Brandväggar och Proxyservers HOWTO (Firewall HOW-TO)

Mark Grennan, markg@netplus.net. Svensk översättning av *Tomas Carlsson* <mailto:md5tc@mdstud.chalmers.se>
v0.4, 8 November 1996, Svensk version 15 Juni 1998

Detta dokument är designat för att lära ut grunderna i brandväggssystem och att ge dig lite detaljer om hur man konfigurerar både en filtrerande och proxy brandvägg på en Linuxbaserad PC. En (engelsk) HTML-version av detta dokument finns tillgänglig på *http://okcforum.org/~markg/Firewall-HOWTO.html*

Innehåll

1 Introduktion

Originalversionen av Firewall-HOWTO skrevs av David Rudder, **drig@execpc.com**. Jag skulle vilja tacka honom för att han tillät mig att uppdatera hans arbete.

Brandväggar har fått stor uppmärksamhet som 'det ultimata' i Internetsäkerhet den senaste tiden. Som med de flesta saker som får stor uppmärksamhet, så följer med detta missuppfattningar. Denna HOWTO kommer att ta upp grunderna om vad en brandvägg är för något, hur man sätter upp en, vad proxyservrar är, hur man sätter upp proxyservrar och applikationerna för denna teknik utanför säkerhetens sfär.

1.1 Feedback

All feedback är väldigt välkommen. **VAR VÄNLIG ATT RAPPORTERA ALLA ORIKTIGHETER I DETTA DOKUMENT!!!**. Jag är mänsklig och benägen att göra fel. Om du hittar några så är jag mycket intresserad att rätta till dem. Jag kommer att försöka svara på alla e-postmeddelanden, men jag är upptagen så bli inte förolämpad om jag inte gör det.

Min e-postadress är ***markg@netplus.net***

1.2 Disclaimer

JAG ÄR INTE ANSVARIG FÖR NÅGRA SKADOR SOM INTRÄFFAR PÅ GRUND AV AGERANDE SOM ÄR BASERAT PÅ DETTA DOKUMENT. Detta dokument är tänkt att vara en introduktion till hur brandväggar och proxyservrar fungerar. Jag är inte, och jag låtsas inte heller att vara, en säkerhetsexpert. Jag är bara en person som har läst för mycket och tycker om datorer mer än de flesta. Snälla, jag skriver detta för att hjälpa folk att bli bekanta med ämnet och jag är inte beredd att satsa mitt liv på att allt som finns i detta dokument är korrekt.

1.3 Copyright

1.3.1 Svensk version

1.3.2 Engelsk version, oförändrad från originaldokumentet

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have any questions, please contact Mark Grennan at <markg@netplus.net>.

1.4 Anledningar till att jag skriver detta

Även om det har varit många diskussioner i `comp.os.linux.*` om brandväggar det senaste året, så hade jag svårt för att hitta den information jag behövde för att sätta upp en brandvägg. Originalversionen av denna HOWTO hjälpte till men det saknades fortfarande lite grann. Jag hoppas att denna 'upp-hottade' version av David Rudder's Firewall HOWTO kommer att ge alla den information de behöver för att skapa en fungerande brandvägg på några timmar, inte veckor.

Jag känner också att jag borde ge tillbaka någonting till Linuxvärlden.

1.5 Att göra

- Ge lite instruktioner om hur man sätter upp klienterna.
- Hitta en bra UDP proxyserver som fungerar med Linux

1.6 Ytterligare Läsning

- NET-2 HOWTO (nuvarande NET-3-HOWTO (SvÖ))
- Ethernet HOWTO
- Multiple Ethernet Mini HOWTO
- Networking with Linux
- PPP HOWTO
- TCP/IP Network Administrator's Guide av O'Reilly and Associates
- Dokumentation för TIS Firewall Toolkit

Trusted Information Systems (TIS) www-sajt har en stor samling dokumentation om brandväggar och relaterat material. *TIS www-sajt* `<http://www.tis.com/>`

Dessutom så arbetar jag med ett säkerhetsprojekt som jag kallar för *Secure Linux*. På www-sajten för *Secure Linux* så samlar jag all information, dokumentation och program som man behöver för att skapa ett pålitligt Linuxsystem. E-posta mig om du vill ha information.

1.7 Översättarens kommentarer

Jag kommer använda översättningen 'Internet' i detta dokument, med vilket jag menar det allmänt kända Internet. Men informationen i detta dokument bör även gälla för vilket annat internätverk som helst, åtminstonde ett som använder samma protokoll som Internet.

Jag har ändrat alla förekomster av 'NET-2 HOWTO' till 'NET-3 HOWTO' eftersom det är den versionen av det dokumentet som är aktuell för tillfället.

De tillägg som jag har gjort i texten har jag markerat med '(SvÖ)'.

2 Att förstå brandväggar

En brandvägg är en term som används för en bildel. I bilar så är brandväggar fysiska objekt som separerar motorn från passagerarna. De är till för att skydda passageraren ifall bilens motor fattar eld medans den fortfarande låter föraren använda motorns kontroller.

En brandvägg i en dator är en enhet som skyddar ett privat nätverk från den publika delen (Internet som helhet).

Brandväggsdatorn, från och med nu kallad "brandväggen", kan nå både det skyddade nätverket och Internet. Det skyddade nätverket kan inte nå Internet och Internet kan inte nå det skyddade nätverket.

För att någon skall kunna nå Internet från insidan av det skyddade nätverket, så måste de göra en telnetanslutning till brandväggen och använda Internet därifrån.

Den enklaste formen av branvägg är ett 'dual homed' system (ett system med två nätverksanslutningar). Om du kan LITA på ALLA dina användare, så kan du helt enkelt sätta upp ett Linuxsystem (kompilera med IP forwarding/gatewaying avstängt) och ge alla användare konton till den. De kan sedan logga in på detta system och använda telnet, FTP, läsa e-post och andra tjänster som du erbjudit. Med denna inställning så är brandväggen den enda dator på ditt privata nätverk som vet någonting om världen utanför. De andra systemen på ditt skyddade nätverk behöver inte ens någon *default route*.

Detta måste upprepas: För att ovanstående brandvägg skall fungera så **MÅSTE DU LITA PÅ ALLA DINA ANVÄNDARE!**. Jag rekommenderar inte detta.

2.1 Nackdelar med brandväggar

Det finns två typer av brandväggar.

1. IP Filtrerande Brandväggar - som blockerar all nätverkstrafik utom viss utvald trafik.
2. Proxyservrar - som gör nätverksanslutningarna åt dig.

2.1.1 IP Filtrerande Brandväggar

En IP filtrerande brandvägg arbetar på paketnivån. Den är designad för att kontrollera flödet av paket baserat på källan, destinationen, porten och pakettypen som finns lagrade i varje paket.

Denna typ av brandvägg är väldigt säker men den saknar all form av användbara loggningsfunktioner. Den kan hindra personer från att komma åt privata system men den kommer inte att tala om vem som accessade dina publika system eller vem som accessade Internet från insidan.

Filtrerande brandväggar är absoluta filter. Även om du vill ge någon access till dina privata servrar från utsidan så kan du inte göra detta utan att ge alla access till servrarna.

Linux innehåller mjukvara för paketfiltrering i kärnan från och med version 1.3.x.

2.1.2 Proxyservrar

Proxyservrar tillåter indirekt access till Internet genom brandväggen. Det bästa exemplet på hur detta fungerar är en person som telnettar till ett system och sedan telnettar därifrån till ytterligare ett system. Men med en proxyserver så görs detta automatiskt. När du ansluter till en proxyserver med din klientprogramvara så startar proxyservern sin klientprogramvara och skickar informationen till dig.

Eftersom proxyservrar duplicerar all kommunikation så kan de logga allt de gör.

Den stora fördelen med proxyservrar är att de är helt säkra, när de är korrekt konfigurerade. De tillåter inte några anslutningar igenom dem. Det finns inga direkta IP-routes.

3 Att konfigurera brandväggen

3.1 Hårdvarukrav

I vårt exempel så är datorn en 486-DX66 med 16Mb minne och en 500Mb Linuxpartition. Detta systemet har två nätverkskort, ett anslutet till vårt privata LAN och det andra anslutet till ett LAN som vi kallar för den de-militariserade zonen (DMZ). DMZ är ansluten till en router som i sin tur är ansluten till Internet.

Detta är en ganska typisk konfiguration för ett företag. Man skulle kunna använda ett nätverkskort och ett modem med en PPP-länk till Internet. Poängen är att brandväggen måste ha två IP-adresser.

Jag vet många personer som har små LANs hemma med två eller tre datorer anslutna. Något som man skulle kunna tänka sig då är att koppla alla sina modem till en Linuxburk (kanske en gammal 386:a) och ansluta dem till Internet med lastbalansering. Med denna konfiguration kan man fördubbla genomströmmningen om endast en person tog hem data från Internet. :-)

4 Programvara för brandväggar

4.1 Tillgängliga paket

Om allt du vill ha är en filtrerande brandvägg, så behöver du endast Linux och de grundläggande nätverkspaketen. Ett paket som kanske inte följer med din distribution är IP Firewall Administration tool, *ipfwadm*. Detta kan du hitta på *www.xos.nl* `<http://www.xos.nl/linux/ipfwadm/>`.

Om du vill sätta upp en proxyserver så behöver du ett av följande paket.

1. SOCKS
2. TIS Firewall Toolkit (FWTK)

4.2 TIS Firewall Toolkit eller SOCKS

Trusted Information System `<http://www.tis.com>` har gjort en samling program som är designade för brandväggar. Programmen gör i stort sett samma sak som SOCKS-paketet, men med en annan designstrategi. Där SOCKS har ett program som täcker alla Internettransaktioner, så har TIS ett program för varje tjänst som önskar använda brandväggen.

För att jämföra de båda, låt oss använda exemplet med *www* och *telnet* access. Med SOCKS sätter du upp en konfigurationsfil och en daemon. Genom denna fil och daemon så aktiveras både *www* och *telnet*, såväl som alla andra tjänster som du inte stängt av.

Med TIS-paketet kan du sätta upp en daemon för var och en av *www* och *telnet*, såväl som konfigurationsfiler för båda. När du gjort detta så är andra accesser till Internet fortfarande inte tillåtna förrän du explicit sätter upp dem också. Om det inte har tillhandahållits en daemon för en viss tjänst (såsom *talk*) så finns det en "plug-in" daemon, men den är varken så flexibel eller så enkel att sätta upp som de andra daemonerna.

Detta kan verka obetydligt, men det har en stor betydelse. SOCKS tillåter dig att vara slarvig. Med en dålig konfigurerad SOCKS-server, så kan någon från insidan få mer access till Internet än vad som var meningen från början. Med TIS-paketet så har personerna på insidan endast tillgång till det som systemadministratören vill att de skall ha tillgång till.

SOCKS är lättare att konfigurera, lättare att kompilera och tillåter större flexibilitet. TIS-paketet är säkrare om man vill reglera användarna i det skyddade nätverket. Båda tillhandahåller absolut skydd från utsidan.

Jag kommer att täcka installationen och konfigurationen av båda.

5 Att förbereda Linuxsystemet

5.1 Att kompilera kärnan

Börja med en ren installation av din Linuxdistribution (jag använde RedHat 3.0.3 och exemplen här baseras på den distributionen). Ju mindre programvara du har installerad desto mindre säkerhetshål, bakdörrar och/eller buggar kommer det att finnas som inbringar säkerhetsproblem i ditt system. Så installera endast minimalt med applikationer.

Välj en stabil kärna. Jag använde Linux 2.0.14 kärnan till mitt system, så denna dokumentationen är baserad på dess inställningar.

Du kommer att behöva kompilera om Linuxkärnan med passande altenativ. Vid denna tidpunkten bör du titta i Kernel HOWTO, Ethernet HOWTO och NET-3 HOWTO om du inte redan gjort detta.

Här är de nätverksrelaterade inställningar som jag vet fungerar i 'make config'

1. Under 'General setup'
   (a) Sätt på 'Networking Support'
2. Under 'Networking Options'
   (a) Sätt på 'Network firewalls'
   (b) Sätt på 'TCP/IP Networking'
   (c) Stäng av 'IP forwarding/gatewaying' (OM DU INTE vill använda IP-filtrering)
   (d) Sätt på 'IP Firewalling'
   (e) Sätt på 'IP firewall packet logging' (detta är inte nödvändigt men det är en bra ide)
   (f) Stäng av 'IP: masquerading' (Jag täcker inte detta ämnet här.)
   (g) Sätt på 'IP: accounting'
   (h) Stäng av 'IP: tunneling'
   (i) Stäng av 'IP: aliasing'
   (j) Stäng av 'IP: PC/TCP compatibility mode'
   (k) Stäng av 'IP: Reverse ARP'
   (l) Sätt på 'Drop source routed frames'
3. Under 'Network device support'
   (a) Sätt på 'Network device support'
   (b) Sätt på 'Dummy net driver support'
   (c) Sätt på 'Ethernet (10 or 100Mbit)'
   (d) Välj ditt nätverkskort

Nu kan du kompilera om och installera om kärnan och sedan starta om systemet. Dina nätverkskort skall nu dyka upp bland all text vid uppstarten. Om inte, gå igenom de andra HOWTOerna igen tills det fungerar.

5.2 Att konfigurera två nätverkskort

Om du har två nätverkskort i din dator, så måste du med all sannolikhet lägga till ett appenduttryck i din `/etc/lilo.conf`-fil för att tala om IRQ och I/O-adress för båda korten. Mitt appenduttryck ser ut så här:

```
append="ether=12,0x300,eth0 ether=15,0x340,eth1"
```

5.3 Att konfigurera nätverksadresser

Detta är den riktigt intressanta delen. Nu har du några olika val att göra. Eftersom vi inte vill att Internet skall ha tillgång till någon del av det privata nätverket, så behöver vi inte använda riktiga adresser. Det finns ett antal intressanta adresser som är reserverade för privata nätverk. Eftersom alla behöver fler adresser och eftersom dessa adresserna inte kan nås över Internet så är de ett bra val.

Ibland dessa adresser så är tex 192.168.2.xxx reserverad och vi kommer att använda den i våra exempel.

Din proxybrandvägg kommer att vara medlem av båda nätverken och kan därför skicka igenom data till och från det privata nätverket.

```
        199.1.2.10   __________   192.168.2.1
  _  __  _        \ |          | /          _______________
 | \/  \/ |        \| Firewall |/          |               |
/ Internet \--------|  System  |-----------| Workstation/s |
\_/\_/\_/\_/        |__________|           |_______________|
```

Om du skall använda en filtrerande brandvägg så kan du fortfarande använda de här adresserna. Du behöver använda IP-maskering (Masquerading) för att det skall fungera. Med IP-maskering så skickar brandväggen vidare alla paket och översätter dem till "riktiga" IP-paket så att de kan skickas på Internet.

Du måste tilldela den riktiga IP-adressen till nätverkskortet som är anslutet till Internetsidan (utsidan). Och tilldela 192.168.2.1 till Ethernet-kortet på insidan. Detta kommer att vara din proxy/gateway IP-adress. Du kan tilldela alla de andra maskinerna på det skyddade nätverket någon adress i området 192.168.2.2 till 192.168.2.254.

Eftersom jag använder RedHat Linux så konfigurerar jag nätverket vid uppstart genom att lägga till en `ifcfg-eth1`-fil i katalogen `/etc/sysconfig/network-scripts`. Denna filen läses under uppstarten för att konfigurera ditt nätverk och dina routingtabeller.

Så här ser min `ifcfg-eth1` ut:

```
#!/bin/sh
#>>>Device type: ethernet
#>>>Variable declarations:
DEVICE=eth1
IPADDR=192.168.2.1
NETMASK=255.255.255.0
NETWORK=192.168.2.0
BROADCAST=192.168.2.255
GATEWAY=199.1.2.10
ONBOOT=yes
#>>>End variable declarations
```

Du kan också använda dessa script för att automatiskt ansluta till Internet med modem. Titta på scriptet `ipup-ppp`.

Om du skall använda ett modem till din Internetanslutning så kommer din IP-adress på utsidan att tilldelas av din leverantör när du ansluter.

5.4 Att testa ditt nätverk

Börja med med att kolla *ifconfig* och *route*. Om du har två nätverkskort så bör din *ifconfig* se ut ungefär så här:

```
#ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.0  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          RX packets:1620 errors:0 dropped:0 overruns:0
          TX packets:1620 errors:0 dropped:0 overruns:0

eth0      Link encap:10Mbps Ethernet  HWaddr 00:00:09:85:AC:55
          inet addr:199.1.2.10 Bcast:199.1.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:12 Base address:0x310

eth1      Link encap:10Mbps Ethernet  HWaddr 00:00:09:80:1E:D7
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:15 Base address:0x350
```

och din routingtabell ungefär så här:

```
#route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
199.1.2.0       *               255.255.255.0   U     1500   0        15 eth0
192.168.2.0     *               255.255.255.0   U     1500   0         0 eth1
127.0.0.0       *               255.0.0.0       U     3584   0         2 lo
default         199.1.2.10      *               UG    1500   0        72 eth0
```

Observera: 199.1.2.0 är Internet-sidan av denna brandväggen och 192.168.2.0 är den privata sidan.

Försök nu att *ping*a Internet från brandväggen. Jag brukade använda `nic.ddn.mil` som testpunkt. Det är fortfarande en bra test, men har visat sig mindre pålitlig än jag hade hoppats. Om det inte fungerar med en gång, försök med att *ping*a några andra ställen som inte är anslutna till ditt LAN. Om detta inte fungerar så är din PPP felaktigt konfigurerad. Läs NET-3 HOWTO igen och försök sedan igen.

Försök nu *ping*a en dator på det skyddade nätverket från brandväggen. Alla datorer på det privata nätverket skall kunna *ping*a varandra. Om inte så försök med NET-3 HOWTO igen och jobba lite till på nätverket.

Försök nu att *ping*a brandväggens yttre IP-adress (observera att detta inte är en av 192.168.2.xxx adresserna) från en dator inne i det skyddade nätverket. Om detta fungerar så har du inte stängt av IP-vidareskickning (forwarding). Se till att vara säker på att detta är vad du vill i sådana fall. Om du låter den vara påslagen så måste du även läsa igenom avsnittet om IP-filtrering också.

Försök nu att *ping*a Internet ifrån det skyddade nätverket (dvs bakom brandväggen). Använd samma adress som fungerade när du gjorde detta från brandväggen (tex nic.ddn.mil). Återigen, om du har stängt av IP-vidareskickning så skall detta inte fungera men annars så skall det fungera.

Om du har IP-vidareskickning påslagen och använder "RIKTIGA" (inte 192.168.2.*) IP-adresser till ditt privata nätverk, och om du inte kan *ping*a Internet men du kan *ping*a Internet-sidan av din brandvägg, så kolla om nästa router 'uppströms' routar paket för din privata nätverksadress (din leverantör kanske måste göra detta åt dig).

Om du har gett ditt nätverk reserverade adresser (192.168.2.*), så kan inga paket routas till det ändå. Om du har gått i förväg och redan har IP-maskering påslagen, så skall detta testet fungera.

Nu har du din grundläggande systemkonfiguration.

5.5 Att säkra brandväggen

En brandvägg är inte till någon nytta om den lämnas vidöppen för attacker genom en oanvänd tjänst. En "bad guy" skulle kunna få tillgång till brandväggen och modifiera den så att den passar hans egna syften.

Börja med att stänga av alla tjänster som inte behövs. Titta på filen `/etc/inetd.conf`. Denna filen kontrollerar vad som kallas "super servern". Den i sin tur kontrollerar en mängd andra serverdaemoner och startar dem när de efterfrågas.

Stäng definitivt av *netstat*, *systat*, *tftp*, *bootp* och *finger*. För att stänga av en tjänst, sätt ett # som första tecken på raden som beskriver tjänsten. När du är färdig så skickar du en SIG-HUP signal till *inetd*-processen genom att skriva "**kill -HUP <pid>**", där <pid> är processnumret för *inetd*. Detta gör att *inetd* läser om sin konfigurationsfil (`inetd.conf`) och återstartas.

Testa detta genom att telnetta till port 15 (porten för *netstat*) på brandväggen. Om du får en utskrift från *netstat* så har du inte återstartat *inetd* korrekt.

6 Att konfigurera IP-filtrering (*ipfwadm*)

Till att börja med så bör du ha IP-vidareskickning (IP Forwarding) påslagen i din kärna och ditt system skall vara igång och skicka vidare allt som du skickar till den. Dina routingtabeller skall vara på plats och du skall kunna ha tillgång till allt, både från insidan till utsidan och från utsidan till insidan.

Men vi bygger en brandvägg så vi behöver börja begränsa vad som går att få tillgång till.

I mitt system så skapade jag ett par script för att ställa in brandväggens vidareskickningspolicy och redovisningspolicy. Jag kör dessa script ifrån scripten i `/etc/rc.d` så att mitt system konfigureras vid uppstart.

Med standardinställningarna så skickar Linuxkärnans system för IP-vidareskickning vidare allt. På grund av detta så bör din brandvägg börja med att neka access till allt och tömma alla *ipfwadm*-regler som finns kvar sedan det kördes sist. Följande script fixar detta:

```
#
# 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
```

Nu har vi den ultimata brandväggen. Inget kan komma igenom. Du har utan tvekan några tjänster som du behöver skicka vidare genom brandväggen, så här kommer några exempel som kan vara användbara för dig:

```
#  Skicka vidare e-post till din server
ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.10 25

# Skicka vidare e-postanslutningar till utomstående e-postservrar
ipfwadm -F -a accept -b -P tcp -S 196.1.2.10 25 -D 0.0.0.0/0 1024:65535

# Skicka vidare www-anslutningar till din www-server
/sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 196.1.2.11 80

# Skicka vidare www-anslutningar till utomstående www-server
/sbin/ipfwadm -F -a accept -b -P tcp -S 196.1.2.* 80 -D 0.0.0.0/0 1024:65535

# Skicka vidare DNS-trafik
/sbin/ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 196.1.2.0/24
```

Du kanske också är intresserad av att hålla reda på trafik som går igenom din brandvägg. Följande script kommer att räkna alla paket. Du kan lägga till en eller ett par rader för att hålla reda på paket till ett enskilt system.

```
# Flush the current accounting rules
ipfwadm -A -f
# Accounting
/sbin/ipfwadm -A -f
/sbin/ipfwadm -A out -i -S 196.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A out -i -S 0.0.0.0/0 -D 196.1.2.0/24
/sbin/ipfwadm -A in -i -S 196.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A in -i -S 0.0.0.0/0 -D 196.1.2.0/24
```

Om allt du ville ha var en filtrerande brandvägg så kan du sluta här. Ha skoj :-)

7 Att installera TIS proxyserver

7.1 Att få tag på programvaran

TIS FWTK finns på *TIS ftp-sajt* <ftp://ftp.tis.com>.

Gör inte mistaget som jag gjorde. När du hämtar filer från TIS, LÄS README-filerna. TIS FWTK är inlåst i en gömd katalog på deras server. TIS kräver att du **sickar ett e-postmeddelande till *fwtk-request@tis.com*** <mailto:fwtk-request@tis.com> med endast ordet **SEND** i textkroppen för att få reda på namnet på den gömda katalogen. Det behövs inget 'subject' i meddelandet. Deras system kommer då att e-posta tillbaka namnet på katalogen (som gäller i tolv timmar) så att du kan ladda hem källkoden.

När detta skrivs så har TIS släppt version 2.0 (beta) av FWTK. Denna versionen verkar kompilera bra (med ett par undantag) och allt fungerar för mig. Detta är versionen som jag täcker här. När de släpper den slutgiltliga koden så kommer jag att uppdatera HOWTOn.

För att installera FWTK, skapa en `fwtk-2.0` katalog i din `/usr/src` katalog. Flytta din kopia av FWTK (`fwtk-2.0.tar.gz`) till denna katalogen och packa upp den (`tar zxf fwtk-2.0.tar.gz`).

FWTK kan inte använda proxyn på SSL www-dokument men det finns en addon för detta skriven av Jean-Christophe Touvet. Den finns på *ftp.edelweb.fr* `<ftp://ftp.edelweb.fr/pub/contrib/fwtk/ssl-gw.tar.Z>`. Touvet supportar inte denna koden.

Jag använder en modifierad version som inkluderar access till Netscapes säkra nyhetsservrar som är skriven av Eric Wedel. Den finns tillgänglig på *mdi.meridian-data.com* `<ftp://mdi.meridian-data.com/pub/tis.fwtk/ssl-gw/ssl-gw2.tar.Z>`.

I vårt exempel kommer jag att använda Eric Wedels version.

För att installera, skapa en `ssl-gw` katalog i din `/usr/src/fwtk-2.0` katalog och lägg filerna i den.

När jag installerade denna gatewayen så krävdes det lite ändringar innan den kunde kompileras tillsammans med resten av paketet.

Den första ändringen skall göras i filen `ssl-gw.c`. Jag kom på att den inte inkluderade en fil som den behövde.

```
#if defined(__linux)
#include        <sys/ioctl.h>
#endif
```

För det andra så följde det inte med någon `Makefile`. Jag kopierade en ifrån en av de andra gatewayernas kataloger och ersatte den gatewayens namn med `ssl-gw`.

7.2 Att kompilera TIS FWTK

Version 2.0 av FWTK kompilerar mycket lättare än någon av de tidigare versionerna. Jag hittade ett par saker som behövde ändras innan BETA-versionen kompilerade ordentligt. Förhoppningsvis så finns dessa ändringar i den slutgiltliga versionen.

För att fixa till detta så börja med att byta till katalogen `/usr/src/fwtk/fwtk` och kopiera filen `Makefile.config.linux` till `Makefile.config`.

KÖR INTE FIXMAKE. Instruktionerna säger åt dig att köra den. Om du gör det så kommer den att förstöra `Makefile`-filerna i alla kataloger.

Jag har en fix för fixmake. Problemet är att sed-scriptet lägger till en ’.’ och ” i `include`-raden i alla `Makefile`-filerna. Följande sed-script fungerar.

```
sed 's/^include[        ]*\([^  ].*\)/include \1/' $name .proto > $name
```

Efter detta måste vi editera filen `Makefile.config`. Det finns två ändringar som du kanske måste göra.

Författaren ställde in käll-katalogen till sin hemkatalog. Vi kompilerar våran kod i `/usr/src` så du bör ändra variabeln `FWTKSRCDIR` så att det stämmer överens.

```
FWTKSRCDIR=/usr/src/fwtk/fwtk
```

Sedan, vissa Linuxsytem använder gdbm-databasen. `Makefile.config` använder dbm. Du kanske behöver ändra detta. Jag behövde göra det för RedHat 3.0.3.

```
DBMLIB=-lgdbm
```

Sista fixen är i `x-gw`. Buggen är i BETA-versionen av `socket.c` koden. För att fixa det, ta bort följande rader från koden.

```
#ifdef SCM_RIGHTS  /* 4.3BSD Reno and later */
                     + sizeof(un_name->sun_len) + 1
#endif
```

Om du lade till `ssl-gw` till din FWTK källkatalog så måste du lägga till den till listan av kataloger i `Makefile`.

```
DIRS=   smap smapd netacl plug-gw ftp-gw tn-gw rlogin-gw http-gw x-gw ssl-gw
```

Nu kan du köra `make`.

7.3 Att installera TIS FWTK

Kör `make install`.

Standardkatalogen för installation är `/usr/local/etc`. Du kan ändra detta (jag gjorde det inte) till en säkrare katalog. Jag valde att ändra rättigheterna på denna katalogen till `chmod 700`.

Allt som är kvar nu är att konfigurera brandväggen.

7.4 Att konfigurera TIS FWTK

Nu börjar det roliga. Vi måste lära systemet att använda dessa nya tjänster och skapa tabellerna för att kontrollera dem.

Jag tänker inte försöka skriva om manualen för TIS FWTK här. Jag kommer att visa dig inställningarna som fungerade för mig och förklara de problem som jag sprang på och hur jag kom runt dem.

Det finns tre filer som utgör dessa kontroller.

- `/etc/services`
  - Talar om för systemet vilka portar tjänsterna finns på.
- `/etc/inetd.conf`
  - Talar om för *inetd* vilket program som skall köras när någon ‘knackar’ på en tjänsts port.
- `/usr/local/etc/netperm-table`
  - Talar om för FWTK-tjänsterna vem som skall tillåtas och vem som skall nekas access till tjänster.

För att få igång FWTK så bör du editera dessa filer nerifrån och upp. Att editera `services` utan att `inetd.conf` eller `netperm-table` är korrekt inställda kan göra ditt system oåtkomligt.

7.4.1 Filen /usr/local/etc/netperm-table

Denna filen kontrollerar vem som får tillgång till tjänsterna i TIS FWTK. Du bör tänka på trafiken som använder brandväggen från båda sidor. Personer utanför ditt nätverk skall identifiera sig innan de får tillgång, men personer inuti ditt nätverk kan tillåtas att passera rakt igenom.

Brandväggen använder ett program som heter *authsrv* för att hålla en databas med användarnamn och lösenord, så att personer kan identifiera sig. Autentiseringsdelen av `netperm-table` kontrollerar var databasen finns och vem som kan komma åt den.

Jag hade lite problem med att stänga av access till denna tjänsten. Notera att *permit-hosts*-raden som jag visar innehåller en '*' vilket ger alla access. Den korrekta inställningen är `authsrv: premit-hosts localhost` om du kan få det att fungera.

```
#
# 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
```

För att initialisera databasen, *su*:a till *root* och kör `./authsrv` i katalogen `/var/local/etc` för att skapa administratörens användarprofil. Här följer en exempelsession.

Läs dokumentationen för FWTK för att lära dig hur man lägger till användare och grupper.

```
#
# 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
#
```

Kontrollerna för *telnet*-gatewayen (`tn-gw`) är okomplicerade och är de första som du bör sätta upp.

I mitt exempel så tillåter jag datorer från det privata nätverket att passera igenom brandväggen utan att autentisera sig (`permit-hosts 196.1.2.* -passok`). Men alla andra användare måste lämna användarid och lösenord för att använda proxyn (`permit-hosts * -auth`).

Jag låter även ett annat system (196.1.2.202) få tillgång till brandväggen utan att gå igenom brandväggen över huvud taget. Raderna med `inetacl-in.telnetd` fixar detta. Jag förklarar hur dessa rader anropas senare.

Timeouten för *telnet* bör vara kort.

```
# 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 196.1.2.* -passok -xok
tn-gw:               permit-hosts * -auth
# Only the Administrator can telnet directly to the Firewall via Port 24
netacl-in.telnetd: permit-hosts 196.1.2.202 -exec /usr/sbin/in.telnetd
```

Det fungerar på samma sätt för *r-kommandona* som för *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 196.1.2.* -passok -xok
rlogin-gw:    permit-hosts * -auth -xok
# Only the Administrator can telnet directly to the Firewall via Port
netacl-rlogind: permit-hosts 196.1.2.202 -exec /usr/libexec/rlogind -a
```

Du bör inte låta någon få direkt tillgång till din brandvägg och det inkluderar FTP, så lägg inte en FTP-server på din brandvägg.

Nu till *ftp-gw*. Återigen så gör raden med `permit-hosts` att alla i det skyddade nätverket får fri tillgång till Internet men alla andra måste autentisera sig. Jag har även med loggning av alla filer som sänds och tas emot av mina kontroller (`-log { retr stor }`).

Timeouten för FTP kontrollerar hur lång tid det kommer att ta att släppa en dålig förbindelse och dessutom hur lång tid en anslutning hålls öppen utan aktivitet.

```
# 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 196.1.2.* -log { retr stor }
ftp-gw:              permit-hosts * -authall -log { retr stor }
```

WWW, gopher och bläddrarbaserad FTP kontrolleras av *http-gw*. De två första raderna skapar en katalog för att spara *ftp* och *www* dokument allteftersom de passerar genom brandväggen. Jag gör så att *root* blir ägare till filerna och så att endast *root* kan läsa katalogen.

Timeouten för *www* skall vara kort. Den kontrollerar hur lång tid en användare skall vänta på dåliga förbindelser.

```
# 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           196.1.2.* -log { read write ftp }
http-gw:      deny-hosts      *
```

ssl-gw är egentligen bara en 'släpp igenom allt'-gateway. Var försiktig med den. I detta exemplet låter jag vem som helst i det skyddade nätverket ansluta till vilken server som helst utanför nätverket, förutom adresserna 127.0.0.* och 192.1.1.*, och då endast till portarna 443 till 563. Portarna 443 till 563 är kända SSL-portar.

```
# ssl gateway rules:
ssl-gw:         timeout 300
ssl-gw:         hosts           196.1.2.* -dest { !127.0.0.* !192.1.1.* *:443:563 }
ssl-gw:         deny-hosts      *
```

Här är ett exempel på hur man använder *plug-gw* för att tillåta anslutningar till en nyhetsserver. I detta exemplet låter jag vem som helst i det skyddade nätverket ansluta endast till ett system och då endast till dess *news*-port.

Den andra raden tillåter nyhetsservern att skicka sin data tillbaka till det skyddade nätverket.

Eftersom de flesta klienter förväntas hålla kvar anslutningen medans användaren läser nyheter så bör timeouten vara lång.

```
# NetNews Pluged gateway
plug-gw:        timeout 3600
plug-gw: port nntp 196.1.2.* -plug-to 199.5.175.22 -port nntp
plug-gw: port nntp 199.5.175.22 -plug-to 196.1.2.* -port nntp
```

Gatewayen för *finger* är enkel. Alla i det skyddade nätverket måste först logga in och sedan tillåts de att använda *finger*-programmet på brandväggen. Alla andra får bara ett meddelande.

```
# Enable finger service
netacl-fingerd: permit-hosts 196.1.2.* -exec /usr/libexec/fingerd
netacl-fingerd: permit-hosts * -exec /bin/cat /usr/local/etc/finger.txt
```

Jag har inte satt upp *Mail* och *X-Windows* tjänsterna än så jag inkluderar inte några exempel. Om någon har fungerande exempel så får ni gärna e-posta dem till mig.

7.4.2 Filen `/etc/inetd.conf`

Här följer en komplett `/etc/inetd.conf` fil. Alla onödiga tjänster har kommenterats bort. Jag har med hela filen för att visa vad som bör stängas av, såväl som för att visa hur man konfigurerar de nya tjänsterna för brandväggen.

```
#echo stream  tcp  nowait  root      internal
#echo dgram   udp  wait    root      internal
#discard      stream  tcp  nowait  root      internal
#discard      dgram   udp  wait    root      internal
#daytime      stream  tcp  nowait  root      internal
#daytime      dgram   udp  wait    root      internal
#chargen      stream  tcp  nowait  root      internal
#chargen      dgram   udp  wait    root      internal
# FTP firewall gateway
ftp-gw     stream  tcp  nowait.400  root  /usr/local/etc/ftp-gw  ftp-gw
# Telnet firewall gateway
telnet        stream  tcp  nowait     root  /usr/local/etc/tn-gw /usr/local/etc/tn-gw
# local telnet services
telnet-a    stream  tcp  nowait     root  /usr/local/etc/netacl in.telnetd
# Gopher firewall gateway
gopher        stream  tcp  nowait.400  root  /usr/local/etc/http-gw /usr/local/etc/http-gw
# WWW firewall gateway
http  stream  tcp  nowait.400  root  /usr/local/etc/http-gw /usr/local/etc/http-gw
# SSL firewall gateway
ssl-gw  stream  tcp    nowait  root /usr/local/etc/ssl-gw   ssl-gw
# NetNews firewall proxy (using plug-gw)
nntp    stream  tcp    nowait  root   /usr/local/etc/plug-gw plug-gw nntp
#nntp stream  tcp    nowait  root   /usr/sbin/tcpd  in.nntpd
# SMTP (email) firewall gateway
#smtp stream  tcp    nowait  root   /usr/local/etc/smap smap
#
# Shell, login, exec and talk are BSD protocols.
#
#shell        stream  tcp    nowait  root   /usr/sbin/tcpd  in.rshd
#login        stream  tcp    nowait  root   /usr/sbin/tcpd  in.rlogind
#exec stream  tcp    nowait  root   /usr/sbin/tcpd  in.rexecd
#talk dgram   udp    wait    root   /usr/sbin/tcpd  in.talkd
#ntalk        dgram   udp    wait    root   /usr/sbin/tcpd  in.ntalkd
#dtalk        stream  tcp    waut    nobody  /usr/sbin/tcpd  in.dtalkd
#
# Pop and imap mail services et al
#
#pop-2   stream  tcp  nowait  root  /usr/sbin/tcpd    ipop2d
#pop-3   stream  tcp  nowait  root  /usr/sbin/tcpd    ipop3d
#imap    stream  tcp  nowait  root  /usr/sbin/tcpd    imapd
#
# The Internet UUCP service.
#
#uucp    stream  tcp  nowait  uucp  /usr/sbin/tcpd  /usr/lib/uucp/uucico -l
#
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers." Do not uncomment
# this unless you *need* it.
#
```

```
#tftp dgram   udp     wait    root    /usr/sbin/tcpd  in.tftpd
#bootps       dgram   udp     wait    root    /usr/sbin/tcpd  bootpd
#
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers."  Many sites choose to disable
# some or all of these services to improve security.
#
# cfinger is for GNU finger, which is currently not in use in RHS Linux
#
finger        stream  tcp  nowait  root   /usr/sbin/tcpd  in.fingerd
#cfinger      stream  tcp  nowait  root   /usr/sbin/tcpd  in.cfingerd
#systat       stream  tcp  nowait  guest  /usr/sbin/tcpd  /bin/ps -auwwx
#netstat      stream  tcp  nowait  guest  /usr/sbin/tcpd  /bin/netstat -f inet
#
# Time service is used for clock syncronization.
#
#time stream  tcp  nowait  root  /usr/sbin/tcpd  in.timed
#time dgram   udp  wait    root  /usr/sbin/tcpd  in.timed
#
# Authentication
#
auth          stream  tcp  wait    root  /usr/sbin/tcpd  in.identd -w -t120
authsrv       stream  tcp  nowait  root  /usr/local/etc/authsrv authsrv
#
# End of inetd.conf
```

7.4.3 Filen `/etc/services`

Det är här allting börjar. När en klient ansluter till brandväggen så ansluter den till en känd port (mindre än 1024). Till exempel så ansluter *telnet* på port 23. Daemonen för *inetd* hör anslutningen och letar upp namnet för tjänsten i filen `/etc/services`. Sedan startar den programmet som hör ihop med namnet i filen `/etc/inetd.conf`.

Vissa av tjänsterna som vi skapar finns inte normalt i filen `/etc/services`. Du kan para ihop vissa av dem med vilken port du vill. Till exempel så har jag låtit administratörens *telnet*-port (telnet-a) vara port 24. Du skulle kunna sätta den till 2323 om du ville. För att administratören (DU) skall kunna ansluta direkt till brandväggen så måste han telnetta till port 24 och inte 23, och om du sätter upp filen `netperm-table` som jag gjorde så kan detta endast göras från en maskin i ditt skyddade nätverk.

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

8 SOCKS Proxyserver

8.1 Att installera proxyservern

SOCKS proxyserver finns att hämta ifrån *sunsite* `<ftp://sunsite.unc.edu/pub/Linux/system/Network/misc/socks-linux-src.tgz>`. Det finns även ett exempel på en konfigurationsfil i den katalogen som heter `socks-conf`. Packa upp filerna till en katalog på ditt system och följ instruktionerna för hur man kompilerar det. Jag hade ett par problem när jag gjorde detta. Se till att dina `Makefile`-filer är korrekta.

En viktig sak att notera är att proxyservern måste läggas till i `inetd.conf`. Du skall lägga till en rad

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

så att servern kan köras då den efterfrågas.

8.2 Att konfigurera proxyservern

SOCKS-programmet använder två konfigurationsfiler. En som beskriver vilken access som tillåts, och en som routar förfrågningar till den korrekta proxyservern. Access-filen skall finnas på servern och routing-filen skall finnas på alla Un*x-maskiner. DOS och Macintosh datorer gör sin egen routing.

8.2.1 Accessfilen

Med socks4.2 Beta så heter accessfilen `sockd.conf`. Den skall innehålla två rader, en tillåt-rad och en neka-rad. Varje rad har tre element:

- Identifieraren (permit/deny)
- IP-adressen
- Adressmodifieraren

Identifieraren är antingen *permit* eller *deny*. Du skall ha en rad för varje.

IP-adressen är en fyrabytes adress i vanligt format (tex 192.168.2.0).

Adressmodifieraren är också en typisk IP-adress. Den fungerar som en nätmask. Tänk dig adressen som 32-bitars tal. Om en bit är en '1' så måste den motsvarande biten i adressen den kontrollerar vara samma som den i IP-adressen. Till exempel om raden ser ut så här

```
permit 192.168.2.23 255.255.255.255
```

så tillåter den endast den IP-adress där alla bitar är samma som i 192.168.2.23, dvs endast 192.168.2.23. Följande rad

```
permit 192.168.2.0 255.255.255.0
```

tillåter alla adresser i gruppen 192.168.2.0 till 192.168.2.255, dvs hela klass C domänen. Man skall inte ha följande rad

```
permit 192.168.2.0 0.0.0.0
```

eftersom den tillåter alla adresser, oavsett vilken det är.

Så, tillåt först alla adresser som du vill tillåta och neka sedan resten. För att tillåta alla i domänen 192.168.2.xxx fungerar följande rader bra.

```
permit 192.168.2.0 255.255.255.0
deny 0.0.0.0 0.0.0.0
```

Notera den första "0.0.0.0" i *deny*-raden. Med en modifierare som är 0.0.0.0 så spelar fältet med IP-adressen ingen roll. Men att ha endast nollor är normen eftersom det är enkelt att skriva.

Det är tillåtet med mer än en rad av varje.

Specifika användare kan också ges eller nekas access. Detta görs via *ident* autentisering. Alla system stöder inte *ident*, tex Trumpet Winsock, så jag tar inte upp detta här. Dokumentationen för SOCKS täcker detta ganska bra.

8.2.2 Routingfilen

Routingfilen i SOCKS är dåligt namnsatt till `socks.conf`. Jag säger "dåligt namnsatt" eftersom det är så likt namnet på accessfilen så det kan vara lätt att blanda ihop dem.

Routingfilen finns för att tala om för SOCKS-klienterna när de skall använda SOCKS eller inte. Till exempel i vårat nätverk så behöver inte 192.168.2.3 använda SOCKS för att kommunicera med 192.168.2.1 (brandväggen). Den har en direkt anslutning via Ethernet. Den definierar loopback-enheten, 127.0.0.1, automatiskt. Naturligtvis så behövs inte SOCKS för att kommunicera med sig själv. Det finns tre nyckelord:

- deny
- direct
- sockd

`deny` talar om när SOCKS skall neka en förfrågan. Här används samma tre fält som i `sockd.conf`: identifierare, adress och modifierare. Generellt, eftersom detta även kontrolleras av accessfilen `sockd.conf`, så sätts fältet för modifieraren till 0.0.0.0. Om du vill göra så att du inte kan kontakta någon dator, så kan du göra det här.

Raden med `direct` bestämmer vilka adresser man inte skall använda SOCKS för. Detta är adresser vilka man kan nå utan att gå igenom proxyservern. Återigen har vi de tre fälten identifierare, adress och modifierare. Vårt exempel skulle ha

```
direct 192.168.2.0 255.255.255.0
```

vilket ger direkt kontakt med alla datorer på vårt skyddade nätverk.

Raden med `sockd` talar om vilken dator som har daemonen för SOCKS-servern på sig. Syntaxen är

```
sockd @=<serverlist> <IP address> <modifier>
```

Notera '@='. Detta låter dig specificera IP-adresserna för en lista av proxyservrar. I vårt exempel så använder vi endast en proxyserver. Men man kan ha många, för att tillåta större belastning och för redundans i fall något går sönder.

Fälten för IP-adress och modifierare fungerar som för de andra exemplena. Du specificerar vilka adresser som går vart genom dessa.

8.3 DNS bakom en brandvägg

Att sätta upp en DNS innanför en brandvägg är en ganska enkel uppgift. Du behöver bara sätta upp en DNS på brandväggen och sedan låta alla maskiner bakom brandväggen använda denna.

8.4 Att arbeta med en proxyserver

8.4.1 Unix

För att dina applikationer skall fungera med proxyservern så måste de vara "sockifierade". Du behöver två olika *telnet*ar, en för direkt kommunikation och en för kommuntikation via proxyservern. Med SOCKS följer instruktioner om hur man SOCKiferar ett program, såväl som ett par för-SOCKifierade program. Om du använder den SOCKifireade versionen för en direkt anslutning så kommer SOCKS automatiskt att byta till rätt version. På grund av detta så vill vi ändra namn på alla programmen på vårt skyddade nätverk och ersätta dem med de SOCKifierade versionerna. "Finger" blir "finger.orig", "telnet" blir "telnet.orig", osv. Du måste informera SOCKS om dessa via filen `include/socks.h`.

Vissa program tar hand om routing och sockifiering själva. Netscape är ett av dessa. Du kan använda en proxyserver under Netscape genom att skriva in serverns adress (192.168.2.1 i vårt fall) i SOCKS-fältet under Proxies. Man måste iallafall pilla lite med alla applikationer oavsett hur de hanterar en proxyserver.

8.4.2 MS Windows med Trumpet Winsock

Trumpet Winsock har inbyggt stöd för proxyserver. I "setup" menyn, skriv in IP-adressen för proxyservern och alla adresser som kan nås direkt. Trumpet hanterar sedan alla utgående paket.

8.4.3 Att få proxyservern att fungera med UDP-paket

SOCKS-paketet fungerar endast med TCP-paket, inte UDP. Detta gör det lite mindre användbart. Många användbara program, som *talk* och *Archie*, använder UDP. Det finns ett paket som är designat för att användas som en proxyserver för UDP-paket som heter UDPrelay, av Tom Fitzgerald <fitz@wang.com>. Tyvärr så är det inte, när detta skrivs, kompatibelt med Linux.

8.5 Nackdelar med proxyservrar

Proxyservern är, framför allt, en **säkerhetsenhet**. Att använda det för att öka tillgången till Internet med ett begränsat antal IP-adresser har många nackdelar. En proxyserver medför bättre access till Internet från insidan av det skyddade nätverket, men det gör insidan helt oåtkomligt från utsidan. Detta betyder att inga servrar, talk eller archieanslutningar, eller direkt e-post fungerar till maskinerna i det skyddade nätverket. Dessa nackdelar kan verka obetydliga, men tänk på det på detta sättet:

- Du har lagt en rapport som du håller på att jobba med på din dator i ett brandväggsskyddat nätverk. Du är hemma och bestämmer dig för att du vill gå igenom den. Det kan du inte. Du kan inte nå din dator eftersom den är bakom brandväggen. Du försöker att logga in på brandväggen först, men eftersom alla har proxyserveraccess, så har ingen lagt upp ett konto åt dig på den.

- Din dotter går på college. Du vill mäjla henne. Du har några privata saker att prata med henne om, och vill helst att hon skickar e-posten direkt till din maskin. Du litar helt och hållet på systemadministratören, men det är ju trots allt privat post. Men detta går inte.

- Oförmågan att använda UDP-paket är en stor nackdel med proxyservrar. Jag föreställer mig att detta kommer att fungera snart.

FTP skapar ett annat problem med en proxyserver. När man ger kommandot `ls`, så öppnar FTP-servern en socket på klientmaskinen och skickar informationen genom den. En proxyserver tillåter inte detta, så FTP fungerar inte.

Dessutom är proxyservrar långsamma. På grund av den större overheaden så är nästan alla andra sätt att få accessen snabbare.

Så om du har IP-adresserna och inte bekymrar dig för säkerheten så skall du inte använda en brandvägg och/eller proxyserver. Om du inte har IP-adresserna och inte bekymrar dig för säkerheten så kan du kanske använda en IP-emulator som *Term*, *Slirp* eller *TIA*. *Term* finns på *sunsite* <ftp://sunsite.unc.edu>, *Slirp* finns på *blitzen.canberra.edu.au* <ftp://blitzen.canberra.edu.au/pub/slirp> och *TIA* finns på *marketplace.com* <http://www.marketplace.com>. Dessa paket är snabbare, tillåter bättre anslutningar och tillhandahåller bättre tillgänglighet till det privata nätverket från Internet. Proxyservrar är bra för de nätverk som har många datorer som vill ansluta till Internet 'on the fly', med en inställning och lite arbete efter det.

9 Avancerade Konfigurationer

Det finns en konfiguration som jag vill gå igenom innan jag avslutar detta dokumentet. Det som jag precis har gått igenom räcker antagligen för de flesta. Men det som följer kommer att visa en mer avancerad konfiguration och kan dessutom klara upp lite frågor. Om du har frågor om det som jag just gått igenom, eller bara är intresserad av mångsidigheten hos proxyservrar och brandväggar, läs vidare.

9.1 Ett stort nätverk med tyngdpunkten på säkerhet

Antag till exempel att du är ledare för en milis och du vill ha ett nätverk. Du har 50 datorer och ett subnät med 32 (5 bitar) IP-adresser. Du behöver lite olika nivåer på access inom nätverket eftersom du säger olika saker till dina anhängare. Därför behöver du skydda vissa delar av nätverket från resten.

Nivåerna är:

1. **Den yttre nivån.** Denna nivån är öppen för alla. Det är här där du skrönar och skrävlar för att locka nya frivilliga.
2. **Troop**. Detta är nivån för personer som har kommit förbi den yttre nivån. Det är här du lär dem om den elaka statsmakten och hur man gör bomber.
3. **Mercenary.** Det är här de *riktiga* planerna finns. På denna nivån lagras all information om hur den 3:e världens regering skall ta över världen, dina planer härrörande Newt Gingrich, Oklahoma City och vad som verkligen finns i de där hangarerna i area 51.

9.1.1 Att sätta upp nätverket

IP-adresserna arrangeras som följer:

- En adress är 192.168.2.255, vilken är broadcastadressen och kan inte användas.
- 23 av de 32 IP-adresserna allokeras till 23 datorer som skall vara åtkomliga från Internet.

- En IP-adress går till en Linuxburk på det nätverket
- En går till en annan Linuxburk på det nätverket.
- Två IP-adresser går till routern.
- Fyra adresser blir över, men ges domännamnen paul, ringo, john och george, bara för att förvilla lite grann.
- De skyddade nätverken har båda adresserna 192.168.2.xxx

Sedan byggs två separata nätverk i olika rum. De routas via infraröd Ethernet så de är helt osynliga utanför rummen. Lyckligtvis så fungerar infraröd Ethernet precis som vanlig Ethernet.

Dessa båda nätverk kopplas till en av Linuxburkarna med en extra IP-adress.

Det finns en filserver som kopplar ihop de två skyddade nätverken. Detta för att planerna för att ta över världen involverar några av de högre trupperna. Filservern har adressen 192.168.2.17 för 'Troop'-nätverket och 192.168.2.23 för 'Mercenary'-nätverket. Den måste ha två olika IP-adresser eftersom den har två nätverkskort. IP-vidareskickning är avstängt.

IP-vidareskickning är även avstängt på de två Linuxburkarna. Routern kommer inte att skicka vidare paket ämnade för 192.168.2.xxx adresser om den inte explicit får sådana direktiv, så Internet kan inte komma in. Anledningen att vidareskickning skall vara avstängt här är att paket från 'Troop'-nätverket inte skall kunna nå 'Mercenary'-nätverket, och vice versa.

NFS-servern kan även ställas in för att erbjuda olika filer till de olika nätverken. Detta kan bli användbart, och lite trixande med symboliska länkar gör så att vissa filer kan delas av alla. Denna inställning och ett extra nätverkskort gör att denna filservern kan användas av alla tre nätverken.

9.1.2 Att sätta upp proxyservern

Eftersom alla tre nivåerna vill kunna ha koll på Internet för sina egna oärliga syften, så måste alla tre ha access till Internet. Det externa nätverket är direktkopplat till Internet så vi behöver inte anstränga oss med proxyservrar där. 'Mercenary' och 'Troop' nätverken är bakom brandväggar så vi behöver sätta upp proxyservrar där.

Båda nätverken kommer att sättas upp likartat. Båda är tilldelade samma IP-adresser. Jag lägger in några extra parametrar, bara för att göra det lite mer intressant.

1. Ingen kan använda filservern för access till Internet. Det skulle öppna filservern för virus och andra hemska saker, och den är hyfsat viktig, så den får inte användas mot Internet.
2. Vi kommer inte tillåta att trupperna har tillgång till World Wide Web. De är under utbildning och den typen av informationshämtande kan vara skadlig.

Så filen `sockd.conf` på 'Troop'-Linuxburken kommer ha en rad som den här:

```
deny 192.168.2.17 255.255.255.255
```

Och på 'Mercenary'-maskinen:

```
deny 192.168.2.23 255.255.255.255
```

Och, 'Troop'-Linuxburken har även en rad som denna:

```
deny 0.0.0.0 0.0.0.0 eq 80
```

Detta säger att man skall neka alla maskiner som försöker komma åt port 80 (http porten). Det tillåter fortfarande alla andra tjänster, endast www-access nekas.

Sedan så har båda filerna följande:

```
permit 192.168.2.0 255.255.255.0
```

för att tillåta alla datorer på 192.168.2.xxx nätverket att använda denna proxyserver, förutom de som redan har blivit nekade (dvs filservern och www-access från 'Troop'-nätverket).

Så filen `sockd.conf` på 'Troop'-nätverkets Linuxburk kommer se ut så här:

```
deny 192.168.2.17 255.255.255.255
deny 0.0.0.0 0.0.0.0 eq 80
permit 192.168.2.0 255.255.255.0
```

och på 'Mercenary'-nätverkets Linuxburk ser den ut så här:

```
deny 192.168.2.23 255.255.255.255
permit 192.168.2.0 255.255.255.0
```

Detta borde konfigurera allting korrekt. Alla nätverk är isolerade med lagom mängd interaktion. Alla borde vara lyckliga.

Nu, ta över världen!