Firewall i serwer proxy

1. Zrozumieć firewall

Firewall jest strukturą przeznaczoną do zatrzymywania ognia przed rozprzestrzenianiem. Budowany z cegieł firewall całkowicie oddziela pomieszczenia budynku. W samochodzie, jest to metalowa ściana oddzielająca silnik od przedziału pasażera. Internetowe firewall'e są przeznaczone do powstrzymania płomieni Internetowego piekła przed dostaniem się do twojej prywatnej sieci LAN. Lub, zabezpieczenie członków sieci LAN, czystych moralnie, przed uzyskaniem dostępu do wszystkich piekielnych części Internetu.;-)

Pierwszy firewall komputerowym, był to nie-trasowany host Unixa z połączeniem do dwóch różnych sieci. Jedna karta sieciowa była połączona do Internetu, a druga do prywatnej sieci LAN. Aby dostać się do Internetu z sieci prywatnej, musiałeś logować się na serwerze firewall'a (Unix).Potem korzystałeś z zasobów systemu przy dostępie do Internetu. Na przykład mogłeś użyć X-Window do uruchomienia przeglądarki Netscape w systemie firewall i wyświetlać na swoim komputerze. Z przeglądarką uruchomioną na firewall'u ma ona dostęp do obu sieci. Ten rodzaj dualnego systemu domowego (system z dwoma połączeniami sieciowymi) jest w porządku jeśli masz ZAUFANIE DO WSZYSTKICH twoich użytkowników. Możesz po prostu ustawić system Linux i dać konta w nim każdemu, kto chce mieć dostęp do Internetu. Przy takim ustawieniu, jedynym komputerem w twojej sieci, który wie, co się dzieje na zewnątrz jest firewall. Nikt nie może nic pobierać do swoich komputerów. Muszą najpierw pobrać plik do firewall'a a potem pobrać plik z firewall'a do swoich komputerów.

WIELKA NOTKA: 99% wszystkich włamań zaczyna się od uzyskania dostępu z poziomu konta w atakowanym systemie. Z tego powodu nie polecam tego typu firewall'a. Jest on bardzo ograniczony.

1.1 Założenia firewall'a

Nie powinieneś wierzyć, że sam firewall to wszystko, czego ci potrzeba. *Najpierw ustaw założenia.*

Firewall'e są używane dla dwóch celów.

1. trzymania z dala od komputera ludzi (robaki /crakcerów)
2. zatrzymanie ludzi (pracowników / dzieci) przed wyjściem poza firewall

Nie ma się co dziwić. Ludzie powinni pracować a nie bawić się w pracy. A czułem, że etyka pracy koroduje. Jednak, zaobserwowałem, że menagement sam nadużywał zasad, jakie sam stworzył. Moim rozwiązaniem tego typu nadużyć było publiczne logowanie się w firewall'u dla osób chcących oglądać strony WWW. Można się było obawiać o bezpieczeństwo interesu. Jeśli zarządzasz firewall'em zwróć na to uwagę.

Jak stworzyć zasady bezpieczeństwa

Stworzenie zasad bezpieczeństwa jest proste

1. opisz, jakich usług potrzebujesz
2. opisz, jaka grupa ludzi potrzebuje tych usług
3. opisz, do jakich usług, jaka grupa ma mieć dostęp
4. dla każdej usługi grupy opisz jak usługa powinna być zabezpieczona
5. napisz instrukcję, jakie formy dostępu są zabronione

Twoje zasady staną się bardziej złożone z czasem, więc nie próbuj wszystkiego od razu. Stwórz najpierw proste i jasne.

1.2 Typy firewall'i

Są da typy firewall'i

1. Firewall'e Filtrujące – blokujące wybrane pakiety sieciowe
2. Serwery Proxy (czasami nazywane firewall'ami) – tworzące połączenia sieciowe dla ciebie

Firewall'e filtrujące pakiety

Filtrowanie pakietów, jest to typ firewall'a wbudowany w kernel Linuxa. Firewall filtrujący działa na poziomie sieci. Dana jest przekazywana do systemu tylko, jeśli zasady firewall'a na to pozwalają. Przychodzące pakiety są filtrowane poprzez informacje o ich typie, adresie źródła, adresie docelowy i porcie, zwarte w każdym pakiecie. Wiele routerów sieciowych ma możliwości wykonywania pewnych usług firewall'a. Firewall'e filtrujące mogą być rozpatrywane jako typ routera. Z tego powodu musiałbyś dokładnie zrozumieć strukturę pakietu IP, aby z nim pracować. Ponieważ jest analizowanych i zapisywanych trochę danych, firewall'e filtrujące zajmują mniej CPU i tworzą mniejsze opóźnienia w sieci. Firewall'e filtrujące nie sprawdzają haseł. Użytkownik nie może się zidentyfikować. Jedyna identyfikacja użytkownika to identyfikacja po numerze IP przypisanym do określonej stacji roboczej. Może być to problem, jeśli korzystasz z DHCP (Dynamic IP assignements) Jest tak, ponieważ zasady są oparte i numery IP, do których musisz dostosować zasady, przy przypisywaniu n owego numeru IP. Nie wiem jak zautomatyzować ten proces.
Firewall'e filtrujące są bardziej przejrzyste dla użytkowników. Nie muszą oni ustawiać zasad w swoich aplikacjach korzystających z Internetu. Przy większości serwerów proxy nie jest to prawda

Serwery Proxy

Proxy są najczęściej używane do kontroli, monitorowania ruchu do sieci. Niektóre aplikacje proxy buforują wymagane dane. Obniża to przepustowość i zmniejsza dostęp do tych samych dla kolejnego użytkownika. Daje to również niezaprzeczalny dowód na to, co zostało przetransferowane.
Są dwa typy serwerów proxy

1. Aplikacje Proxy – pracujące dla ciebie
2. SOCKS Proxy - które krzyżują porty

Aplikacje Proxy

Najlepszym przykładem jest osoba telnetująca do innego komputera a potem telentująca z niego do świata zewnętrznego. Z aplikacją serwera proxy, ten proces jest zautomatyzowany. Ponieważ telnet łączy się ze światem, najpierw klient wysyła cię do proxy. Potem proxy łączy się do serwera, jakiego żądasz (ma zewnątrz) i zwraca ci dane. Ponieważ serwery proxy obsługują całą komunikację, mogą zapisywać w dzienniku wszystko, co się dzieje. Dla usługi HTTP (WWW) obejmuje to URL'e z jakich korzystasz. Dla FTP obejmuje to każdy plik jaki ściągasz. Mogą one nawet filtrować „niewłaściwe" słowa na stronach jakie odwiedzasz lub skanować wirusy. Aplikacje serwerów proxy mogą uwierzytelniać użytkowników. Przed połączeniem na zewnątrz, serwer może poprosić użytkownika, aby się najpierw zalogował. Dla WWW użytkownik odwiedzający każdą stronę musiałby się logować

SOCKS Proxy

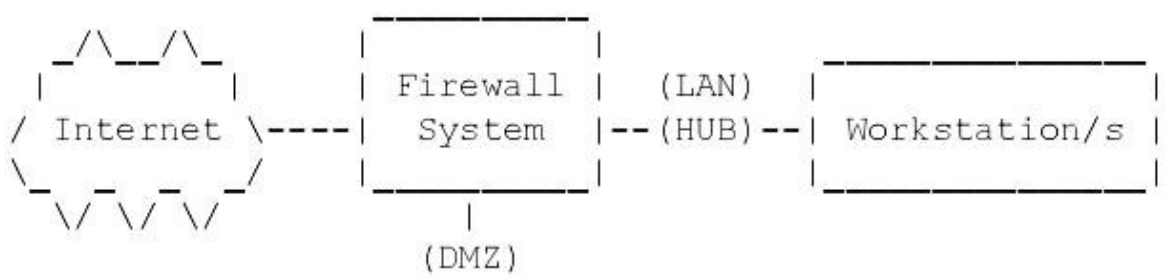
SOCKS serwer jest trochę podobny do tablicy z przełącznikami. Po prostu krzyżuje twoje połączenie z systemu do innego połączenia na zewnątrz. Większość serwerów SOCKS działa tylko z połączeniem typu TCP.I podobnie jak firewall'e filtrujące nie zapewniają uwierzytelniania użytkowników. Mogą być jednak rejestrem gdzie każdy użytkownik może się połączyć.

2. Architektura firewall'a

Jest wiele sposobów na strukturę sieci, chroniącą twój system przy użyciu firewall'a
Jeśli masz połączenie dedykowane do Internetu poprzez router, możesz podpiąć router bezpośrednio do systemu firewall'a. Lub możesz pójść przez hub dostarczający pełnego dostępu do serwerów poza firewall'em.

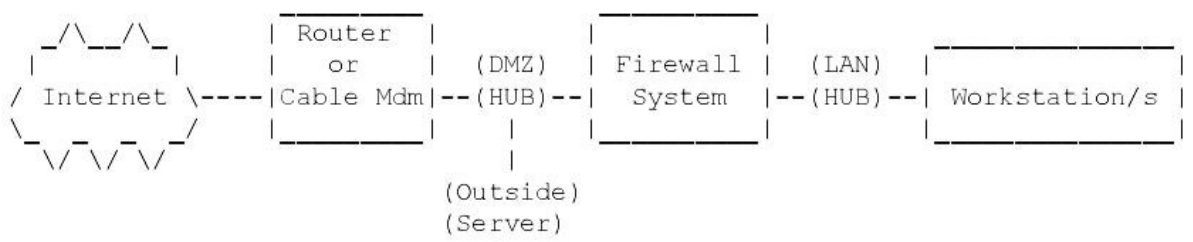
2.1 Architektura Dial-up

Możesz używać usługi dialup na linii ISDN. W tym przypadku możesz użyć trzeciej karty sieciowej pozwalającej na filtrowanie DMZ. Daje ci to pełną kontrolę na usługami internetowymi i jeszcze oddziela je od twojej podstawowej sieci



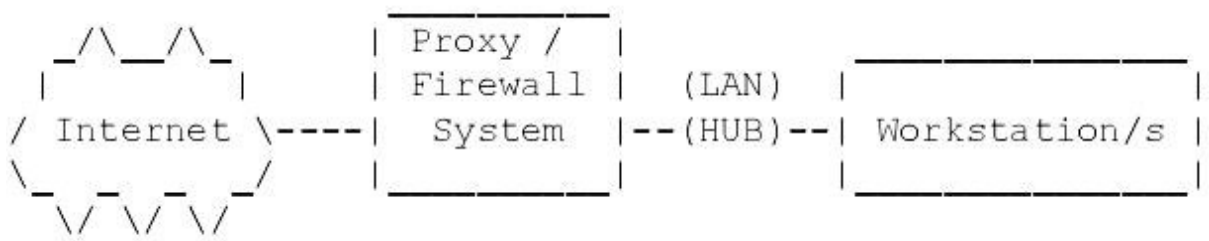
2.2 Architektura pojedynczego routera

Jeśli jest router lub modem pomiędzy tobą a Internetem. Jeśli posiadasz router, możesz ustawić pewne trudne zasady filtrowania w routerze. Jeśli jest to router twojego ISP, możesz nie mieć nad nim żadnej kontroli

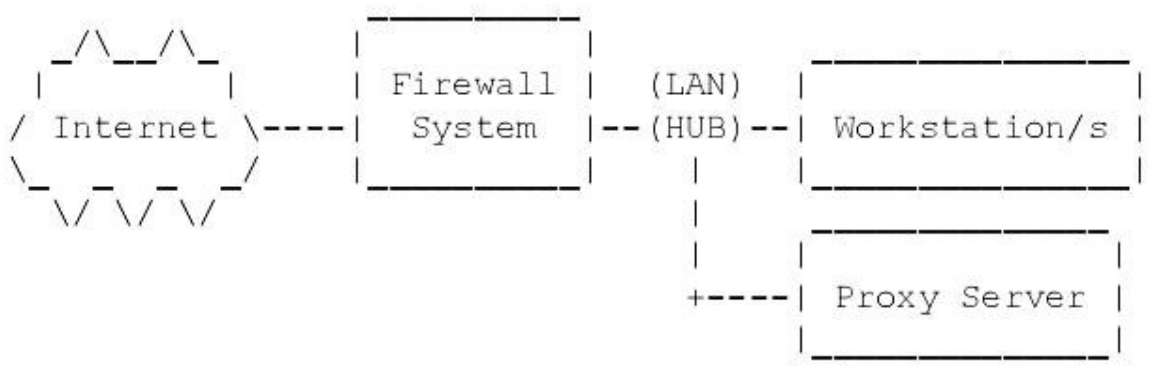


2.3 Firewall z serwerem proxy

Jeśli musisz monitorować gdzie chadzają użytkownicy twojej sieci, a twoja sieć jest mała, możesz połączyć serwer proxy ze swoim firewall'em. ISP czasami robią to tworząc ciekawą listę swoich użytkowników i sprzedają ją agencjom marketingowym.

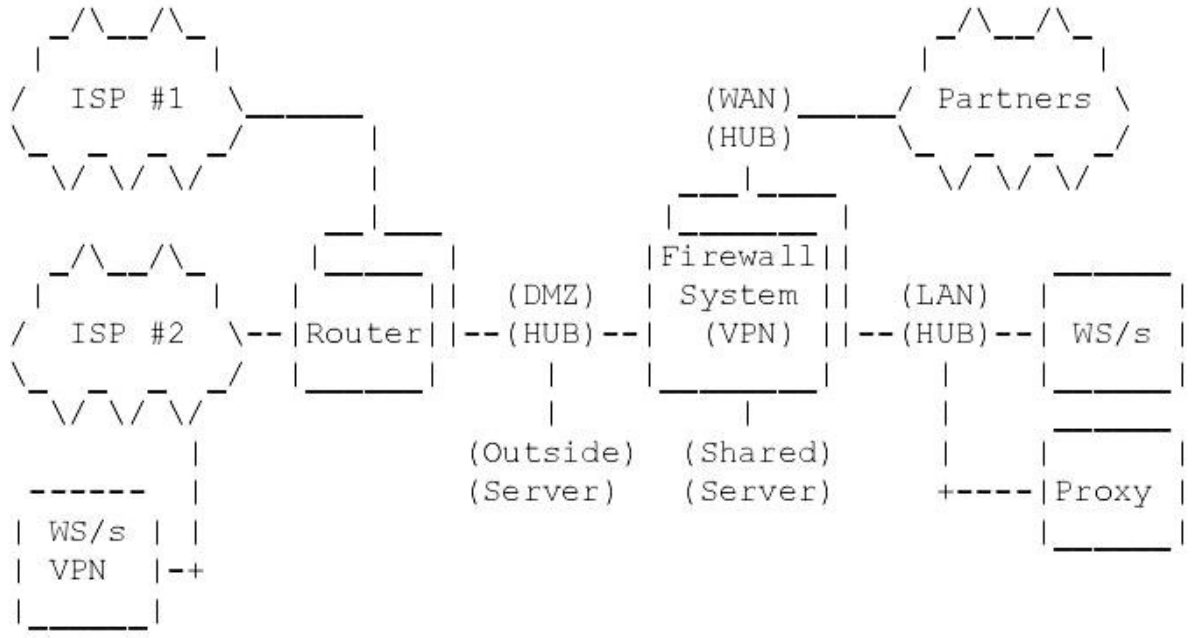


Możesz wstawić serwer proxy do swojej sieci LAN. W tym przypadku powinieneś mieć zasady pozwalające tylko serwerowi proxy łączyć się z Internetem dla usług jakich dostarcza. W ten sposób użytkownicy mogą uzyskać dostęp do Internetu tylko poprzez proxy



2.4 Nadmiarowa konfiguracja Internetu

Jeśli masz zamiar uruchomić usługę taką jak YAHOO lub być może SlashDot, możesz chcieć uczynić tak, aby twój system używał nadmiarowych routerów lub firewall'i. Przez zastosowanie metod cyklicznych DNS do dostarczenia dostępu do wielu serwerów sieciowych z jednego URL'a i wielu ISP, routery i firewall'e używające technik High Avaibility mogą stworzyć 100% czas działania usługi.



Łatwo jest pozwolić, aby twoja sieć wymknęła ci się spod kontroli. Kontroluj każde połączenie.

3. Ustawienia Firewall'a Filtrującego w Linuksie

3.1 Wymagania sprzętowe

Firewall'e filtrujące nie wymagają specjalnego sprzętu. To trochę więcej niż proste routery. Wszystko, czego ci trzeba to:

1. 486 – DX66 z 32 MB pamięci
2. 250MB dysku twardego (zalecane 500MB)
3. połączenie sieciowe (karty LAN, porty szeregowe, bezprzewodowe?)
4. monitor i klawiatura

W niektórych systemach przez użycie portów szeregowych konsoli, możesz wyeliminować monitor i klawiaturę. Jeśli potrzebujesz serwera, proxy, który obsługuje wielki ruch, powinieneś mieć większy system niż możesz dostarczyć. Jest tak, ponieważ dla każdego użytkownika, który się łączy do systemu będzie tworzony nowy proces. Jeśli będziesz miał 50 lub więcej użytkowników potrzebować będziesz:

1. Pentium II z 64 MB pamięci
2. 2 GB dysk twardy dla przechowywania wszystkich logów
3. dwa połączenia sieciowe
4. monitor i klawiaturę

Połączenia sieciowe mogą być dowolnego typu (karta NIC, ISDN, nawet modemy)

4. Wymagania programowe

4.1 Wybór kernela

Tworząc firewall filtrujący, nie potrzebujesz specjalnego oprogramowania. Linuks wystarczy. Ja używałem RedHat 6.1. Firewall Linuxa zmieniał się kilka razy. Jeśli używasz starszego kernela Linuksa (1.0 lub starszy) pobierz nowszą wersję Starsze używają ipfwadm z http://www.xos.nl/linux/ipfwadm i nie są dłużej wspierane w rozwoju. Jeśli używasz 2.2.13 lub nowszego będziesz używał ipchaining dostępnego na http://www.rustcorp.com/lonux/ipchains. Jeśli używasz nowszego kernela 2.4 jest nam nowsze narzędzie firewall'a.

4.1 Wybór serwera proxy

Jeśli chcesz ustawić serwer proxy potrzebujesz jednej z tych rzeczy

1. Squid
2. TIS Firewall Toolkit (FWTK)
3. SOCKS

Squid jest to wielki pakiet działający z funkcją Transparent Proxy Linuksa. Opiszę jak ustawić ten serwer. FWTK jest dostępny na http://www.tis.com/research/software. TIS stanowi zbiór programów zaprojektowanych pod kątem stosowania z zaporami ogniowymi. Z tymi narzędziami ustawisz demona dla każdej usługi (WWW, telnet, itp.).

5. Przygotowanie systemu Linux

Zainstaluje Linux. Swoją instalację zaczynam od konfiguracji serwera a potem wyłączam wszystkie niepotrzebne usługi w /etc/inetd.conf. Dla większego bezpieczeństwa powinieneś odinstalować niepotrzebne usługi. Musisz skompilować swój własny kernel. Najlepiej by było gdybyś zrobił to na komputerze innym niż firewall .Jeśli instalujesz kompilator i narzędzia w swoim własnym firewall'u, usuń je po całkowitej kompilacji kernela.

5.1 Kompilacja kernela

Zacznij od minimalnej instalacji dystrybucji Linuksa. Mniej oprogramowania jakie masz załadowane to mniej dziur, „tylnych drzwi" i / lub bugów, które mogłyby stworzyć problemy na twoim serwerze. Wybierz stabilny kernel. Ja używam kernela 2.2.13 w moim systemie. Więc dokumentacja jest oparta na tym ustawieniu. Musisz zrekompilować kernel z właściwymi ustawieniami. . Poniżej mam ustawienia sieciowe na jakich pracuję

```
<*> Packet socket
[ ] Kernel/User netlink socket
[*] Network firewalls
[ ] Socket Filtering
<*> Unix domain sockets
[*] TCP/IP networking
[ ] IP: multicasting
[*] IP: advanced router
[ ] IP: kernel level autoconfiguration
[*] IP: firewalling
[?] IP: always defragment (required for masquerading)
[?] IP: transparent proxy support
[?] IP: masquerading
--- Protocol-specific masquerading support will be built as modules.
[?] IP: ICMP masquerading
--- Protocol-specific masquerading support will be built as modules.
[ ] IP: masquerading special modules support
[*] IP: optimize as router not host
< > IP: tunneling
< > IP: GRE tunnels over IP
[?] IP: aliasing support
[*] IP: TCP syncookie support (not enabled per default)
--- (it is safe to leave these untouched)
< > IP: Reverse ARP
[*] IP: Allow large windows (not recommended if <16Mb of memory)
< > The IPv6 protocol (EXPERIMENTAL)
---
< > The IPX protocol
< > Appletalk DDP
< > CCITT X.25 Packet Layer (EXPERIMENTAL)
< > LAPB Data Link Driver (EXPERIMENTAL)
[ ] Bridging (EXPERIMENTAL)
[ ] 802.2 LLC (EXPERIMENTAL)
< > Acorn Econet/AUN protocols (EXPERIMENTAL)
< > WAN router
[ ] Fast switching (read help!)
[ ] Forwarding between high speed interfaces
[ ] PU is too slow to handle full bandwidth
QoS and/or fair queueing  --->
```

Po wykonaniu wszystkich ustawień jakich potrzebujesz do rekompilacji, przeinstaluj kernel i uruchom ponownie. Używam polecenia:
Make dep;make cleanmake bzililo;make modules;make modules_install;init 6 wszystko w jednym kroku

5.2 Konfiguracja dwóch kart sieciowych

Jeśli masz dwie karty sieciowe w komputerze, możesz dodać instrukcję append do pliku /etc/lilo.conf opisujący IRQ i adresy obu kart. Moje polecenie append wygląda tak:

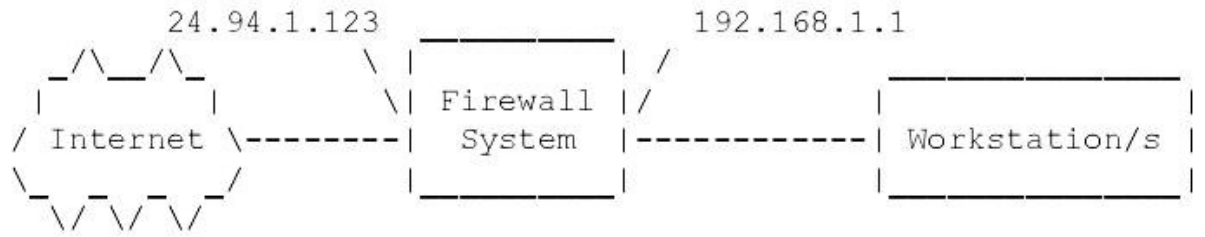
append=”ether=12,0x300,eth0 ether=15,0x340,eth1”

5.3 Konfiguracja adresów sieciowych

Teraz przejdziemy do ciekawej części naszego uruchamiania. Nie będę wchodził głęboko w to jak ustawić LAN. Twoim celem jest dostarczyć dwóch połączeń sieciowych do systemu firewall'a filtrującego. Jedno do Internetu (strona niezabezpieczona) i żadnego do LAN (strona zabezpieczona). Musisz podjąć kilka decyzji

1. Będziesz używał rzeczywistych numerów IP lub stworzysz gotowe dla sieci LAN
2. Czy twój ISP przypisze ci numer czy będziesz musiał używać statycznych numerów IP

Ponieważ nie chcesz aby internet miał dostęp do twojej sieci prywatnej, nie musisz używać „rzeczywistych adresów". Możesz przetworzyć adresy dla swojego prywatnego LAN. Ale nie jest to zalecane. Jeśli dane wychodzą z twojej sieci, mogą skończyć w porcie w innym systemie. Jest kilka zakresów adresów internetowych zarezerwowanych dla sieci prywatnych. Ten 192.168.1.xxx jest zarezerwowany i będziemy go używać w naszych przykładach. Będziemy używać podszywania IP aby zobaczyć co się zdarzy. W tym procesie firewall prześle pakiety i przetłumaczy je na „RZECZYWISTY" adres IP wędrujący do Internetu. Używając tego nie –rutowalnego adresu IP możesz uczynić swoją sieć bardziej bezpieczną. Routery internetowe nie przekażą pakietów z takimi adresami.



Musisz mieć „realny" adres IP aby przypisać ją do twojej karty sieciowej. Adres ten może być permanentnie przypisywany tobie (statyczny adres IP) lub może być przypisywany do połączenia sieciowego przez proces PPP. Przypisz swoje wewnętrzne numery IP. Jak 192.168.1.1 do karty LAN. Będzie to adres IP bramki. Możesz przypisać wszystkim pozostałym maszynom w sieci chronionej (LAN) numery z zakresu 192.168.1.xxx (od 192.168.1.2 do 192.168.1.254). Używam RedHat. Konfiguruję sieć w czasie uruchamiania. Dodałem plik ifcfg-eth1 w katalogu skryptów /etc/sysconfig/nework. Możesz również znaleźć ifcfg-ppp0 lub ifcfg-tr0 w tym katalogu. Te pliki 'ifcfg-' są używane przez RedHat do konfiguracji i włączania sieciowych sterowników w czasie uruchamiania

Podam tu ifcfg-eth1 (druga karta ethernetowa) dla naszego przykładu

```
DEVICE=eth1
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
GATEWAY=24.94.1.123
ONBOOT=yes
```

Jeśli masz zamiar używać połączenia dialup, będziesz musiał przepatrzyć pliki ifcfg-ppp0 i chat-ppp0, które kontrolują połączeniem PPP. Plik ifcfg może wyglądać tak:

```
DEVICE="ppp0"
ONBOOT="yes"
USERCTL="no"
MODEMPORT="/dev/modem"
LINESPEED="115200"
PERSIST="yes"
DEFABORT="yes"
DEBUG="yes"
INITSTRING="ATZ"
DEFROUTE="yes"
HARDFLOWCTL="yes"
ESCAPECHARS="no"
PPPOPTIONS=""
PAPNAME="LoginID"
REMIP=""
NETMASK=""
IPADDR=""
MRU=""
MTU=""
DISCONNECTTIMEOUT=""
RETRYTIMEOUT="5"
BOOTPROTO="none"
```

5.4 Testowanie sieci

Zacznij od zastosowania poleceń ifconfig i route. Jeśli masz dwie karty sieciowe ifconfig powinien wyglądać tak:

```
#ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:1620 errors:0 dropped:0 overruns:0
          TX packets:1620 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelan:0

eth0      Link encap:10Mbps Ethernet  HWaddr 00:00:09:85:AC:55
          inet addr:24.94.1.123 Bcast:24.94.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1000 errors:0 dropped:0 overruns:0
          TX packets:1100 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelan:0
          Interrupt:12 Base address:0x310

eth1      Link encap:10Mbps Ethernet  HWaddr 00:00:09:80:1E:D7
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1110 errors:0 dropped:0 overruns:0
          TX packets:1111 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelan:0
          Interrupt:15 Base address:0x350
```

A tablica trasowania powinna wyglądać tak:

```
#route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
24.94.1.0       *               255.255.255.0   U     1500   0       15 eth0
192.168.1.0     *               255.255.255.0   U     1500   0        0 eth1
127.0.0.0       *               255.0.0.0       U     3584   0        2 lo
default         24.94.1.123     *               UG    1500   0       72 eth0
```

Notka: 24.94.1.0 jest stroną internetową tego firewall'a a 192.168.1.0 jest stroną prywatną (LAN).
Powinieneś zacząć upewniwszy się, że komputer w twojej sieci LAN może pingować wewnątrz adresu systemu firewall'a (192.168.1.1 w tym przykładzie). Następnie z firewall'a, spróbuj pingować w Internecie. Używam www.internic.net jako moje miejsce testowe. Jeśli nie działa, spróbuj serwera swojego ISP. Jeśli i ten nie zadziała jako część Internetu, wtedy to znak ,że połączenie jest złe. Powinieneś móc łączyć się gdziekolwiek w Internecie z firewall'a .Spróbuj poszukać domyślnych ustawień bramki. Jeśli używasz połączenia dialup, kliknij sprawdź ID użytkownika i hasło. Teraz spróbuj spingować adresy zewnętrzne za firewall'em (24.96.1.123) z komputera w LAN. Nie powinno zadziałać. Jeśli tak, masz włączone maskowanie, lub już ustawiony jakiś pakiet do filtrowania. Wyłącz i spróbuj ponownie. Dla kernela nowszego niż 2.1.102 możesz zastosować polecenie:

echo „0" > /proc/sys/net/ipv4/ip_forward
Jeśli używasz starszego kernela, musisz zrekompilować kernel z uprzednim wyłączeniem.
Spróbuj spingować adresy zewnętrzne za firewall'em (24.94.1.123) ponownie. Nie powinno działać,. Teraz włącz przekazywanie pakietów IP i / lub maskaradę. Powinieneś móc pingować gdziekolwiek w Internecie z dowolnego systemu w twojej sieci LAN.

echo „1" > /proc/sys/net/ipv4/ip_forward
Duża Nota: Jeśli używasz „realnego" adresu IP w sieci LAN (nie 192.168.1.*) i nie możesz pingować internetu ale możesz pingować stronę Internetu zza firewall'a, upewnij się czy ISP routuje pakiety na twój prywatny adres sieciowy. Testem na ten problem jest posiadania kogoś w Internecie (powiedzmy przyjaciela używającego lokalnego dostawcę) używającego śledzenia drogi do twojej sieci. Jeśli śledzenie zatrzyma się na routerze twojego providera, wtedy wiadomo, że nie przekazuje twojego ruchu.
Działa? Wspaniale. Trudniejsza część zrobiona. :-)

5.4 Bezpieczeństwo firewall'a

Firewall nie jest dobry, jeśli system w którym jest wbudowany pozostaje szeroko otwarty na ataki. „Zły chłopiec" może uzyskać dostęp do usługi niezabezpieczonej przez firewall i zmodyfikować ją według swoich potrzeb. Musisz wyłączyć wszystkie niepotrzebne usługi. Spójrz do pliku /etc/inetd.conf. Ten plik konfiguruje inetd, również znany jako „super serwer". Steruje gałęzią serwera demonów i startuje je jeśli są wymagane przez przychodzący pakiet na „dobrze znany" port. Powinieneś wyłączyć echo,discard,daytime,chargen, ftp, gopher,shell,login,exec,talk,ntalk,pop-2,pop-3,netstat,systat,tftp,bootp,finger,cfinger,time,swat i linuxconfig, jeśli je masz. Aby wyłączyć usługę, wstaw # przed pierwszym znakiem w linii usługi. Kiedy to zrobisz, wyślij SIG-HUP do procesu przez wpisanie **„kill –HUP<pid>"** ,gdzie <pid> jest to numer procesu inetd. To spowoduje ,że inetd odczyta ponownie plik konfiguracyjny (inetd.conf) i zrestartuje bez

zamykania systemu. Przetestuj to telnetując do portu 15 (netstat) na firewall'u. Jeśli uzyskasz jakieś dane wyjściowe, to znaczy ,że nie masz wyłączonej tej usługi

telnet localhost 19

Możesz również stworzyć plik /etc/nologin. Wstaw w nim kilka linijek tekstu. Kiedy ten plik istnieje, użytkownik nie będzie się mógł zalogować, Zobaczysz zawartość tego pliku i odmowę logowania. Tylko root może się zalogować. Możesz również wyedytować plik /etc/securetty. Jeśli użytkownikiem jest root, wtedy login musi wystąpić na tty wylistowanym w /etc/securetty. Niepowodzenie będzie odnotowane w funkcji syslog.
NIGDY NI KORZYSTAJ Z TELENETU w systemie i loguj się JAKO ROOT. Możesz nawet wyłączyć telnet.
Jeśli jesteś paranoikiem możesz skorzystać z lids (Linux Intrusion Detect System). Jest to łatka do wykrywania intruzów do kernela Linuksa; może chronić ważne pliki przed zmianami. Kiedy działa , tylko root może zmienić chronione pliki lub katalogi i ich podkatalogi. Możesz zrestartować system z bezpieczeństwem =1 LILO ustawione do modyfikacji chronionych plików

6. Ustawienia filtrowania IP (IPFWADM)

Jeśli używasz kernela 2.1.102 lub nowszego przeskocz do sekcji o IPCHAINS. W starszych kernelach przesyłanie pakietów IP jest włączone domyślnie w kernelu. Z tego powodu, twoja sieć powinna zacząć od odmowy dostępu do wszystkiego i wykasowania wszystkich zasad ipfw w miejscu skąd ostatni raz zostały uruchomione. Ten fragment skryptu powinieneś wstawić do swojego startowego skryptu sieciowego (/etc/rc.d/init.d/network)

```
#
# setup IP packet Accounting and Forwarding
#
#   Forwarding
#
# By default DENY all services
ipfwadm -F -p deny
# Flush all commands
ipfwadm -F -f
ipfwadm -I -f
ipfwadm -O -f
```

Teraz mamy końcowy firewall. Nic się nie przedostanie. Teraz tworzymy plik /etc/rc.d/rc.firewall. te skrypt pozwala na email, WWW i ruch DNS ;-)

```
#! /bin/sh
#
# rc.firewall
#
# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi
case "$1" in
  start)
  echo -n "Starting Firewall Services: "
  # Allow email to got to the server
  /sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.10 25
  # Allow email connections to outside email servers
  /sbin/ipfwadm -F -a accept -b -P tcp -S 192.1.2.10 25 -D 0.0.0.0/0 1024:65535
  # Allow Web connections to your Web Server
  /sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.11 80
  # Allow Web connections to outside Web Server
  /sbin/ipfwadm -F -a accept -b -P tcp -S 192.1.2.* 80 -D 0.0.0.0/0 1024:65535
  # Allow DNS traffic
  /sbin/ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 192.1.2.0/24
  ;;
  stop)
  echo -n "Stooping Firewall Services: "
  ipfwadm -F -p deny
  ;;
  status)
  echo -n "Now do you show firewall stats?"
  ;;
  restart|reload)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: firewall {start|stop|status|restart|reload}"
        exit 1
esac
```

Notka: W tym przykładzie mamy uruchomiony serwer email (smtp) pod 192.1.2.10 ,który musi móc odbierać i wysyłać na porcie 25. Web serwer uruchomiony jest pod 192.1.2.11. Zezwalamy również każdemu z LAN na uzyskanie stron WWW i serwerów DNS na zewnątrz. Nie jest to doskonałe zabezpieczenie. Ponieważ port 80 nie musi być używany jako port WWW, sprytny hacker może użyć tego portu do stworzenia prywatnej sieci wirtualnej (VPN) poprzez firewall. W ten sposób ominie ustawienia sieciowe proxy. Użytkownicy LAN będą musieli przejść przez proxy aby uzyskać dojście do zewnętrznych serwerów sieciowych. Możesz również być zainteresowanym zliczaniem ruchu poprzez firewall. Ten skrypt będzie zliczał każdy pakiet. Możesz dodać linię lub dwie do zliczania pakietów z pojedynczego systemu.

```
# Flush the current accounting rules
ipfwadm -A -f
# Accounting
/sbin/ipfwadm -A -f
/sbin/ipfwadm -A out -i -S 192.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A out -i -S 0.0.0.0/0 -D 192.1.2.0/24
/sbin/ipfwadm -A in -i -S 192.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A in -i -S 0.0.0.0/0 -D 192.1.2.0/24
```

Jeśli wszystko czego potrzebujesz to firewall filtrujący tu możesz się zatrzymać. Przetestuj go i baw się dobrze.

7. Ustawienia filtrowania IP (IPCHAINS)

ipchains Linuksa to nadpisany kod Linux Ipv4 i nadpisany ipfwdam, które były nadpisane z ipfw BSD. Wymaga to zarządzania filtrowaniem pakietów IP w kernelu Linuksa w wersji 2.1.102 i wyższej. Starszy kod nie działa z fragmentami, ma 32 bitowy licznik (przynajmniej Intel), nie pozwala na inne protokoły niż TCP,UDP lub ICMP, nie można poczynić dużych zmian, nie można określić odwrotnych zasad, ma niestabilność i może być trudny do zarządzania (czyniąc go skłonnym na błędy użytkowników). Nie mam zamiaru zagłębiać się w firewall IPChains ponieważ jest OLBRZYMI!. Działasz z łańcuchem przez nazwę. Zaczynasz z trzema wbudowanym łańcuchami ,wejściowym, wyjściowymi i przekazywanym, których nie można usunąć. Możesz stworzyć swoje własne łańcuchy. Zasady mogą być dodane potem i usunięte z tych zasad. Operacje są dokonywane na całych łańcuchach.

1. Stwórz nowy łańcuch (-N)
2. usuń pusty łańcuch (-X)
3. Zmień zasady dla wbudowanego łańcucha (-P)
4. Listuj zasady w łańcuchu (-L)
5. Wykasuj zasady z łańcucha (-F)
6. Wyzeruj pakiet i bajty licznika wszystkich zasad w łańcuchu (-Z)

Jest kilka sposobów manipulacji zasadami w łańcuchu:

1. Dołącz nową zasadę do łańcucha (-A)
2. Wstaw nową zasadę na pewnej pozycji w łańcuchu (-I)
3. Zastąp zasadę na jakiejś pozycji w łańcuchu (-R)
4. Usuń zasadę z jakiejś pozycji w łańcuchu (-D)
5. Usuń pierwszą zasadę , która pasuje w łańcuchu (-D)

Jest kilka działań dla maskowania, które są w ipchains

1. Wylistuj bieżące zamaskowane połączenia (-M – L)
2. Ustaw wartość ograniczenia czasowego maskowania (-M – S)

Jest kilka kwestii czasowych obejmujące zmiany zasad firewall'a. Najprostsze podejście do tego jest następujące:

```
# ipchains -I input 1 -j DENY
# ipchains -I output 1 -j DENY
# ipchains -I forward 1 -j DENY
```

...robiąc zmiany...

```
# ipchains -D input 1
# ipchains -D output 1
# ipchains -D forward 1
#
```

To dołączy wszystkie pakiety podczas trwania zmian.
Tu duplikujemy powyższe zasady firewall'a w IPChains

```
#!/bin/sh
#
# rc.firewall
#
## Flush everything, start from scratch
  /sbin/ipchains -F input
  /sbin/ipchains -F output
  /sbin/ipchains -F forward

## Redirect for HTTP Transparent Proxy
  #$IPCHAINS  -A input -p tcp -s 192.1.2.0/24 -d 0.0.0.0/0 80 -j REDIRECT 8080

## Create your own chain
   /sbin/ipchains -N my-chain
   # Allow email to got to the server
   /sbin/ipchains -A my-chain -s 0.0.0.0/0 smtp -d 192.1.2.10 1024:-j ACCEPT
   # Allow email connections to outside email servers
   /sbin/ipchains -A my-chain -s 192.1.2.10 -d 0.0.0.0/0 smtp -j ACCEPT
   # Allow Web connections to your Web Server
   /sbin/ipchains -A my-chain -s 0.0.0.0/0 www -d 192.1.2.11 1024: -j ACCEPT
   # Allow Web connections to outside Web Server
   /sbin/ipchains -A my-chain -s 192.1.2.0/24 1024: -d 0.0.0.0/0 www -j ACCEPT
   # Allow DNS traffic
   /sbin/ipchains -A my-chain -p UDP -s 0.0.0.0/0 dns -d 192.1.2.0/24 -j ACCEPT

 ## If you are using masquerading
   # don't masq internal-internal traffic
   /sbin/ipchains -A forward -s 192.1.2.0/24 -d 192.1.2.0/24 -j ACCEPT
   # don't masq external interface direct
   /sbin/ipchains -A forward -s 24.94.1.0/24 -d 0.0.0.0/0 -j ACCEPT
   # masquerade all internal IP's going outside
   /sbin/ipchains -A forward -s 192.1.2.0/24 -d 0.0.0.0/0 -j MASQ

 ## Deny everything else
   /sbin/ipchains -P my-chain input DENY
```

Nie zatrzymuj się tu. To nie jest wspaniały firewall. Jestem pewny ,że będziesz chciał dołączyć jakieś swoje usługi.

8. Instalowanie przejrzystego proxy SQUID

Proxy squid jest dostępny na http://squid.nlanr.net . SQUID zawiera się w pakietach RedHata i Debiana. Jeśli możesz użyj jednego z nich.

9. Instalowanie serwera proxy TIS

9.1 Uzyskanie oprogramownia

TIS FWTK jest dostępny na http://www.tis.com/research/software. Kiedy ściągniesz pliki z TIS, PRZECZYTAJ README. TIS fwtk jest umieszczony w ukrytym katalogu na tym serwerze. TIS wymaga przeczytania umowy na http://www.tis.com/research/software/fwtk_readme.html a potem wysłania emaila do fwtk-request@tislab.com ze słowem **accepted** w treści wiadomości aby uzyskać nazwę tego ukrytego katalogu. Nie jest konieczny żaden temat w tej wiadomości. Ich system wyśle potem do ciebie maila z nazwą katalogu (coś około 12 godzin) do ściągnięcia źródła.

9.2 Kompilacja TIS FWTK

Wersja 2.1 FWTK kompiluje się dużo łatwiej niż starsze wersje. Uruchom **make**

9.3 Instalacja TIS FWTK

Uruchom make install. Domyślnym katalogiem instalacyjnym jest /usr/local/etc. Możesz zmienić na bardziej bezpieczny katalog. Zmieniłem dostęp do tego katalogu przez 'chmod 700'. Wszystko jest przygotowane do konfiguracji firewall'a

9.4 Konfiguracja TIS FWTK

Teraz dopiero zaczynamy Musimy powiedzieć systemowi aby wywołał nowe usługi I stworzył tablicę do ich kontrolowania. Nie próbuje tu nadpisać podręcznik TIS FWTK. Pokażę ci ustawienia działające i wyjaśnię problemy z uruchamianiem.

Są trzy pliki, które zajmują się tą kontrolą

- /etc/services – mówi systemowi jaka usługa jest na danym porcie
- /etc/inetd.conf – mówi programowi inetd aby go wywołać kiedy ktoś dostanie się do portu usługi
- /usr/local/etc/netperm-table – mówi usługom FWTK kto może a kto nie korzystać z usługi

Aby dobrać się do funkcji FWTK, powinieneś wyedytować te pliki. Edytowanie pliku services bez inetd.conf lub netperm-table ustawionych poprawnie może uczynić system niedostępnym.

Plik netperm-table

Plik ten kontroluje kto może uzyskać dostęp do usług TIS FWTK. Powinieneś pomyśleć o ruchu przez firewall w obie strony Ludzie poza twoją siecią powinni być zidentyfikowanie przed uzyskaniem dostępu, ale ludzie wewnątrz sieci mogą mieć zgodę tylko na dojście do firewall'a Do identyfikacji ludzi firewall używa programu authsrv utrzymujący bazę danych ID użytkowników i haseł . Sekcja uwierzytelniania netperm-table kontroluje gdzie baza danych jest trzymana i kto może uzyskać dostęp.

Miałem pewne problemy zamykając dostęp do tej usługi. Zauważ ,że użyłem „*" dając dostęp każdemu. Poprawne ustawienie dla tej linii to „authsrv: permit-hosts loclahost" jeśli miałoby to działać

```
#
# Proxy configuration table
#
# Authentication server and client rules
authsrv:      database /usr/local/etc/fw-authdb
authsrv:      permit-hosts *
authsrv:      badsleep 1200
authsrv:      nobogus true
# Client Applications using the Authentication server
*:            authserver 127.0.0.1 114
```

Aby zainicjalizować bazę danych, przejdź na roota i uruchom /authsrv w katalogu /var/local/etc aby stworzyć rekord administracyjny użytkownika. Tu mammy przykład sesji:

```
#
# authsrv
authsrv# list
authsrv# adduser admin "Auth DB admin"
ok - user added initially disabled
authsrv# ena admin
enabled
authsrv# proto admin pass
changed
authsrv# pass admin "plugh"
Password changed.
authsrv# superwiz admin
set wizard
authsrv# list
Report for users in database
user   group  longname            ok?    proto   last
------ ------ ------------------ -----  ------  -----
admin         Auth DB admin      ena    passw   never
authsrv# display admin
Report for user admin (Auth DB admin)
Authentication protocol: password
Flags: WIZARD
authsrv# ^D
EOT
#
```

W moim przykładzie, zezwalam hostowi ze środka prywatnej sieci przejść przez uwierzytelnienie. (permit-hosts 19961.2.* -passok), ale inny użytkownik musi podać swoje ID

i hasło aby używać proxy (permit-hosts * -auth) Zezwalam również jednemu z systemów (192.1.2.202) na dostęp do firewall'a bezpośrednio bez przechodzenia przez cały firewall. Robią to dwie linie inetacl –in.telnetd. Później wyjaśnię jak te linie są wywoływane.
Limit czasowy Telnet powinien być krótki

```
# telnet gateway rules:
tn-gw:                denial-msg      /usr/local/etc/tn-deny.txt
tn-gw:                welcome-msg     /usr/local/etc/tn-welcome.txt
tn-gw:                help-msg        /usr/local/etc/tn-help.txt
tn-gw:                timeout 90
tn-gw:                permit-hosts 192.1.2.* -passok -xok
tn-gw:                permit-hosts * -auth
# Only the Administrator can telnet directly to the Firewall via Port 24
netacl-in.telnetd: permit-hosts 192.1.2.202 -exec /usr/sbin/in.telnetd
```

Polecenie r działa w ten sam sposób co telnet

```
# rlogin gateway rules:
rlogin-gw:    denial-msg      /usr/local/etc/rlogin-deny.txt
rlogin-gw:    welcome-msg     /usr/local/etc/rlogin-welcome.txt
rlogin-gw:    help-msg        /usr/local/etc/rlogin-help.txt
rlogin-gw:    timeout 90
rlogin-gw:    permit-hosts 192.1.2.* -passok -xok
rlogin-gw:    permit-hosts * -auth -xok
# Only the Administrator can telnet directly to the Firewall via Port
netacl-rlogind: permit-hosts 192.1.2.202 -exec /usr/libexec/rlogind -a
```

Nie powinieneś mieć nikogo z bezpośrednim dostępem do firewall'a , wliczając w to FTP, więc nie wstawiaj serwera FTP w firewall'u.
Ponownie, linia permit-hosts pozwala każdemu w sieci chronionej na swobodny dostęp do Internetu a wszyscy inni muszą sami się uwierzytelniać. Włączyłem logowanie każdego pliku, wysyłanie i odbieranie pod moją kontrolą (-log {retr stor}) Czas trwania ftp kontroluje jak długo pobierane będzie złe połączenie jak również jak długo połączenie pozostanie otwarte przy jego aktywności.

```
# ftp gateway rules:
ftp-gw:               denial-msg      /usr/local/etc/ftp-deny.txt
ftp-gw:               welcome-msg     /usr/local/etc/ftp-welcome.txt
ftp-gw:               help-msg        /usr/local/etc/ftp-help.txt
ftp-gw:               timeout 300
ftp-gw:               permit-hosts 192.1.2.* -log { retr stor }
ftp-gw:               permit-hosts * -authall -log { retr stor }
```

WWW, gopher i przeglądarka oparte o ftp są sterowane przez http-gw. Pierwsze dwie linie tworzą katalog przechowujący ftp i dokumenty sieciowe jeśli są przekazywane przez firewall. Uczyniłem te pliki własnością roota i wstawiłem w katalogu dostępnym tylko rootowi. Połączenie WWW powinno być krótkie. Kontroluje ono jak długo użytkownik będzie oczekiwał na złe połączenie

```
# www and gopher gateway rules:
http-gw:       userid          root
http-gw:       directory       /jail
http-gw:       timeout 90
http-gw:       default-httpd   www.afs.net
http-gw:       hosts           192.1.2.* -log { read write ftp }
http-gw:       deny-hosts      *
```

ssl-gw rzeczywiście jest przekazywany przez dowolną bramkę. Bądź przy tym ostrożny. W tym przykładzie pozwoliłem każdemu wewnątrz sieci chronionej do połączenia z dowolnym serwerem na zewnątrz sieci za wyjątkiem adresów 127.0.0.* i 192.1.1.* a potem tylko na portach od 443 do 563 .Porty 443 do 564 są znanymi portami SSL.

```
# ssl gateway rules:
ssl-gw:         timeout 300
ssl-gw:         hosts           192.1.2.* -dest { !127.0.0.* !192.1.1.* *:443:563 }
ssl-gw:         deny-hosts      *
```

Tu mamy przykład jak używać plug-gw pozwalając na połączenia z serwerem wiadomości. W tym przykładzie pozwalam każdemu wewnątrz sieci chronionej na połączenie tylko z jednym systemem i tylko na porcie wiadomości. Podwójne linie zezwalają serwerowi wiadomości przekazać dane do sieci chronionej. Ponieważ większość klientów oczekuje stałego połączenia podczas gdy użytkownik czyta wiadomości, czas trwania dla serwera wiadomości powinien być długi

```
# NetNews Pluged gateway
plug-gw:        timeout 3600
plug-gw: port nntp 192.1.2.* -plug-to 24.94.1.22 -port nntp
plug-gw: port nntp 24.94.1.22 -plug-to 192.1.2.* -port nntp
```

Bramka fingera jest prosta .Każdy wewnątrz sieci chronionej musi najpierw się zalogować a potem zezwalamy mu na zastosowanie programu finger w firewall'u

```
# Enable finger service
netacl-fingerd: permit-hosts 192.1.2.* -exec /usr/libexec/fingerd
netacl-fingerd: permit-hosts * -exec /bin/cat /usr/local/etc/finger.txt
```

Nie ustawiałem usług Mail i X-windows, więc nie załączam przykładów.

Plik /etc/services

Jest to miejsce gdzie wszystko się zaczyna. Kiedy klient łączy się z firewall'em łączy się na znanym porcie (mniejszym niż 1024). Na przykład telnet łączy się na porcie 23. Demon inetd słyszy to połączenie i wyszukuje nazwy usługi w pliku /etc/services. Potem wywołuje program powiązany z tą nazwą w pliku /etc/inetd.conf

Niektóre z usług są tworzone nie jak zwykle w pliku /etc/services. Możesz przypisać pewne z nich do dowolnego portu jaki chcesz. Na przykład, ja przypisałem port telnetu administratora (telnet –a) do portu 24. Możesz przypisać go do portu 2323 jeśli sobie życzysz. Administrator (TY) łączy się bezpośrednio do firewall'a przez telnet na porcie 24 nie 23, a jeśli ustawisz plik netperm-table, jak to zrobiłem, będziesz tylko mógł zrobić to z jednego systemu wewnątrz twojej sieci chronionej

```
telnet-a        24/tcp
ftp-gw          21/tcp              # this named changed
auth            113/tcp   ident     # User Verification
ssl-gw          443/tcp
```

10 Serwer Proxy SOCKS

10.1 Ustawienia serwera proxy

Serwer proxy SOCKS jest dostępny na http://www.socks.nec.com. Rozkompresuj i przesuń pliki do katalogu w systemie. Miałem parę problemów kiedy to robiłem. Upewnij się ,że makefiles są poprawne. Jedną z ważnych rzeczy do odnotowania, jest to,że serwer proxy musi być dodany do /etc/inetd.conf. Musisz dodać linię:

socks stream tcp nowait nobody /usr/local/etc/sockd sockd

aby przekazać ,że serwer uruchomi się kiedy jest żądany

10.2 Konfiguracja serwera proxy

Program SOCKS potrzebuje dwóch oddzielnych plików konfiguracyjnych. Jeden mówi o pozwoleniu na dostęp, a jeden przekazuje żądanie do właściwego serwera proxy. Plik dostępu powinien być umieszczony na serwerze. Plik trasowania powinien być umieszczony na każdej maszynie uniksowej. DOS i przykładowo ,komputery Macintosh będą wykonywać swoje własne trasowanie.

Plik dostępu

Przy socks4.2 Beta, plik dostępu jest nazywany „sockd.conf" Powinien zawierać 2 linie, linię zezwolenia i linię zabronienia. Każda linia ma trzy wejścia:

- Identyfikator (zezwolenie/zabronienie)
- Adres IP
- Adres modyfikatora

Identyfikator albo zezwala albo zabrania. Powinieneś mieć obie linie. Adres IP przechowuje cztery bajty adresu w typowej notacji kropki IP ,np. 192.168.1.0. Modyfikator adresu jest również typowym adresem IP, czterobajtową liczbą. Działa jako netmask. Przewidywana liczba jest 32 bitowa (1 lub 0).Jeśli bit to 1, odpowiedni bit adresu,który jest sprawdzany musi odpowiadać odpowiedniemu bitowi w polu adresu IP.Na przykład, jeśli linia to:

permit 192.168.1.23 255.255.255.255

zezwoli ona tylko na adresy IP, które odpowiadają każdemu bitowi w 192.168.1.23, np. tylko 192.168.1.3. Linia;

permit 192.168.1.0 255.255.255.0

zezwoli każdemu numerowi wewnątrz grupy 192.168.1.0 do 192.168.1.255 w całej domenie klasy C. Nie powinno być linii:

permit 192.168.1.0 0.0.0.0

ponieważ zezwoli ona każdemu adresowi.

Więc najpierw zezwól każdemu adresowi jakiemu chcesz zezwolić, a potem resztę zablokować. Aby zezwolić każdy w domenie 192.168.1.xxx, linie:

permit 192.168.1.0 255.255.255.0
deny 0.0.0.0   0.0.0.0

będzie pracował dobrze. Odnotuj najpierw „0.0.0.0" w lii deny. Z modyfikatorem 0.0.0.0 , pole adresu IP nie ma znaczenia. Wszystkie 0 są reguła ponieważ jest to łatwo wpisać. Dozwolone jest więcej niż jedno wejście. Określeni użytkownicy mogą również mieć zagwarantowany lub zablokowany dostęp Jest to robione poprzez uwierzytelnienie ident. Nie wszystkie systemy obsługują ident, wliczając w to Trumpet Winsock, wiec raczej go nie używaj.

Plik trasowania
Plik trasowania w SOCKS jest nazywany „socks.conf". Plik trasowania jest wykorzystywany kiedy korzystasz z gniazd jak i kiedy nie. Na przykład, w naszej sieci 192.168.1.3 nie musimy używać gniazd aby kontaktować się w192.168.1.1, firewall'em .Ma bezpośrednie połączenie poprzez Ethernet. Automatycznie definiuje 127.0.0.1, pętla zwrotna. Oczywiście nie musisz korzystać z SOCKS do rozmowy z samym sobą. Ma trzy wejścia:

- deny
- direct
- sockd

Deny mówi SOCKS kiedy odrzuca żądanie. To wejście ma takie same trzy pola jak sockd.conf, identyfikator, adres i modyfikator. Generalnie, ponieważ jest obsługiwany przez sockd.conf, w pliku dostępu, pole modyfikatora jest ustawione 0.0.0.0. Jeśli chcesz uniemożliwić wywołanie z dowolnego miejsca, możesz to zrobić teraz. Wejście direct mówi, które adresy nie używają gniazd. Są to wszystkie adresy, które mogą być osiągalne bez serwera proxy. Ponownie mamy trzy pola, identyfikator, adresy i modyfikator. Nasz przykład to może mieć

direct 192.168.1.0 255.255.255.0

Zatem idzie bezpośrednio do naszej chronionej sieci .Wejście sockd mówi komputerowi, który host ma demon serwera gniazd u siebie.. Składnia:

sockd @=<serverlist> <IP adress> <modifier>

Zauważ @ = wejście. Pozwala to ustawić adresy IP listy serwerów proxy. W naszym przykładzie używamy jednego serwera proxy. Adres IP i pole modyfikatora działają podobnie jak w innych przykładach. Określasz które adresy przechodzą przez 6.2.3 DNS z poza firewall'a. Ustawienie usługi Domeny Nazw z poza firewall'a jest stosunkowo prostym zadaniem. Musisz tylko ustawić jedynie DNS na maszynie firewall'a. Potem ustawia każdą maszynę poza firewall'em aby użyć tego DNS'a.

10.2 Praca z serwerem proxy
Unix
Mając aplikacje działające z serwerem proxy, muszą być „zgniadkowane" Będziesz miał dwa różne telnety ,jeden dla bezpośredniej komunikacji, jedne dla komunikacji poprzez serwer proxy. SOCKS przychodzi z instrukcjami jak SOCKować program, jak również parę pre-

SOCKowanych programów. Jeśli używasz wersji SOCKowanej do bezpośredniego dojścia. SOCKS automatycznie przełączy się do bezpośredniej wersji .Z tego powodu chcemy zmienić nazwy wszystkich programów w naszej sieci chronionej i zastąpić je programami SOCKowanymi. „Finger" staje się „finger.orig", „telnet" staje się „telnet.orgi" itd. .Musisz przekazać to SOCKS przez dołączenie pliku socks.h. Pewne programy będą obsługiwały trasowanie i sockowanie. Netscape jest jednym z nich. Możesz użyć serwera proxy z Netscape przez wprowadzenie adresu serwera (192.168.1.1 w naszym przykładzie) w polu SOCKS zgodnie z Proxy.

MS Windows z Trumpet Winsock

Trumpet Winsock ma wbudowane możliwości serwera proxy .W menu „setup" wpisz adres IP serwera, i adresy wszystkich komputerów osiągalnych bezpośrednio. Trumpet będzie obsługiwał wszystkie wychodzące pakiety.

Wykorzystanie serwera proxy do pracy z pakietami UDP

Pakiet SOCKS działa tylko z pakietami TCP, nie UDP. To czyni go trochę mniej użytecznym. Wiele programów ,takich jak talk i Archie, używają UDP. Jest to pakiet zaprojektowany do używania jako serwer proxy dla pakietów UDP nazywany UDPrelay.

Wady serwerów proxy

Serwer proxy, poza wszystkim ,jest urządzeniem bezpiecznym. Użycie go do zwiększania dostępu do internetu z ograniczonymi adresami IP będą miały wiele wad .Serwer proxy zezwoli na większy dostęp z wewnątrz sieci chronionej na zewnątrz ale będzie utrzymywał niedostępną sieć wewnętrzną przed dostępem z zewnątrz. Oznacz to ,brak serwerów, rozmów lub połączeń archiwalnych, lub bezpośredniego mailiowania do komputerów wewnętrznych. Te wady mogą wydawać się nieznaczne, ale pomyśl o tym w ten sposób:

- Pozostawiłeś raport na komputerze wewnątrz firewall'a sieci chronionej. Jesteś w domu, i postanowiłeś ,że musisz go pobrać. Nie możesz. Nie masz dojścia do komputera ponieważ jest poza firewall'em. Próbujesz się zalogować najpierw na firewall, ale ponieważ każdy ma dostęp do serwera proxy, nikt nie ma dostępu do twojego konta na nim.
- Twoja córka idzie do szkoły. Chcesz do niej zamailować. Masz do niej prywatną sprawę i chciałbyś wysłać maila ze swojego komputera. Musisz zaufać całkowicie swojemu administratorowi systemu, ale ciągle jest to list prywatny.
- Niemożność wykorzystania pakietów UDP to duża wada przy serwerach proxy.

FTP powoduje inny problem z serwerem proxy .Kiedy korzystasz z ls, serwer FTP otwiera gniazdo na maszynie klienta i wysyła informację przez nie .Serwer proxy nie pozwala na to, więc FTP nie pracuje właściwie. A serwer proxy uruchamia się wolno. Zasadniczo, jeśli masz adresy IP i nie martwisz się zbytnio o bezpieczeństwo, nie używaj firewall'a i / lub serwera proxy. Jeśli nie masz adresów IP, ale również nie martwisz się zbytnio o bezpieczeństwo, będziesz również używał emulatora IP, takiego jak term ,Slirp lub TIA. Term jest dostępny na ftp://susnsite.unc.edu, Slirp na ftp://blitzen.canberra.edu.au/pub/slirp a TIA jest dostępny na marketplace.com. te pakiety będą uruchamiały się szybciej , pozwalając na lepsze połączenia,i dostarczają wyższego poziomu dostępu do wewnętrznej sieci z internetu. Serwery proxy są dobre dla tych sieci, które mają wiele hostów, które chcą połączyć się z internetem w locie.

11 Zaawansowana konfiguracja

Jest jedna konfiguracja jaką chciałem naszkicować przed zakończeniem tego tekstu. Ta konfiguracja powinna być wystarczająca dla większości ludzi. Jednak kolejny szkic jaki pokaże bardziej zaawansowaną konfigurację, która wyjaśni pewne kwestie.

11.1 Duże sieci z naciskiem na bezpieczeństwo

Powiedzmy, że życzysz sobie mieć własną sieć. Masz 50 komputerów i podsieć z 32 (5 bitów) numerami IP. Potrzebujesz różnych poziomów dostępu wewnątrz sieci ponieważ przekazujesz członkom sieci różne rzeczy. Dlatego też musisz chronić pewne części sieci przed resztą.

Te poziomy to:

1. Poziom zewnętrzny. Jest to poziom dostępny dla każdego,. Tam możesz pozyskiwać nowych członków.
2. **Troop.** Jest to poziom ludzi, którzy pozostają poza poziomem zewnętrznym. Tu możesz nauczyć ich np., jak robić bomby
3. **Mercenary.** Jest to miejsce gdzie są trzymane *rzeczywiste* plany. Na tym poziomie są przechowywane wszystkie informacje o tym jak rządy 3 świata chcą poprawić swoją sytuację

Ustawienie sieci

Numery IP są zorganizowane następująco:

- 1 numer to 192.168.1.255, jest to adresem nadawczym i nie jest przydatny
- 23 z 32 adresów IP są umieszczone na 23 maszynach, które będą dostępne w internecie
- 1 dodatkowe IP dla Linuksa w twojej sieci
- 1 dodatkowe dla różnych Linuksów w tej sieci
- 2 IP # idą do routera
- 4 są nieużywane, ale podają nazwę domen paul, rongo, john i goerge
- Sieci chronione mają adresy 192.168.1.xxx

Potem dwie oddzielne sieci są budowane, każda w innym miejscu. Są one routowane poprzez podczerwony ethernet aby były całkowicie niewidoczne z zewnątrz. Szczęśliwie podczerwony ethernet działa jak normalny ethernet, Te sieci są połączone z Linuksem z dodatkowym adresem IP. Jest plik serwera łączący dwie sieci chronione. Plik serwera przechowuje adres 192.168.1.17 dla sieci Troop a 192.168.1.23 dla sieci Mercenary Ma da różne adresy IP ponieważ ma różne karty ethernetowe. Przekazywanie IP jest wyłączone .Przekazywanie IP na obu Linuksach jest również wyłaczone. Router nie przekazuje pakietów przezanczonych dla 192.168.1.xxx chyba ,że jasno powiesz mu to, więc interent nie będzie mógł go uzyskać Powodem wyąłczenia tu Przekazywania IP jest to ,aby pakiety z sieci Troop nie były dostępne dla sieci Mercenary i vice versa.

Serwer NFS może być również ustawiony , oferując różne pliki dla różnych sieci. Może się to przydać. Uzywając tego ustawienia i innych kart ethernetowych możesz oferować ten plik serwera wszystkim trzem sieciom

Ustawienia Proxy

Teraz keidy wszystkie trzy poziomy chca monitorować sieć dla swoich przebieglych celów, wszytskie trzy musza mieć do niej dostep. Sieć zewnętrzna jest połączona bezpośrednio do inernetu, więc nie będziemy tu mówić o serwerze proxy. Sieci Mercenery i Troop sa poza firewall'em, więc konieczne jest ustawienie serwera proxy. Obie sieci będą ustawiane bardzo

podobnie. Obie mają ten sam adres IP do nich przypisany Podam klika interesujących parametrów

1. Nikt nie może użyć pliku serwera dla dostępu do internetu. To wystawia plik serwera na wirusy i inne złe rzeczy, a jest on ważny, więc jest wyłączony.
2. Nie zezwalamy Troop na dostęp do WWW

Plik sockd.conf w Troop Linuksa będzie miał linię:

```
deny 192.168.1.17  255.255.255.255
```

a maszyna Mercenary:

```
deny 192.168.1.23  255.255.255.255
```

Troop Linuksa będzie miał linię

```
deny 0.0.0.0  0.0.0.0  eq 80
```

To zabrania dostępu wszystkim maszynom próbującym dostępu do portu 80 , portu http. Zezwala na wszystkie inne usługi, z wyjątkiem dostępu do WWW
Oba pliki mają:

```
permit 192.168.1.0  255.255.255.0
```

zezwalając wszystkim komputerom w sieci 192.168.1.xxx uzywać tego serwera proxy z wyjątkiem tych, które sa już zablokowane (tj, plik serwera i dostęp do WWW z seici Troop)

Plik sockd.conf Troop wyglądac będzie tak:

```
deny 192.168.1.17  255.255.255.255
deny 0.0.0.0  0.0.0.0  eq 80
permit 192.168.1.0  255.255.255.0
```

a plik Mercenary tak:

```
deny 192.168.1.23  255.255.255.255
permit 192.168.1.0  255.255.255.0
```

To powinno skonfigurowac wszystko poprawnie. Każda sieć jest konsekwentnie izolowana, z właściwą ilością interakcji. Każdy powinien być zadowolony

12 Uproszczenie zarządzania

12.1 Narzędzia firewall'a

Jest kilka pakietów oprogramowania, które mogą uczynić zarządzanie firewall'em łatwiejszym. Uzywaj tych narzędzi ostroznie , chyba ,że możesz obejść się bez nich. Zarówno graficzne jaki sieciowe intrefejsy zostały stworzone do pracy z zasadami filtorwania Linuksa. Niektóre firmy mają stworzone komercyjne firewall'e oparte na Linuksie, przez wstawienie swoich okein dialogowych w kodzie zarządzającym. Jestem zwolennikiem GUI. Jednak, uztwam firewall'a z GUI czasami. Znalzłem pomoc przez dostarczanie raportów z wszystkich zasad w jedne prosty sposób.

Gfcc (GTK + Firewall Control Center) jest to GTK + aplikacja , która może sterowac zasadami firewall'a Linuksa i zasadami opartymi o pakiet ipchains. Zawarłęm skrypt RC w dodatku A. Ten skrypt działą bez gfcc
Jest wiele skryptów dostępnych do ustawiania firewall'a .jden dobry skrypt jest dostępny na http://www.jasmine/org.uk/~simon/bookshelf/papers/instant-firewall/ instant-firewall.html. Inny skrypt na http://www.pointman.org/
Kfirewall jest z GUI dla ipchains lub ipfwadm (w zależności od wersji kernela) http://megaman.ypsilonia.net/kfirewall
FTC jest narzędziem opartym o HTML dla konfiguracji firewall'a Cechuje go automatyczne generowanie skryptu dla polecenia IP- filtorwanie (ipfwadm) na firewall'u dla wielu interfejsów i innych usług sieciowych. http://www.fen.baynet.de/~ft114/FCT/firewall.htm

12.2 Narzędzia ogólne
WebMin jest pakietem systemu ogólnego admina. Nie pomoże zarządzać zasadami firewall'a ale pomoże ci włączać i wyłaczać demony i procesy Ten program jest BARDZO dobry http://www.webmin.com
Jeśli jesteś u ISP, będziesz chciał wiedzieć o IPFA (IP Firewall Accounting) http://www.soaring-bird.com/ipfa .Może zapisywac do dziennika logów co miesiąc, co dzień , co minutę i ma GUIdo admiistrowania oparte o WWW