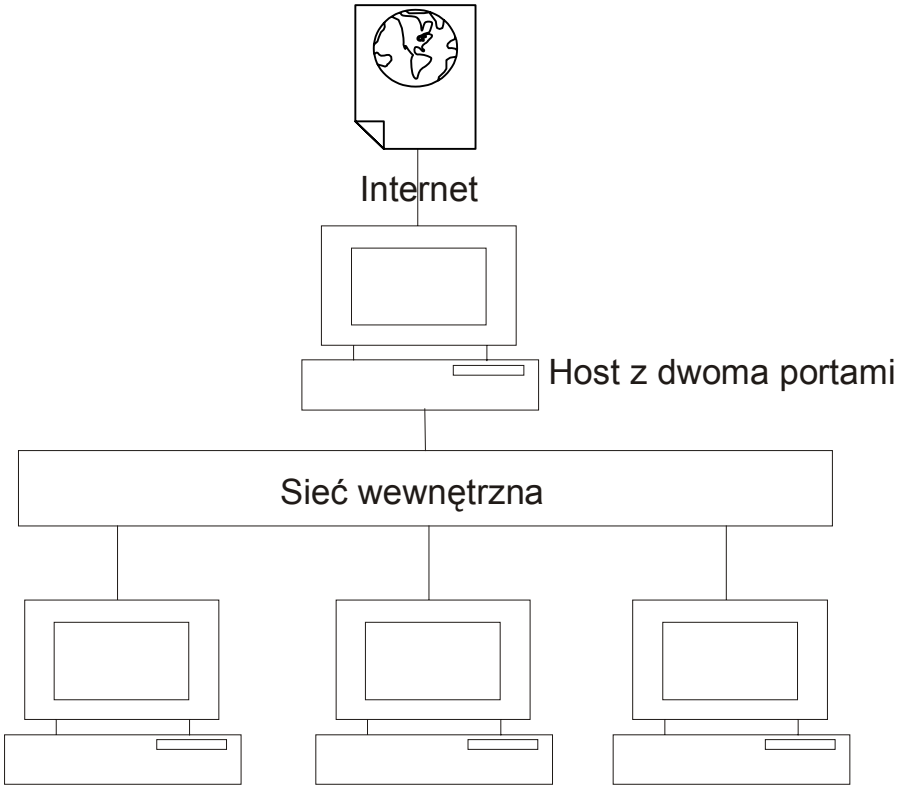
[1]

2.1 Host z dwoma portami

Polega na osadzeniu zapory na komputerze wyposażonym w dwa interfejsy sieciowe, pracującym zwykle pod kontrolą systemu operacyjnego z rodziny UNIX. Komputer w zaporze działa jednocześnie jako dławik i brama. Usługi są zwykle oferowane użytkownikom na dwa sposoby:

- użytkownik loguje się do komputera z dwoma portami,
- na hoście z dwoma portami mogą działać serwery proxy poszczególnych, przepuszczanych przez zaporę usług.

W systemie operacyjnym, a dokładnie mówiąc w jego jądrze musi być włączona opcja *ip_forwarding*. [1]



Rys 1 Host z dwom portami

2.2 Filtr pakietów

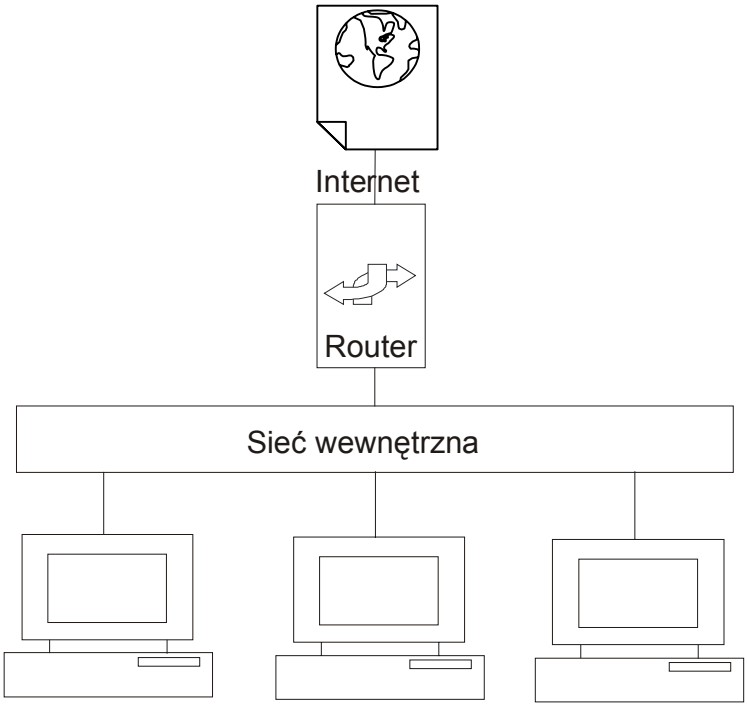
Ten typ zapory buduje się na bazie jednego filtru pakietów. Może nim być np. router, w którym dostępna jest funkcja filtrowania pakietów. Jest to konfiguracja prosta i dość popularna.

Programowanie filtru polega na:

- zablokowaniu wszystkich nieużywanych usług,
- zablokowaniu pakietów z ustawioną opcją routingu źródłowego,

- zezwoleniu na połączenia przychodzące tylko z określonych serwerów sieciowych i blokowaniu pozostałych,
- zezwoleniu komputerom z sieci wewnętrznej na połączenia z dowolnym komputerem z sieci zewnętrznej

Do zalet takiej konfiguracji należy zaliczyć prostotę, taniość i elastyczność wyrażającą się łatwością blokowania dostępu z wybranej sieci zewnętrznej. [1]



Rys 2 Filtr pakietów

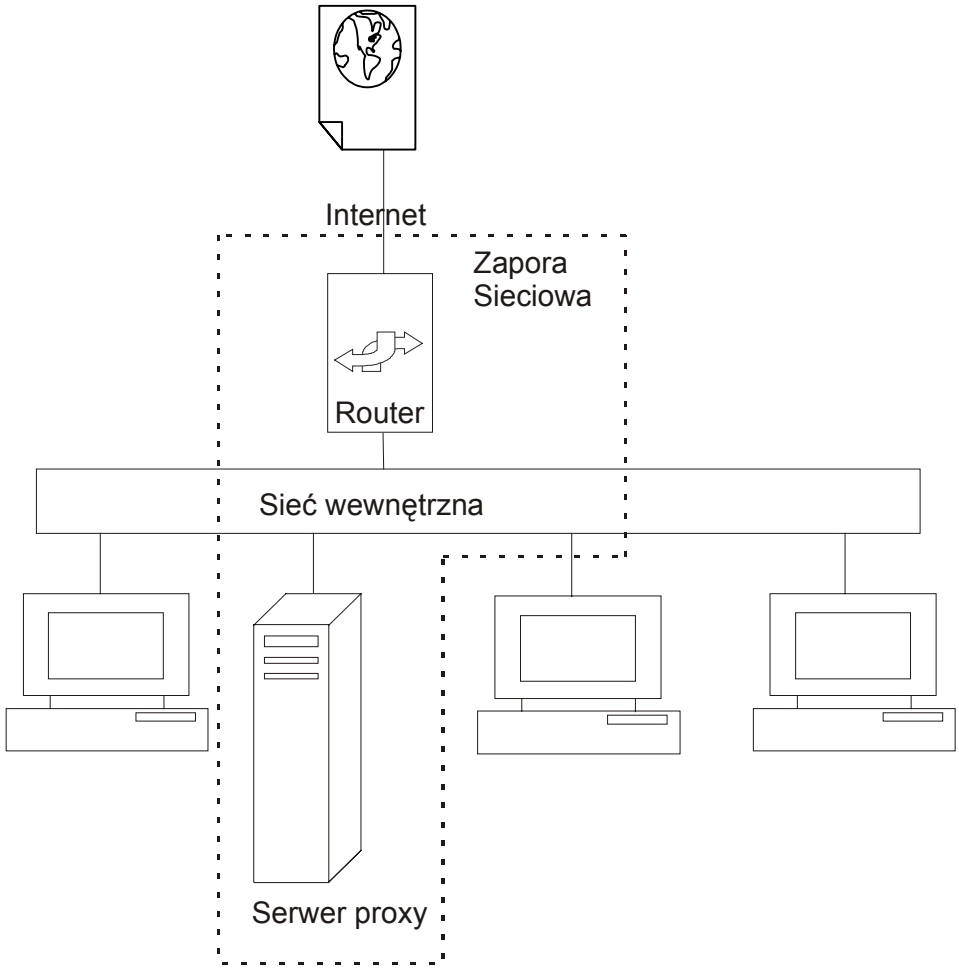
2.3 Zapora z jednym filtrem pakietów i bramą aplikacyjną

Bardziej bezpieczną zaporę sieciową można zbudować stosując jednocześnie filtr pakietów i bram aplikacyjnych. Filtrem pakietów może być router, a bramą aplikacyjną wybrany komputer w sieci wewnętrznej. W bramie działają serwery proxy umożliwiające użytkownikom sieci wewnętrznej korzystanie z usług sieci zewnętrznej.

W tej konfiguracji filtr pakietów jest skonfigurowany w sposób zapewniający:

- blokowanie pakietów usług, które nie są potrzebne w sieci wewnętrznej,
- blokowanie pakietów przesyłanych w ramach routingu źródłowego lub mających ustawione nietypowe opcje
- blokowanie pakietów, których miejscem przeznaczenia jest sieć wewnętrzna (poza adresem bramy)
- przepuszczanie pakietów, których adresem źródłowym lub docelowym jest adres bramy aplikacyjnej

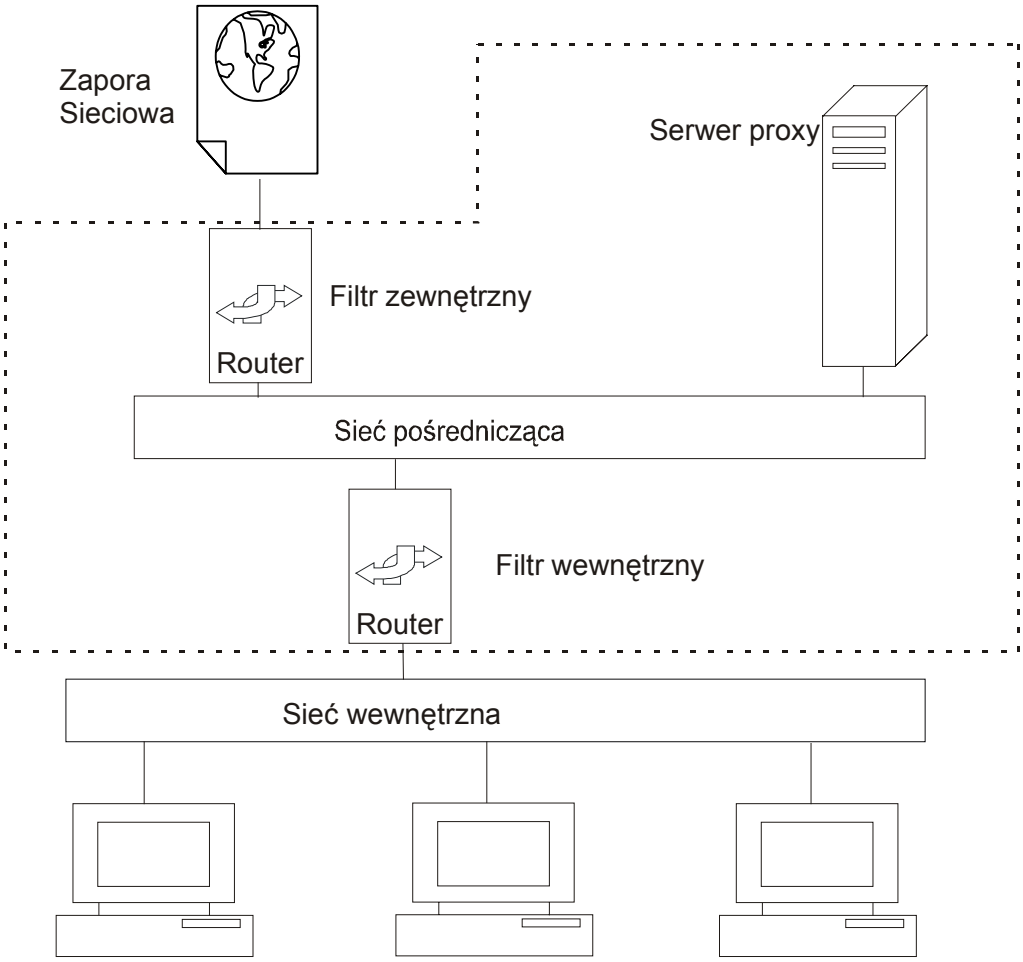
Jeżeli komputer w sieci wewnętrznej chce się skontaktować z siecią zewnętrzna, to pakiet komunikacyjny musi przejść przez serwer proxy funkcjonujący w bramie aplikacyjnej. Użytkownicy z sieci zewnętrznej zanim dostaną się do sieci wewnętrznej muszą się połączyć z odpowiednim serwerem proxy.



Rys 3 Zapora z jednym filtrem pakietów i bramą aplikacyjną

2.4 Zapora z dwoma filtrami pakietów i bramą aplikacyjną

W takiej konfiguracji filtr zewnętrzny i serwer proxy pełnią takie same funkcje jak w konfiguracji z jednym filtrem pakietów i bramą aplikacyjną. Nowym elementem jest filtr wewnętrzny, pełniący funkcję awaryjną. Jeśli intruzowi uda się włamać do serwera proxy i przejąć nad nim kontrolę, filtr wewnętrzny uniemożliwi mu posłużenie się serwerem proxy do przeprowadzania ataków na inne komputery w sieci wewnętrznej (dzięki np. blokowaniu pakietów usług, które nie są potrzebne w sieci wewnętrznej). [1]



Rys 4 Zapora z dwoma filtrami pakietów i bramą aplikacyjną

3. Proces budowania zapory sieciowej

Proces budowania czy stawiania zapory sieciowej można podzielić na kilka etapów:

1. Planowanie konfiguracji zapory sieciowej

   Do zadań tych należy rozpoznanie topologii sieci oraz potrzeb w zakresie aplikacji i protokołów. Polegać ono będzie na analizie topologii sieci pod kątem bezpieczeństwa, na zidentyfikowaniu systemów operacyjnych i aplikacji działających w sieci.

2. Zdefiniowanie reguł dostępu do zasobów sieciowych

W tym etapie należy określić kto i w jaki sposób ma dostęp do sieci i jej zasobów. Reguły muszą odpowiednio być dostosowane do posiadanej infrastruktury. Oznacza to uwzględnienie stosownych platform sprzętowych, czy protokołów sieciowych.

3. Znalezienie zapory odpowiedniej dla naszych potrzeb

W oparciu o zdobyte informacje, wykonane analizy i ustalone reguły dostępu do zasobów można właściwie wybrać potrzebna zaporę sieciową.

4. Właściwa instalacja i konfiguracja zapory

5. Drobiazgowe przetestowanie zapory

Testowanie zapory powinno odbyć się w dwóch etapach. W pierwszym należy przeprowadzić testowanie zasad korzystania z sieci przez użytkowników zewnętrznych. W drugim testujemy wewnętrzne reguły korzystania z sieci. Oba etapy należy wykonać dokładnie, ponieważ jest to ostatnia czynność przed włączeniem zapory do sieci. [1]

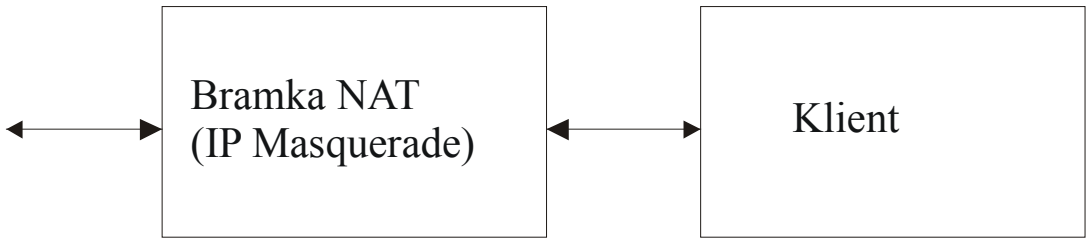
4. TRANSLACJA ADRESÓW

W większości zapór sieciowych można uruchomić mechanizm translacji adresów. Translacja adresów (*Network Address Translation - NAT*) jest formą maskowania rzeczywistych adresów urządzeń z ochranianej sieci. Umożliwia przydzielenie komputerom z sieci wewnętrznej adresów z puli adresów nie rejestrowanych w sieci Internet (RFC 1597) oraz zapewnia tym komputerom możliwość dwustronnego komunikowania się z komputerami sieci Internet. NAT umożliwia rozbudowę i rekonfigurację sieci TCP/IP bez obawy o wyczerpanie się oficjalnie przyznanych adresów IP. Dodatkowo umożliwia ukrycie wewnętrznej struktury sieci przed światem zewnętrznym i dostęp z zewnątrz tylko do wybranych serwerów.

W jądrze Linuxa v. 2.2.x funkcja ta dostępna jest pod nazwą *IP Masquerade*.

Stacja kliencka powinna zdefiniować bramę NAT jako swój *gatway*. Jeżeli tak nie jest, to brama NAT powinna funkcjonować jako *serwer proxy arp*.

Pakiet pochodzący od klienta otrzymuje nowy numer portu źródłowego i adres źródłowy. W takiej postaci jest wysyłany. Brama zapamiętuje zrealizowane przekształcenie. Gdy pakiet wraca to jest rozpoznawany jako przekształcony. Przywracany jest wówczas oryginalny adres klienta i pakiet trafia do klienta. [1]



Rys 5 Translacje adresów

5. MECHANIZMY SYSTEMU OPERACYJNEGO LINUX UMOŻLIWIAJĄCE ZBUDOWANIE ZAPORY SIECIOWEJ.

W standardowej wersji dystrybucyjnej systemu Linux zawarte są podstawowe narzędzia umożliwiające skonstruowanie zapory sieciowej. W przykładach opisana jest konfiguracja dla dystrybucji *RedHat*. W przypadku wykorzystania innej dystrybucji mogą wystąpić w konfiguracji pewne różnice. W jądrze w wersji 2.0 dostępny był mechanizm *IP Firewall*, w wersji 2.2 został on zastąpiony przez mechanizm *IP Chains*. W wersji 2.4 udostępniony jest mechanizm *IP Tables*.

Pakiet *ipchains* udostępnia trzy mechanizmy przydatne przy budowaniu zapory sieciowej:

- filtrowanie pakietów,
- maskowanie adresów IP,
- przeźroczysty serwer proxy.

Filtrowanie pakietów polega na selekcji pakietów przychodzących i wychodzących, ograniczając obustronną komunikację między siecią wewnętrzną, a zewnętrzną. Zazwyczaj polega to głównie na zablokowaniu wejścia do sieci wewnętrznej, poza kilkoma wybranymi usługami. Aby zapewnić wysoki stopień bezpieczeństwa sieci wewnętrznej, należy fizycznie oddzielić ją od sieci zewnętrznej. W takim przypadku dość naturalne jest skonfigurowanie *firewalla* jako *routera* dla całej sieci wewnętrznej.

Maskowanie adresów IP (*masquerating*) polega na zmienieniu adresów IP w przesyłanych pakietach. *Firewall* przechwytuje wszystkie pakiety wysyłane przez klientów z sieci wewnętrznej. Następnie jako adres źródłowy ustawia swój adres i wysyła tak zmienione pakiety do sieci zewnętrznej. Po odebraniu pakietu z odpowiedzią, adres docelowy zostanie zmieniony na adres klienta i pakiet zostanie przesłany do sieci wewnętrznej. W ten sposób komputery w sieci lokalnej są zupełnie niewidzialne dla świata zewnętrznego, chociaż mogą dokonywać połączeń z komputerami zewnętrznymi. Dzięki temu można podłączyć komputery w sieci lokalnej do Internetu, nawet jeśli nie mają one oficjalnie zarejestrowanych adresów IP, a używają adresów tzw. Klasy publicznej 192.168.x.y.

Przeźroczysty serwer proxy (*transparent proxy server*) umożliwia przekierowywanie wybranych pakietów do portów lokalnych *firewalla*. W ten sposób pakiety zamiast do miejsca przeznaczenia trafiają do zapory sieciowej, w której może funkcjonować serwer określonej usługi. [1]

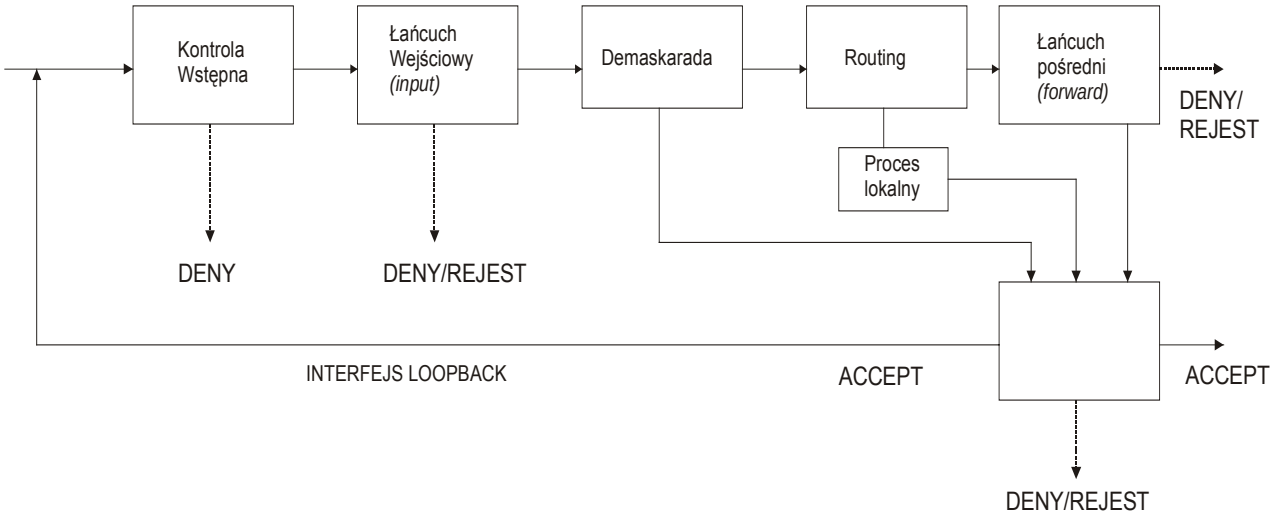
5.1 Przepływ pakietów w systemie Linux

Istotną rzeczą dla zrozumienia działania firewalla w systemie Linux jest poznanie drogi przepływu pakietów. Ilustruje to rys. 6.

Łańcuch (*chain*) to zbiór reguł, którym powinien odpowiadać pakiet. Jeżeli nagłówek pakietu spełnia warunek zdefiniowany w regule, to wykonywana jest akcja określona w tej regule. Akcja może określać przeznaczenie pakietu do kolejnego łańcucha lub działania specjalne. Do działań specjalnych należą:

- ACCEPT oznacza przepuszczenie pakietu.
- DENY oznacza odrzucenie pakietu.
- REJECT oznacza odrzucenie pakietu i powiadomienie poprzez ICMP o tym fakcie nadawcy.

- MASQ dotyczy łańcucha pośredniego i łańcuchów użytkownika, oznacza maskowanie pakietu poprzez zastąpienie adresu nadawcy adresem lokalnym. Przychodzące pakiety zwrotne stanowiące odpowiedz na pakiety maskowane, będą automatycznie rozpoznawane i demaskowane.
- REDIRECT dotyczy tylko łańcucha wejściowego i łańcuchów użytkownika, oznacza przekierowanie pakietu do gniazda lokalnego, czyli do lokalnego serwera proxy.



Rys 6 Przepływ pakietów w systemie Linux

Jeżeli reguła nie dotyczy danego pakietu, to sprawdzana jest następna reguła. Jeżeli nagłówek pakietu nie spełnia warunku w żadnej regule, to jądro uwzględnia zdefiniowaną ogólnie przyjętą strategię danego łańcucha.

Kontrola wstępna obejmuje sprawdzenie m.in. sum kontrolnych oraz poprawności konstrukcji pakietów docierający z sieci wewnętrznych i zewnętrznych. Po kontroli sprawdzane są reguły zdefiniowane jako łańcuch wejściowy.

Demaskarada – jeżeli pakiet zawiera odpowiedź na pakiet, który przy wysłaniu podlegał maskaradzie, to teraz ma miejsce proces odwrotny i przekazanie pakietu bezpośrednio do łańcucha wyjściowego. Pakiety które nie podlegają maskaradzie, są przekazywane do modułu routingu.

Routing – badane jest pole odbiorcy dla określenia, czy pakiet powinien zostać przekazany procesowi lokalnemu, czy też skierowany do innego komputera. Po przejściu przez proces lokalny i łańcuch wyjściowy, pakiet może być skierowany do interfejsu *loopback* lub poprzez łańcuch pośredni i łańcuch wyjściowy skierowany na zewnątrz.

Definiowanie reguł filtrowania realizuje się poprzez polecenie *ipchains*. W starszych wersjach jądra (2.0.x) wykorzystywany był *ipfwadm*. [1]

5.2 Konfiguracja jądra systemu

Aby system operacyjny mógł wykonywać funkcje zapory sieciowej, jądro systemu musi

zostać skompilowane z odpowiednimi opcjami. Wybieranie opcji można zrealizować różnymi metodami. Jedna z nich, dość wygodna w użyciu polega na wykorzystaniu interfejsu *menuconfig*. Uruchamia się go poprzez polecenie *make menuconfig*. Na ekranie zostanie wyświetlona hierarchiczna lista opcji jądra, w której należy dokonać odpowiednich zaznaczeń. Interesujące nas opcje znajdują się w sekcji *Networking options*.

IP: Drop source routed frames (CONFIG_IP_NOSR)

Zwykle w transmitowanym pakiecie umieszczone są adresy IP źródła i przeznaczenia. *Routingiem* (czyli wyznaczaniem trasy przesyłania pakietu) zajmują się komputery zaangażowane w przesyłanie zwane *routerami*. One decydują, którą drogą dalej przesłać pakiet. Jednakże w protokole IP zawarta jest możliwość wyspecyfikowania pełnej drogi dla danego pakietu już przy jego wysłaniu. Pakiety w których w pełni określono drogę przesyłania określane są jako „trasowane według nadawcy". Albo inaczej jako pakiety z ustawioną opcją *routingu źródłowego*. Powstaje pytanie, czy przy nadejściu takiego pakietu, należy brać pod uwagę wymagania dotyczące trasy przesyłania, czy też pakiet taki należy odrzucić. Honorowanie trasy może wprowadzić kłopoty związane z bezpieczeństwem, wobec czego zaleca się dla tej opcji ustawić Y (yes).

Network firewalls (CONFIG_FIREWALL)

IP: firewalling (CONFIG_IP_FIREWALL)

Ustawienie tej opcji jest wymagane jeżeli wykorzystujemy protokół IP. Opcja ta wymagana jest także wówczas, gdy chcemy włączyć przeźroczyste proxy.

IP: forwarding/gatewaying(CONFIG_IP_FORWARD)

Opcja ta umożliwia wykorzystanie naszego komputera jako routera dla sieci lokalnej. W takim przypadku w komputerze są zainstalowane przynajmniej dwie karty sieciowe. Jądro nie jest w stanie wykryć więcej niż jednej karty sieciowej przy starcie komputera i należy je skonfigurować ręcznie. Jeśli komputer jest połączony do dwóch sieci należy w wybrać N (no).

Jeżeli topologia sieci jest bardziej skomplikowana, na przykład rozpatrywany komputer podłączone jest do trzech sieci oraz chcemy, aby funkcjonował jako zapora sieciowa pomiędzy dwoma z nich i jako *router* dla pozostałych, wówczas należy wybrać Y (yes) i włączyć opcję *IP firewalling*.

Jeśli zamierzamy używać mechanizmu *IP masquerading*, wówczas należy bezwzględnie wybrać Y. Podobnie postępujemy w przypadku , gdy chcemy skonfigurować komputer jako serwer SLIP lub serwer PPP, poprzez który uzyskiwać będziemy dostęp do Internetu. Odpowiedź Y musimy wybrać również w przypadku, gdy chcemy uruchomić proces *mrouted* realizujący *multicast routing*.

IP: masquerading (CONFIG_IP_MASQUERADE)

Jeśli chcemy realizować maskowanie adresów IP, to należy tę opcje ustawić. Aby używać maskowania, należy również włączyć opcje: Network firewalls, IP: forwarding/gatewaying, IP: firewalling. Korzystne chociaż niekonieczne jest także włączenie opcji: *IP always defragment*.

IP: transparent proxying (CONFIG_IP_TRANSPARENT_PROXY)

Opcja ta umożliwia w sposób przeźroczysty dla klientów przekierowanie określonych pakietów do lokalnego serwera, określanego jako transparent proxy server. Dzięki temu komputery są przekonane, że są podłączone z właściwym komputerem, podczas, gdy w rzeczywistości są podłączone z lokalnym serwerem proxy. Przekierowanie jest uaktywniane poprzez zdefiniowanie, przy użyciu narzędzia *ipchains* specjalnych reguł wejściowych dla

zapory sieciowej.

IP: accouting (CONFIG_IP_ACCT)

Włączenie tej opcji uaktywnia rejestrowanie ruchu w sieci IP i umożliwia generowanie różnych statystyk w tym zakresie. Zarejestrowane dane dostępne są w pliku */proc/net/ip_acct*. Dokładny zakres rejestrowanych informacji można zdefiniować przy pomocy narzędzia *ipchains*.

IP: ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP)

Maskowanie włączone przez opcję CONFIG_IP_MASQUERADE obsługuje tylko pakiety TCP i UDP (oraz błędy ICMP dla istniejących połączeń). Omawiana opcja włącza dodatkową obsługę maskowania pakietów ICMP.

IP: lpautofw masquerading (CONFIG_IP_MASQUERADE_IPAUTOFW)

Napisany przez Richarda Lynch program *lpautofw* pozwala na maskowanie protokołów, które do tej pory nie były w pełni obsługiwane.

Kernel/User network link driver (CONFIG_IP_FIREWALL_NETLINK)

Włączony przez tą opcję sterownik pozwala na dwustronną komunikację pomiędzy pewnymi częściami jądra lub modułami i procesami użytkowymi. Procesy użytkowe uzyskują możliwość czytania i zapisywania danych do specjalnych plików znakowych o numerze głównym 36 obecnych w katalogu */dev*. Opcję należy włączyć, jeśli chcemy używać serwisu *arpd*, który pozwala utrzymać względnie małą wewnętrzną pamięć ARP (odwzorowanie pomiędzy adresami IP i adresami sprzętowymi w sieci lokalnej). [1]

5.3 Dodatkowa konfiguracja systemu

Aby możliwe było przekazywanie pakietów, co jest niezbędne dla wykonania maskowania adresów IP, należy uaktywnić ten mechanizm zmieniając w pliku */etc/sysconfig/network* linię z opcją FORWARD_IPV4 na: FORWARD_IPV4=yes.

Aby zapora sieciowa zbudowana na Linuxie dobrze realizowała swoje funkcje, należy spełnić jeszcze kilka dodatkowych wymagań. Oto kilka dodatkowych wymagań:

- Zainstalować niezbędne serwery proxy, np. dla usług ftp i http. Można je znaleźć np.: w pakiecie Trusted Information System Firewall Toolkit (TIS Toolkit). Po zainstalowaniu należy zdefiniować reguły ich wykorzystania.
- Usunąć zbędne i niepewne programy usługowe, które zostały standardowo zainstalowane w systemie, np.: *NFS, rexec, rlogin, rsh, telnet, ftp*.
- Połączenia z komputerem pełniącym funkcje zapory sieciowej powinny odbywać się tylko szyfrowanymi kanałami wymiany informacji, przy użyciu takich programów jak *ssh*.
- Główny demon sieciowy *inetd* należy zupełnie zrekonfigurować: niektóre ze standardowych funkcji (np.: echo) należy wyłączyć, ponieważ mogą one posłużyć do wykonania ataków typu *Denial-of-Service*.
- Jeżeli istnieje potrzeba instalacji serwera poczty elektronicznej, należy poważnie zastanowić się czy nie zrezygnować z programu *Sendmail,* znanego z licznych luk, na rzecz innego uważanego za bardziej bezpieczny, np.: *small*, czy *qmail*.
- Należy zabezpieczyć system przed niepożądanymi zmianami w plikach konfiguracyjnych, bądź binarnych. Można to wykonać przy pomocy programu

Tripwire, który pozwala wykryć zmiany w plikach i katalogach. [1]

5.4 Przykład 1

Przykładowa sieć ma topologię przedstawioną na rys. 7.

Charakterystyka sieci:

- Komputer pełniący funkcję bramy w zaporze sieciowej wyposażony jest w jeden interfejs sieciowy o numerze IP z puli światowej,
- Router jest skonfigurowany w sposób zapewniający tylko do i od komputera-bramy w zaporze sieciowej,
- Komputery klienckiej w sieci wewnętrznej mają przydzielone adresy IP z puli prywatnej, wobec czego ich adresy nie są przepuszczane przez router,
- Zapora sieciowa pełni funkcję bramki dla komputerów klienckich, wykonuje dla nich maskowanie adresów IP oraz proste filtrowanie (tzn.: z domyślnym przepuszczaniem pakietów) polegające na blokowaniu niektórych usług i adresów.

Konfiguracja routera nie zostanie tutaj przedstawiona, gdyż jest ona bardzo mocno zależna od stosowanego sprzętu. Konfiguracja interfejsu sieciowego w komputerze-bramie może być wykonana podczas instalacji systemu lub też przez modyfikację pliku: */etc/sysconfig/network-scripts/ifcfg-eth0*. W pliku tym zapisane są podstawowe dane konfiguracyjne konkretnego interfejsu. Zawartość tego pliku w naszym przypadku musi być następująca:

DEVICE=eth0
IPADDR=148.81.116.83
NETMASK=255.255.255.0
NETWORK=148.81.116.0
BROADCAST=148.81.116.255
ONBOOT=yes

Rys 7 Topologia sieci z przykładu 1

Oprócz tego należy jeszcze zdefiniować „alias” dla interfejsu eth0, o numerze IP 192.168.1.254, aby możliwa była komunikacja z komputerami sieci wewnętrznej. Wykonujemy to poleceniem:

Ifconfig eth0:0 192.168.1.254

Polecenie to powinno zostać oczywiście wpisane do skryptów startowych systemu, np.: do */etc/rc.d/rc.sysinit*

Tablica routingu w komputerze-bramie powinna mieć postać:

Destination	***Gateway***	***Genmask***	***Flags***	***Metric***	***Ref***	***Use***	***Iface***
192.168.1.0	*	*255.255.255.0*	*U*	*0*	*0*	*0*	*eth0*
148.81.116.0	*	*255.255.255.0*	*U*	*0*	*0*	*0*	*eth0*
127.0.0.0	*	*255.0.0.0*	*U*	*0*	*0*	*0*	*lo*
Default	*148.81.116.81*	*0.0.0.0*	*UG*	*0*	*0*	*0*	*eth0*

Tylko dodanie trasy domyślnej wymaga dodatkowego omówienia, ponieważ pozostałe linie tablicy routingu są tworzone automatycznie przy starcie sytemu, jeśli interfejs sieciowy jest prawidłowo skonfigurowany. Tę trasę domyślną określamy poleceniem:

route add default gw 148.81.116.81

lub modyfikujemy odpowiednią linię pliku */etc/sysconfig/network* :

GATEWAY=148.81.116.81

Spowoduje to automatyczną konfigurację tablicy routingu przy starcie systemu.

Maskowanie adresów IP uruchamiamy następującym poleceniem:

Ipchains –A forward –j MASQ –s 192.168.1.0/24 | 192.168.1.0/24

Co oznacza: wykonaj maskowanie dla wszystkich połączeń o adresie źródłowym z zakresu 192.168.1.1+ 254 i adresie docelowym spoza tego zakresu.

Kolejnym etapem konfiguracji zapory sieciowej jest ustawienie filtrowania. Zakładamy, że chcielibyśmy zablokować użytkownikom w sieci wewnętrznej możliwość korzystania z protokołu http i telnet oraz możliwość jakiejkolwiek łączności z komputerem o numerze IP 192.81.116.98; znajdującym się poza zaporą. Wykonamy to następującym zestawem poleceń:

ipchains –A input –j DENY –s 192.168.1.0./24 –p tcp –dport http

ipchains –A input –j DENY –s 192.168.1.0./24 –p tcp –dport telnet

ipchains –A input –j DENY –s 192.168.1.0./24 –d 148.81.116.98

Po wydaniu tych poleceń, reguły zapory wylistowane przy pomocy polecenia *ipchains –L* powinny wyglądać następująco:

Chain input (policy ACCEPT):

Target	***prot***	***opt***	***source***	***destination***	***ports***
DENY	*tcp*	*------*	*192.168.1.0/24*	*anywhere*	*any->www*
DENY	*tcp*	*------*	*192.168.1.0/24*	*anywhere*	*any->telnet*

DENY	*all*	*------*	*192.168.1.0/24*	*148.81.116.98*	*n/a*

Chain forward (policy ACCEPT):

Target	***prot***	***opt***	***source***	***destination***	***ports***
MASQ	*all*	*------*	*192.168.1.0/24*	*192.168.1.0/24*	*n/a*

Chain output (policy ACCEPT):

Aby system był w ten sposób skonfigurowany po starcie systemu, przedstawione polecenia należy wpisać do skryptu startowego systemu, np.: */etc/rc.d/rc.sysinit*. [1]

5.5 Przykład 2

W kolejnym przykładzie przyjęliśmy topologię sieci przedstawioną na rys. 8.

Charakterystyka sieci:

- komputer w zaporze wyposażony jest w trzy interfejsy sieciowe: dla połączenia z Internetem i połączenia z sieciami lokalnymi; pełni więc także rolę routera,
- jedna z podsieci zawiera serwery z adresami IP z puli światowej,
- druga podsieć zawiera tylko komputery klienckie z adresami z puli prywatnej,
- zapora sieciowa wykonuje maskowanie adresów IP dla komputerów klienckich,
- zapora sieciowa wykonuje filtrowanie, z domyślnym blokowaniem pakietów, polegające na przepuszczaniu pakietów tylko głównych usług sieciowych.

Aby było możliwe wykorzystanie kilku interfejsów sieciowych, Linux musi je obsługiwać, tzn. musi posiadać moduły obsługujące konkretne karty sieciowe. Sprawdzenia, czy system rozpoznał posiadane przez nas karty sieciowe, można dokonać poprzez przegląd komunikatów jądra zapisanych podczas startu systemu do pliku */var/log/messages*. W czasie testowania niniejszego przykładu testowane były karty sieciowe PCI zgodne ze standardem NE2000, które system rozpoznał bez problemu.

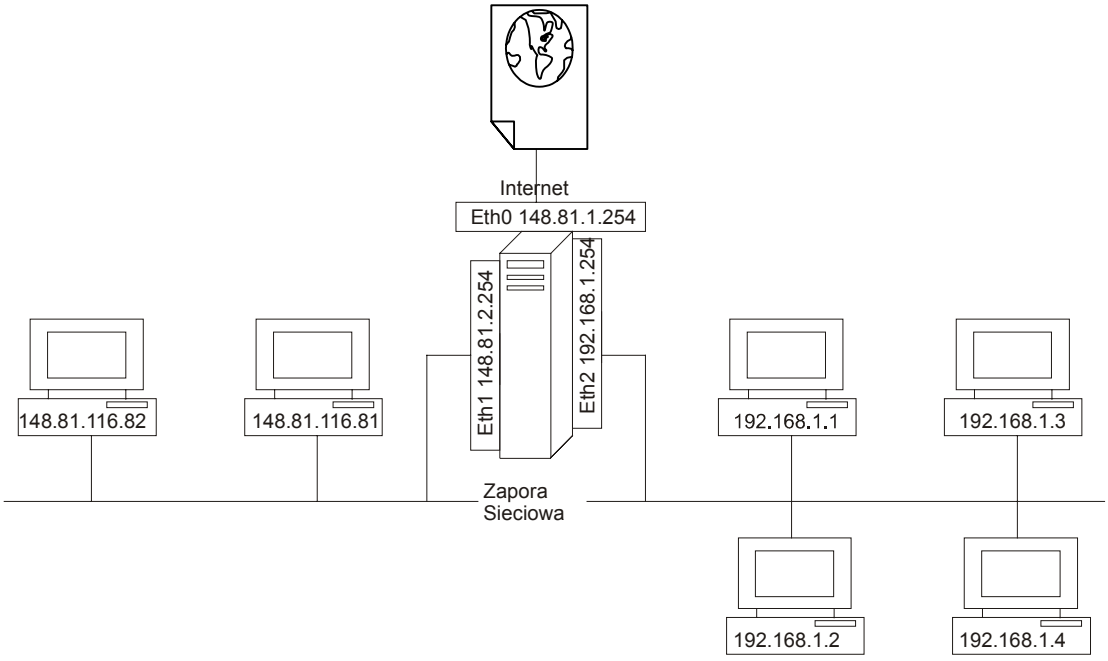
Następnie dla każdego interfejsu sieciowego należy utworzyć plik */etc/sysconfig/network-scripts/ifcfg-ethx* (gdzie x to numer interfejsu). W pliku tym zapisane są podstawowe dane konfiguracyjne konkretnego interfejsu. W omawianym przypadku zawartości tych plików są następujące:

DEVICE=eth0
IPADDR=148.81.1.254
NETMASK=255.255.255.0
NETWORK=148.81.1.0
BROADCAST=148.81.1.255
ONBOOT=yes

Plik */etc/sysconfig/network-scripts/ifcfg-eth1*:
DEVICE=eth1
IPADDR=148.81.2.254
NETMASK=255.255.255.0
NETWORK=148.81.2.0
BROADCAST=148.81.2.255

ONBOOT=yes

Plik */etc/sysconfig/network-scripts/ifcfg-eth2*:
DEVICE=eth2
IPADDR=148.81.1.254
NETMASK=255.255.255.0
NETWORK=148.81.1.0
BROADCAST=148.81.1.255
ONBOOT=yes



Rys 8 Topologia sieci z przykładu 2

Tablica routingu w komputera w zaporze wyświetlana poleceniem *route* musi wyglądać następująco:

Destination	***Gateway***	***Genmask***	***Flags***	***Metric***	***Ref***	***Use***	***Iface***
148. 81.1.0	*	*255.255.255.0*	*U*	*0*	*0*	*0*	*eth0*
148.81.2.0	*	*255.255.255.0*	*U*	*0*	*0*	*0*	*eth1*
192.168.1.0	*	*255.0.0.0*	*U*	*0*	*0*	*0*	*eth2*
127.0.0.0	*	*255.0.0.0*	*U*	*0*	*0*	*0*	*lo*
default	*148.81.1.1*	*0.0.0.0*	*UG*	*0*	*0*	*0*	*eth0*

Sposób dodania trasy domyślnej opisano w przykładzie pierwszym. Pozostałe linie tablicy routingu są tworzone automatycznie przy starcie systemu. Domyślną trasę pakietów obliczamy zakładając, że bramka (gateway) ma numer IP 148.81.1.1.

Przyjmujemy następujące założenia odnośnie filtrowania pakietów w zaporze:

- komputer na którym została uruchomiona zapora sieciowa udostępnia następujące usługi:
  - ssh – do zdalnej pracy na zaporze (np. : w celu dokonania zmian konfiguracji)

z każdego miejsca w Internecie,
- DNS – tylko do komputerów z obu podsieci wewnętrznych,

- serwery udostępniają następujące usługi:
  - http
  - ssh
  - smpt
  - DNS
  - telnet
  - POP3
- użytkownicy pracujący na komputerach klienckich mogą korzystać z następujących usług:
  - http
  - ssh
  - smpt
  - DNS
  - telnet – tylko do serwerów w drugiej podsieci
  - POP3 – tylko do serwerów w drugiej podsieci
  - dostarczanych przez protokół ICMP, np.: echo

Maskowanie adresów IP dla komputerów klienckich uruchamiamy poleceniem:

Ipchains –A forward –j MASQ –s 192.168.1.0/24 | 192.168.1.0/24

Co oznacza: wykonaj maskowanie dla wszystkich połączeń o adresie źródłowym z zakresu 192.168.1.1+ 254 i adresie docelowym spoza tego zakresu.

Dla wbudowanego łańcucha „*input*" określającego reguły dostępu do zapory sieciowej przyjmujemy jako politykę domyślną – odrzucanie pakietów (DENY). Dla wbudowanego łańcucha „*forward*" określającego reguły maskowania oraz filtrowania ruchu pomiędzy wewnętrzną siecią serwerów, wewnętrzną siecią klientów oraz Internetem, przyjmujemy jako politykę domyślną – odrzucanie pakietów (DENY). Wbudowany łańcuch „output' nie będzie odfiltrowywał żadnych pakietów. Jako politykę domyślną przyjmujemy przepuszczanie pakietów (ACCEPT). Realizujemy to poleceniami:

ipchains-P input DENY

ipchains-P forward DENY

ipchains-P output ACCEPT

Ustawienie reguł filtrowania dla pakietów przychodzących i wychodzących przez interfejs loopback jest konieczne, ponieważ niektóre programy mogą korzystać z tego interfejsu. Nie stanowi to luki w systemie bezpieczeństwa, ponieważ interfejs loopback nie jest dostępny z zewnątrz. Ustawienie to realizujemy poleceniem:

Ipchains-A intput –j ACCEPT -i lo

Ustawienie reguł filtrowania dla łańcucha input:

- pozwolenie na korzystanie z ssh i DNS

ipchains –A input –j ACCEPT –d 192.168.1.254 –p tcp –dport ssh
ipchains –A input –j ACCEPT –d 148.81.1.254 –p tcp –dport ssh
ipchains –A input –j ACCEPT –d 148.81.2.254 –p tcp –dport ssh
ipchains –A input –j ACCEPT –d 148.81.2.254 –p tcp –dport domain
ipchains –A input –j ACCEPT –d 148.81.2.254 –p udp –dport domain

ipchains –A input –j ACCEPT –d 192.168.1.254 –p tcp –dport domain
ipchains –A input –j ACCEPT –d 192.168.1.254 –p udp –dport domain
przepuszczenie pakietów będących odpowiedziami na połączenia tcp:
ipchains –A input –j ACCEPT –p tcp | y

- przepuszczenie odpowiedzi dla protokołu udp:

ipchains –A input –j ACCEPT –d 148.81.2.254 –p udp –sport domain
ipchains –A input –j ACCEPT –d 192.168.1.254 –p udp –sport domain

- przepuszczenie dla pozostałych pakietów, nie kierowanych bezpośrednio do zapory sieciowej; należy tu użyć dodatkowego łańcucha inputnet, ponieważ w poleceniu może wystąpić tylko jedna opcja –d:

ipchains –N inputnet
ipchains –A inputnet –j ACCEPT –d | 192.168.1.254
ipchains –A input –j inputnet –s 192.168.1.0/24
ipchains –A input –j inputnet –d 192.168.1.0/24
ipchains –A input –j inputnet –s 148.81.1.0/24
ipchains –A input –j inputnet –d 148.81.1.0/24
ipchains –A input –j inputnet –s 148.81.1.0/24
ipchains –A input –j inputnet –d 148.81.1.0/24

Ustawienie reguł filtrowania dla łańcucha *forward*:

- blokowanie pakietów pochodzących z lub kierowanych do sieci lokalnych (należy pamiętać o umieszczeniu reguły maskowania IP na początku łańcucha forward):

ipchains –A forward –j DENY –s 127.0.0.0/8
ipchains –A forward –j DENY –d 127.0.0.0/8
ipchains –A forward –j DENY –s 192.168.0.0/16
ipchains –A forward –j DENY –d 192.168.0.0/16

- blokowanie pakietów rozgłoszeniowych:

ipchains –A forward –j DENY –d 192.168.1.255
ipchains –A forward –j DENY –d 148.81.1.255
ipchains –A forward –j DENY –d 148.81.2.255
ipchains –A forward –j DENY –d 255. 255. 255.255

- przepuszczanie pakietów dotyczących korzystania z wybranych usług:

ipchains –A forward –j ACCEPT –p tcp | -y
ipchains –A forward –j ACCEPT –p icmp
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p tcp –-dport http
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p tcp –-dport ssh
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p tcp –-dport smtp
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p tcp –-dport domain
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p udp -b –-dport domain
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p tcp –-dport http
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p tcp –-dport ssh

ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p tcp –-dport smtp
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p tcp –-dport domain
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p udp -b –-dport domain
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –s 192.168.1.0/24 –p tcp –-dport telnet
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –s 192.168.1.0/24 -p tcp –-dport pop3

Po wydaniu przedstawionych poleceń reguły wyświetlone poleceniem *ipchains –L* powinny wyglądać następująco:

Chain input (policy DENY):

target	***rot***	***opt***	***source***	***destination***	***ports***
ACCEPT	*all*	*-------*	*anywhere*	*anywhere*	*n/a*
ACCEPT	*tcp*	*| -y*	*anywhere*	*anywhere*	*any->any*
ACCEPT	*tcp*	*-------*	*anywhere*	*192.168.1.254*	*any->ssh*
ACCEPT	*tcp*	*-------*	*anywhere*	*148.81.1.254*	*any->ssh*
ACCEPT	*tcp*	*-------*	*anywhere*	*148.81.2.254*	*any->ssh*
ACCEPT	*tcp*	*-------*	*anywhere*	*148.81.2.254*	*any->domain*
ACCEPT	*udp*	*-------*	*anywhere*	*148.81.2.254*	*any->domain*
ACCEPT	*udp*	*-------*	*148.81.2.254*	*anywhere*	*domain->any*
ACCEPT	*tcp*	*-------*	*anywhere*	*192.168.1.254*	*any->domain*
ACCEPT	*udp*	*-------*	*anywhere*	*192.168.1.254*	*any->domain*
ACCEPT	*udp*	*-------*	*192.168.1.254*	*anywhere*	*domain->any*
Inputnet	*all*	*-------*	*192.168.1.0/24*	*anywhere*	*n/a*
Inputnet	*all*	*-------*	*anywhere*	*192.168.1.0/24*	*n/a*
Inputnet	*all*	*-------*	*148.81.1. 0/24*	*anywhere*	*n/a*
Inputnet	*all*	*-------*	*anywhere*	*148.81.1. 0/24*	*n/a*
Inputnet	*all*	*-------*	*148.81.1. 0/24*	*anywhere*	*n/a*
Inputnet	*all*	*-------*	*anywhere*	*148.81.1. 0/24*	*n/a*
ACCEPT	*tcp*	*-------*	*anywhere*	*192.168.1.254*	*any->telnet*

Chain forward (policy DENY):

target	***rot***	***opt***	***source***	***destination***	***ports***
MASQ	*all*	*-------*	*192.168.1.0/24*	*192.168.1.0/24*	*n/a*
DENY	*all*	*-------*	*127.0.0.0/8*	*anywhere*	*n/a*
DENY	*all*	*-------*	*anywhere*	*127.0.0.0/8*	*n/a*
DENY	*all*	*-------*	*192.168.0.0/16*	*anywhere*	*n/a*
DENY	*all*	*-------*	*anywhere*	*192.168.0.0/16*	*n/a*
DENY	*all*	*-------*	*anywhere*	*192.168.1.255*	*n/a*
DENY	*all*	*-------*	*anywhere*	*148.81.1.255*	*n/a*
DENY	*all*	*-------*	*anywhere*	*148.81.2.255*	*n/a*
DENY	*all*	*-------*	*anywhere*	*255.255.255.255*	*n/a*
fornet	*all*	*-------*	*192.168.1.0/24*	*anywhere*	*n/a*

fornet	*all*	*-------*	*anywhere*	*148.81.2.0/24*	*n/a*
ACCEPT	*tcp*	*-------*	*192.168.1.0/24*	*148.81.2.0/24*	*any->telnet*
ACCEPT	*tcp*	*-------*	*192.168.1.0/24*	*148.81.2.0/24*	*any->pop3*

Chain output (policy ACCEPT):
Chain inputnet (6 references):

target	***rot***	***opt***	***source***	***destination***	***ports***
ACCEPT	*all*	*--------*	*anywhere*	*192.168.1.254*	*n/a*

Chain fornet (2 references):

target	***rot***	***opt***	***source***	***destination***	***ports***
ACCEPT	*tcp*	*| -y*	*anywhere*	*anywhere*	*any-> any*
ACCEPT	*icmp*	*-------*	*anywhere*	*anywhere*	*any-> any*
ACCEPT	*tcp*	*-------*	*anywhere*	*anywhere*	*any->www*
ACCEPT	*tcp*	*-------*	*anywhere*	*anywhere*	*any->ssh*
ACCEPT	*tcp*	*-------*	*anywhere*	*anywhere*	*any->smtp*
ACCEPT	*tcp*	*-------*	*anywhere*	*anywhere*	*any-> domain*
ACCEPT	*udp*	*-------*	*anywhere*	*anywhere*	*any-> domain*
ACCEPT	*udp*	*-------*	*anywhere*	*anywhere*	*domain-> any*

Aby taka konfiguracja była realizowana automatycznie przy starcie systemu, należy utworzyć odpowiedni skrypt i jego wywołanie należy umieścić w skrypcie startowym, np.: */etc/rc.d/rc.sysinit*. Należy tu dodać, że warto byłoby tak zmienić skrypty startowe systemu, aby w pierwszej kolejności ustawić blokowanie wszystkich pakietów. W następnej kolejności uruchomić interfejsy sieciowe, a potem wywołać skrypt realizujący właściwą konfigurację filtrowania. Treść takiego skryptu konfiguracyjnego była by następująca:

```
ipchains-P input DENY
ipchains-P forward DENY
ipchains-P output ACCEPT
ipchains-A intput –j ACCEPT -i lo
ipchains-A intput –j ACCEPT –p tcp | -y
ipchains –A input –j ACCEPT –d 192.168.1.254 –p tcp –dport ssh
ipchains –A input –j ACCEPT –d 148.81.1.254 –p tcp –dport ssh
ipchains –A input –j ACCEPT –d 148.81.2.254 –p tcp –dport ssh
ipchains –A input –j ACCEPT –d 148.81.2.254 –p tcp –dport domain
ipchains –A input –j ACCEPT –d 148.81.2.254 –p udp –b –dport domain
ipchains –A input –j ACCEPT –d 192.168.1.254 –p tcp –dport domain
ipchains –A input –j ACCEPT –d 192.168.1.254 –p udp –b –dport domain
ipchains –N inputnet
ipchains –A inputnet –j ACCEPT –d | 192.168.1.254
ipchains –A input –j inputnet –s 192.168.1.0/24
ipchains –A input –j inputnet –d 192.168.1.0/24
ipchains –A input –j inputnet –s 148.81.1.0/24
ipchains –A input –j inputnet –d 148.81.1.0/24
ipchains –A input –j inputnet –s 148.81.1.0/24
ipchains –A input –j inputnet –d 148.81.1.0/24
```

ipchains –A forward –j MASQ –s 192.168.1.0/24 –d | 192.168.1.0/24
ipchains –A forward –j DENY –s 127.0.0.0/8
ipchains –A forward –j DENY –d 127.0.0.0/8
ipchains –A forward –j DENY –s 192.168.0.0/16
ipchains –A forward –j DENY –d 192.168.0.0/16
ipchains –A forward –j DENY –d 192.168.1.255
ipchains –A forward –j DENY –d 148.81.1.255
ipchains –A forward –j DENY –d 148.81.2.255
ipchains –A forward –j DENY –d 255. 255. 255.255
ipchains –N fornet
ipchains –A forward –j ACCEPT –p tcp | -y
ipchains –A forward –j ACCEPT –p icmp
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p tcp –-dport http
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p tcp –-dport ssh
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p tcp –-dport smtp
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p tcp –-dport domain
ipchains –A forward –j ACCEPT –s 192.168.1.0/24 –p udp -b –-dport domain
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p tcp –-dport http
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p tcp –-dport ssh
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p tcp –-dport smtp
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p tcp –-dport domain
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –p udp -b –-dport domain
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –s 192.168.1.0/24 –p tcp –-dport telnet
ipchains –A forward –j ACCEPT –d 148.81.2.0/24 –s 192.168.1.0/24 -p tcp –-dport pop3

[1]

LITERATURA

[1] Zbigniew Suski, Piotr Kołodziejczyk „Ochrona sieci lokalnej za pomocą zapory sieciowej“ BIULETYN INSTYTUTU AUTOMATYKI I ROBOTYKI WAT NR 14,2000

[2] Olaf Kirch, Terry Dawson „Linux-podręcznik administratora” 2000