Linux

iptables, nmap,

DMZ

Spis treści.

iptables – wprowadzenie.

Począwszy od jądra 2.4 Linux posiada wbudowane narzędzie do filtrowania pakietów iptables. We wcześniejszych wer. jądra funkcjonalność tą posiadał ipchains. Jednak miał on jeden mankament. Mianowicie nie potrafił rozróżniać stanu połączenia. Tą dodatkową, jakże przydatną funkcjonalność posiada iptables.

Iptables opiera się na trzech tabelach, które zawierają domyślnie zdefiniowane łańcuchy. Łańcuch to nic innego jak lista reguł. W iptables przechodzące pakiety są sprawdzane w następujący sposób (pseudo język): Jeśli nagłówek pakietu wygląda tak, to wykonaj z tym pakietem następującą akcję. Jeśli pierwsza reguła nie pasuje do pakietu, sprawdź kolejną itd., aż do ostatniej reguły dla tablicy. Jeśli żadna reguła nie pasuje do pakietu sprawdź ustawienia polityki ustanowionej dla tej tabeli i ją zastosuj. Jeżeli polityka ustawiona jest np. na „drop" pakiet jest blokowany, jeśli na „accept" pakiet jest akceptowany. Jak widać z powyższego **kolejność reguł ma kolosalne znaczenie na poprawne funkcjonowanie firewalla** opartego na iptables.

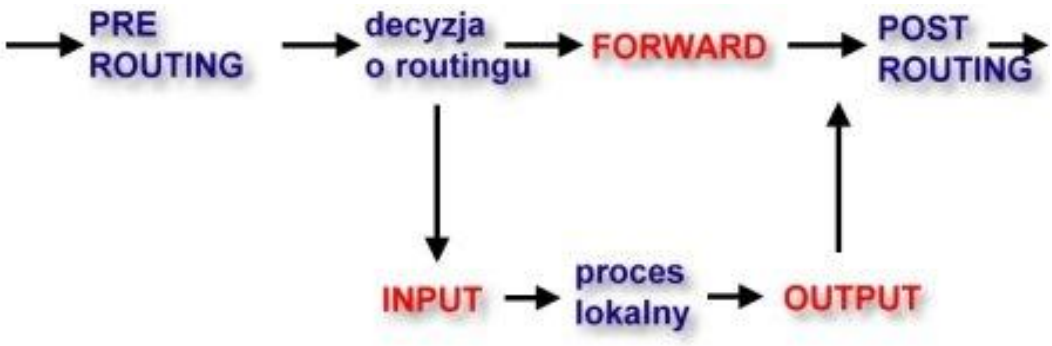
Warto dodać, że reguły wpisywane z konsoli, po restarcie systemu ulegają wyczyszczeniu, więc należy utworzyć sobie skrypt, który będzie się uruchamiał wraz ze startem systemu.

Wpis taki można dodać np. w pliku /etc/init.d/boot.local w formacie: /katalog/skrypt_firewall.sh lub przenieść skrypt z regułami do katalogu **/etc/init.d/**skrypt_firewall.sh, następnie wydać komendę: chkconfig --add skrypt_firewall.sh aby skrypt ten uruchamiał się przy starcie systemu.

Tabele iptables wraz z łańcuchami, oraz najczęściej definiowanymi akcjami.

Tabela	Łańcuchy	Najczęstsze akcje
Filter	INPUT, OUTPUT, FORWARD	ACCEPT, DROP, REJECT
Nat	PREROUTING, OUTPUT, POSTROUTING	SNAT, DNAT, REDIRECT
Mangle	INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING	MARK, TOS, TTL

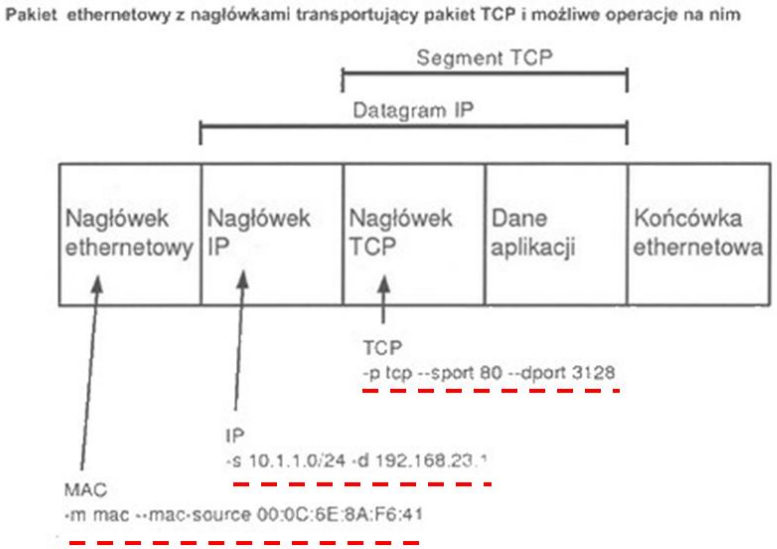
iptables – droga pakietu



Pakiet przychodzi z lewej strony. Po zweryfikowaniu sumy kontrolnej IP pakiet wpada do łańcucha NF_IP_PRE_ROUTING (w skrócie PREROUTING).Następnie zostaje określona trasa routingu dla pakietu tzn. zapada decyzja, czy pakiet jest skierowany do procesu lokalnego czy może musi być przesłany (routowany) gdzie indziej. Jeżeli pakiet jest pakietem o adresie docelowym mówiącym, że ma trafić do procesu lokalnego (działającego na maszynie z firewallem) jest on przesyłany do łańcucha NF_IP_LOCAL_IN (w skrócie INPUT). Jeżeli jednak pakiet ma trafić do innej maszyny musi zostać przesłany poprzez łańcuch NF_IP_FORWARD (w skrócie FORWARD).Łańcuch NF_IP_LOCAL_OUT jest wykorzystywany dla pakietów generowanych lokalnie i wysyłanych na inne maszyny w sieci.Kiedy pakiet wychodzi z firewalla musi przejść przez jeszcze jeden łańcuch zwany NF_IP_POST_ROUTING (w skrócie POSTROUTING). Pakiety przechodzące przez łańcuchy FORWARD i OUTPUT przechodzą przez ten ostatni łańcuch.

Źródło: http://linux.howto.pl/artykuly,linux-20-9-0.html

Pakiet ethernetowy z nagłówkami i możliwe operacje na nim



Iptables działa w następujących warstwach :

Łącza danych (Ethernet), sieciowej (IP), oraz transportowej (TCP).

Nagłówki pakietów – przegląd.

Nagłówek TCP

Source port			Destination port
Seqence numer			
Acknowledgment numer			
Data offset	Reserved	Flags	Window
Checksum			Urgens pointer
Options			
Data			

W modelu TCP/IP warstwa transportowa odpowiedzialna jest za dostarczenie pakietów do aplikacji docelowych. Osiągane jest to przez tzw. porty. Numer portu posiada strona nadająca (TCP Source Port), jak i odbierająca (TCP Destination Port). Pola Sequence Numer (numer sekwencji), oraz Acknowledgment Numer (numer potwierdzenia) odpowiadają za niezawodność dostawy.

Nagłówek IP wer. 4.

Version	IHL	Type of sernice (flags)	Total lenght	
Identyfication			Flags	Fragment offset
Time – to – live (TTL)		Protocol	Header checksum	
Source address				
Destination address				
Options				
Data				

Wśród wielu pól znajdują się dwa miejsca na 32-bitowe adresy IP. Source Address to adres IP komputera wysyłającego dane, natomiast Destination Address to adres IP komputera docelowego. Pole TTL Time to Live określa czas życia pakietu.

Nagłówek IP wer. 6

Version	Traffic class	Flow label	
Identyfication		Next header	Hop limit
Source address			
Destination address			
Data			

Source address Destination address (adres źródłowy, docelowy) – pole 128 bitowe zawierające adres źródłowy/docelowy pakietu.

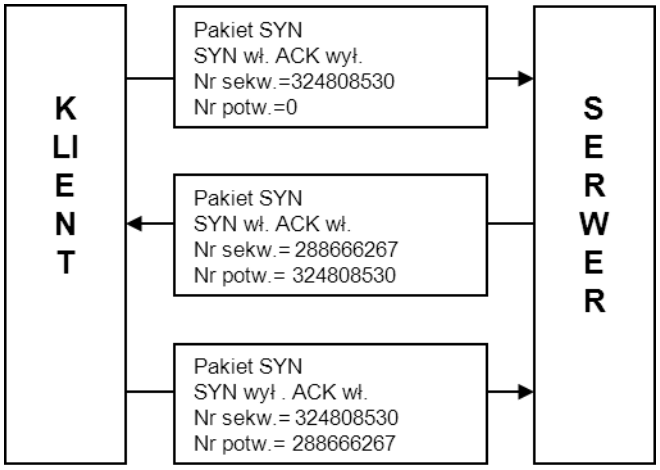
Nagłówek ICMP.

Internet Control Message Protocol (cd)

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             unused                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Internet Header + 64 bits of Original Data Datagram      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

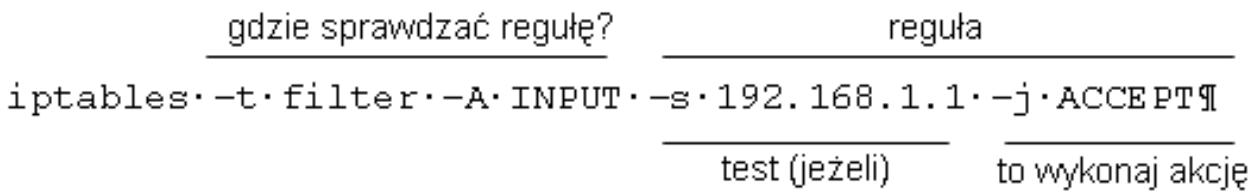
```
Message Types:
    0  Echo Reply
    3  Destination Unreachable    4  Source Quench
    5  Redirect                   8  Echo
   11  Time Exceeded             12  Parameter Problem
   13  Timestamp                 14  Timestamp Reply
   15  Information Request       16  Information Reply
```

Trójstopniowy proces ustanawiania połączenia sieciowego.



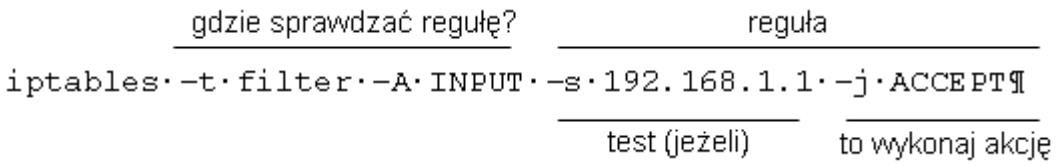
Powyższy proces ma wpływ na działanie modułu **state** oraz różnych metod skanowania portów np. za pomocą narzędzia **nmap**.

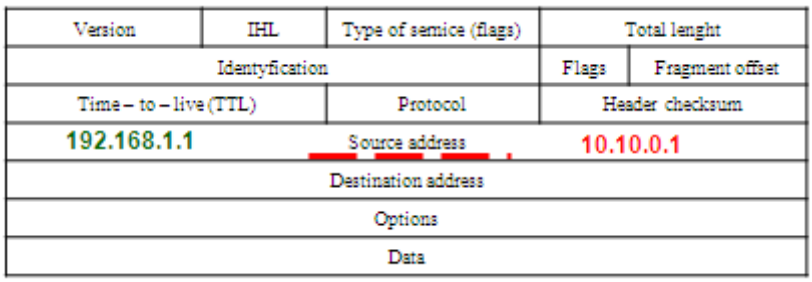
Składnia dodawania reguły filtrowania pakietów.



Jak widać na powyższym przykładzie składnia zaczyna się od wywołania iptables. Następnie definiujemy, która tablica ma być brana pod uwagę (nazwę tablicy poprzedzamy literą **–t**). Jeżeli parametr ten jest pominięty iptables domyślnie przyjmuje, że reguła odnosi się do tabeli FILTER), oraz jaki ruch sprawdzamy (czy ma być to ruch wchodzący do maszyny - INPUT, wychodzący - OUTPUT, a może rutowalny przez maszynę - FORWARD). Litera **–A** określa czynność wykonaną na tej regule (w tym przypadku jest to dodanie reguły na koniec łańcucha). W sekcji „reguła" na wstępie przeprowadzany jest test (w tym przypadku **–s**, sprawdzane jest źródło (source), z jakiego adresu IP przybył pakiet). Od wyniku przeprowadzonego testu podejmowane są dalsze akcje (w powyższym przykładzie **-j ACCEPT**, czyli jeżeli pakiet pasuje do reguły zaakceptuj go).

Zasada działania.





Jeżeli reguła wygląda jak powyżej to sprawdzany jest nagłówek pakietu IP, a dokładnie pole source address. Jeżeli pakiet pochodzi z adresu 192.168.1.1 to pakiet zostanie

przepuszczony. Jeżeli z innego adresu np. 10.0.0.1 to zostanie odrzucony, chyba, że kolejna reguła lub polityka łańcucha INPUT mówi inaczej.

Wyświetlenie aktualnych ustawień.

iptables –L –v

Uwaga !!! Wydanie polecenia iptables –L wyświetla ustawienia tabeli FILTER (ponieważ taka tabela jest domyślna jeżeli nie podamy opcji –t). Jeśli chcemy wyświetlić ustawienia innych tabel należy po opcji –t podać nazwę tej tabeli np.:
iptables –t nat –L lub iptables –t mangle -L

iptables -L -n --line-numbers
Przełączniki:
-L {łańcuch} - pozwala zobaczyć wszystkie reguły lub te z danego łańcucha,
-n - nie odpytuje DNS, podaje ip zamiast zmieniać je na hosty,
--line-numbers - dodatkowo wyświetla numery obok reguł dzięki czemu łatwiej je zmieniać.

Czynności możliwe do zastosowania na łańcuchu.

-L wyświetl ustawienia dla danego łańcucha. Jeśli łańcuch nie został podany wyświetlone zostaną reguły dla wszystkich łańcuchów z tabeli.
-F czyść regułę podanego łańcucha. Jeżeli łańcuch nie został podany wyczyszczone zostaną wszystkie reguły.
-N tworzy nowy łańcuch.
-P ustaw politykę dla łańcucha.
-X usuń łańcuch (zdefiniowany przez użytkownika).
-h wyświetl pomoc.

Czynności możliwe do wykonanie na regule.

-A dodaj regułę na końcu łańcucha.
-D usuń regułę z łańcucha.
-I wstaw regułę na początku łańcucha.
-R zastąp regułę.

Parametry reguły.

Definiując reguły możemy używać następujących parametrów:

-p protokół (TCP, UDP, ICMP).

-s adres IP i maska źródła pakietów.

-d adres IP i maska celu pakietów.

-j akcja wykonuje określone zadanie na regule (ACCEPT - akceptuj, DROP – odrzuć, REJECT – odrzuć i zwróć komunikat do nadawcy o odrzuceniu, SNAT, MASQUERADE – podmień adres źródłowy, DNAT – podmień adres docelowy).

-i interface wejściowy (dla łańcuchów INPUT, FORWARD, PREROUTING)

-o interface wyjściowy (dla łańcuchów OUTPUT, FORWARD, POSTROUTING)

! „wykrzyknik" odwraca znaczenie parametru którego poprzedza.

Moduł state.

-m state

Dodanie opcji **-m** do łańcucha **powoduje załadowanie modułu** w tym przypadku **state** który sprawdza stan pakietu.

Po załadowaniu modułu można opierać się na stanie połączeń dostępne opcje: **NEW** (połączenia nowo przychodzące), **RELATED** (połączenia powiązane z wcześniej nawiązanymi), **ESTABLISHED** (połączenia już ustanowione), **UNTRACKED** oraz **INVALID** oznaczają stany z reguły niepoprawne, uszkodzone pakiety.

Moduł MAC

Dodanie opcji **–m mac** do łańcucha powoduje załadowanie modułu, który umożliwia sprawdzenie adresu MAC źródła pakietu.

Przykład:
-m mac --mac-source 00:0C:0E:A2:4M:37

Pełny przykład:
iptables –A FORWARD –i eth0 –p tcp --dport 80 -m mac --mac-source 00:0C:0E:A2:4M:37 –m state --state NEW –j ACCEPT

Moduł ten działa tylko dla tabeli INPUT, dla pakietów bez rutowania (czyli praktycznie tylko w sieci lokalnej) !

Resetowanie połączeń wcześniej ustanowionych.

Niekiedy może zajść potrzeba aby zerwać istniejące połączenia, np. po rekonfiguracji firewalla i ponownym jego uruchomieniu. Możemy to zrobić następująco:
iptables –A INPUT –p **tcp** –j REJECT --reject-with tcp-reset
iptables –A INPUT –p **udp** –j REJECT --reject-with icmp-port-unreachable

Logowanie zdarzeń za pomocą iptables.

--log-prefix ['tekst'] - powoduje dopisanie prefiksu do interesującego nas loga dzięki czemu będzie je nam łatwiej rozróżnić.

--limit – opcja dostępna po wybraniu modułu limit, jeśli liczba będzie większa niż ta podana w danej regule wpis się nie dokona. Np.: --limit 10 (dodaj do laga tylko 10 wpisów na godzinę).

nmap - Testowanie Firewall-a.

Do wszystkich poniższych przykładów można dodać opcję **–v** (aby zobaczyć pełne „wyjście" tego polecenia).

Skanowanie typu "**tcp connect**" - to najbardziej podstawowa technika. Polega na przeprowadzeniu pełnego połączenia TCP z danym portem. Jako że podczas tzw. trójstronnego uzgadniania połączenia (SYN, SYN/ACK i ACK) żaden komputer pracujący w sieci nie może ignorować pakietów TCP/SYN - jest to również metoda całkiem skuteczna. Jeśli więc dany port jest otwarty i otrzyma od nmapa pakiet TCP z ustawioną flagą syn, odpowie pakietem SYN/ACK. Jeśli jest zamknięty odpowie RST/ACK.

nmap -sT adres_ip

Skanowanie **tcp syn** polega na wysłaniu pakietu z ustawioną flagą SYN. Nie jest jednak nawiązywane pełne połączenie, ponieważ jeśli zdalny host odpowie SYN/ACK (czyli port jest otwarty), atakujący nigdy nie wyśle ACK. W logach typowych firewalli nie będzie więc nawet śladów połączenia. W przypadku tej techniki posłużymy się opcją -sS:

nmap -sS adres_ip

Dla odmiany **tcp fin** to metoda polegająca na przesłaniu do zdalnego portu pakietu z flagą FIN (czyli takiego który zazwyczaj, w przypadku "kulturalnego i podręcznikowego" działania kończy uzgadnianie połączenia). Host powinien odpowiedzieć pakietem RST jeśli port jest zamknięty. Jest to tzw skanowanie ukryte:

nmap -sF adres_ip

Skanowanie typu **tcp ack** to również stosunkowo bezpieczna, ukryta metoda. Polega na wysłaniu od razu pakietu z flagą ACK (z pominięciem SYN i SYN/ACK). W tym przypadku wartość ACK jest sfałszowana, bo dotyczy połączenia które nigdy nie zostało nawiązane. Port zamknięty odpowie RST, port otwarty ...wcale nie odpowie.
Z pomocą nmap-a robimy to następująco:

nmap -sA adres_ip

tcp null to technika w ktorej do hosta wysyłane są pakiety bez żadnej flagi (SYN, SYN/ACK czy FIN). Na takie "zaczepki" port otwarty nie odpowie (czyli skaner wie że jest otwarty), port zamknięty natomiast odpowie zwykłym RST:

nmap -sN adres_ip

Detekcja systemy operacyjnego:

Nmap pozwala również na zdalne wykrycie systemu operacyjnego. Służy do tego opcja "-O".

Wykrywanie prób skanowania portów dzięki iptables.

iptables -A INPUT -m conntrack --cstate NEW -p tcp --tcp-flags SYN,RST,ACK,FIN,URG,PSH **SYN** -j LOG --log-level info --log-prefix "**uwaga! ktoś skanuje tcp syn**!"

iptables -A INPUT -m conntrack --cstate NEW -p tcp --tcp-flags SYN,RST,ACK,FIN,URG,PSH **FIN** -j LOG --log-level info --log-prefix "uwaga! ktoś skanuje tcp fin!"

ptables -A INPUT -m conntrack --cstate INVALID -p tcp --tcp-flags ! SYN,RST,ACK,FIN,URG,PSH SYN,RST,ACK,FIN,URG,PSH -j LOG --log-level info --log-prefix "**uwaga! skany tcp NULL**!"

Przykładowy firewall dla stacji roboczej

```
#! /bin/bash
#czyścimy wszystkie tablice iptables
iptables -t filter -F
iptables -t nat -F
iptables -t mangle -F
#ustawiamy politykę
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
#umożliwiamy ruch po lo
iptables -A INPUT -i lo -j ACCEPT
#akceptujemy polaczenia przychodzące, wcześniej nawiązane
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#przyjmujemy ping
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
#resetujemy wczesniejsze polaczenia tcp i udp
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
iptables -A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable
```

Przykładowy firewall dla serwera WWW

#! /bin/bash

#czyścimy wszystkie tablice iptables

iptables -t filter -F

iptables -t nat -F

iptables -t mangle -F

#ustawiamy politykę

iptables -P INPUT DROP

iptables -P OUTPUT ACCEPT

iptables -P FORWARD DROP

#umożliwiamy ruch po lo

iptables -A INPUT -i lo -j ACCEPT

#akceptujemy polaczenia przychodzące, wcześniej nawiązane

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

#przyjmujemy ping

iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

#otwieramy usługę http – dodatkowo sprawdzamy aby były to poł. nowe z flagą SYN

iptables –A INPUT –p tcp --dport 80 --syn -m state --state NEW –j ACCEPT

#otwieramy usługę ssh – dodatkowo sprawdzamy aby były to poł. nowe z flagą SYN

iptables –A INPUT –p tcp --dport 22 --syn -m state --state NEW –j ACCEPT

#resetujemy wcześniejsze polaczenia tcp i udp

iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset

iptables -A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable

Reguły resetujące wcześniej nawiązane połączenia znajdują się na końcu skryptu, aby nie "odciąć" poprawnie nawiązanych połączeń na dozwolonych portach. Jeżeli znalazły by się one na początku to np. odłączymy sesję ssh (zdalnej administracji) na której np. aktualnie pracujemy.

Wersja 2:

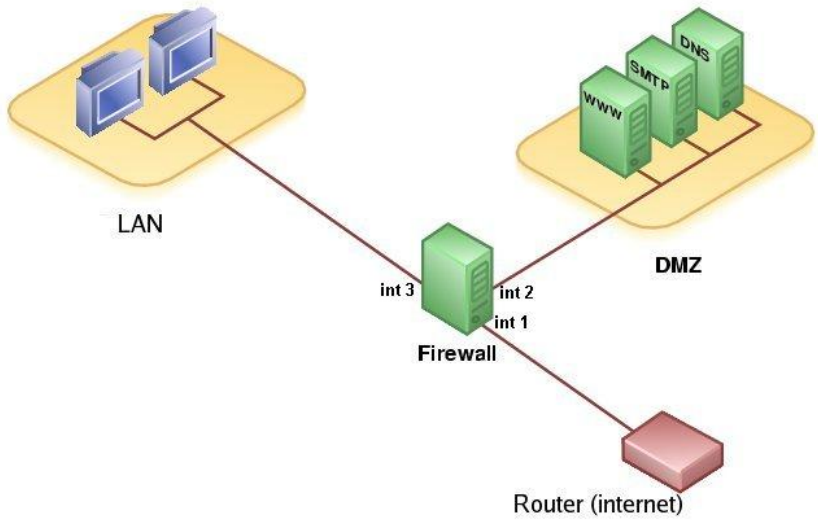
```
#!/bin/sh
iptables -F
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

```
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT

iptables -A OUTPUT -p tcp --sport 20:21 -j ACCEPT
iptables -A INPUT -p tcp --dport 20:21 -j ACCEPT

iptables -A OUTPUT -p tcp --sport 25 -j ACCEPT
iptables -A INPUT -p tcp --dport 25 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 110 -j ACCEPT
iptables -A INPUT -p tcp --dport 110 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset

iptables -A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable
```

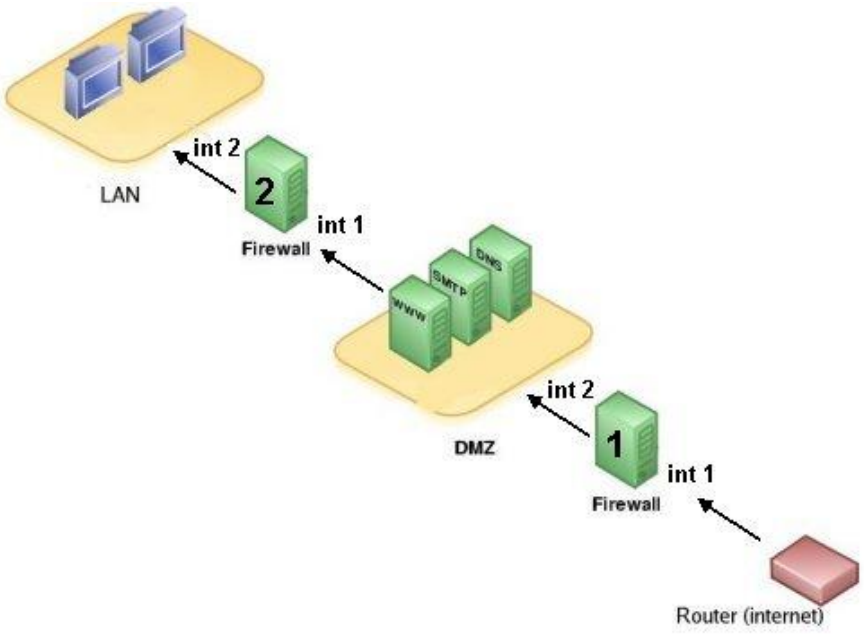
DMZ.

Poruszając temat konfiguracji firewalla należy zwrócić uwagę na następujące zagadnienie. Załóżmy, że nasza firma posiada własny serwer WWW, SMTP, czy DNS. Nasuwa się pytanie jak bezpiecznie udostępnić światu dostęp do zasobów tych serwerów, np. do firmowych stron WWW umieszczonych na takim serwerze (maszyna taka często nazywana jest jako „bastion host" – komputer baszta). Niektórzy fizycznie implementują to tak, że serwer WWW znajduje się w sieci LAN. Na firewall-u brzegowym tej sieci zdefiniowane jest przekierowanie pakietów przychodzących na port 80 do maszyny pełniącej rolę serwera WWW. Jest to częsty błąd popełniany przez administratorów systemów. Dlaczego? Ponieważ intruz przejmując kontrolę nad taką maszyną (znajdującą się wewnątrz chronionej sieci LAN) ma praktycznie niczym nie skrępowany dostęp do pozostałych maszyn, znajdujących się w tej sieci. Co więc należy zrobić w takim przypadku? W takiej sytuacji tworzymy tzw. strefę zdemilitaryzowaną Demilitarized Zone, w skrócie DMZ. Określenie to wywodzi się z terminologii wojskowej i oznacza strefę autonomiczną (bufor) pomiędzy wrogimi siłami militarnymi. Wracając do terminologii informatycznej strefę taką rozumiemy jako podsieć, która wydziela i separuje globalnie dostępne serwery od sieci wewnętrznej firmy. W przypadku złamania zabezpieczeń serwera WWW intruz nie będzie bezpośrednio zagrażał sieci LAN, ponieważ na jego drodze pojawiają się kolejne zabezpieczenia.

Przykładowa architektura 1:



Przykładowa architektura 2:



Przykładowa architektura 3: