Configuración de un Firewall en Solaris 9

Gilberto Zavala Pérez
Centro de Radioastronomía y Astrofísica - UNAM, Campus Morelia
Apartado Postal 3-72, 58090
Morelia, Mich. México
g.zavala@astrosmo.unam.mx

1. Introducción

La **seguridad**, es uno de los rubros más importantes a considerar en las redes de computadoras hoy en día, por lo que se han desarrollado múltiples mecanismos enfocados a proteger la información y privacidad de las diversas organizaciones que cotidianamente hacen uso de los sistemas de cómputo.

Este documento describe la implementación de un sistema de seguridad en la red de cómputo del *Centro de Radioastronomía y Astrofísica (CRyA) UNAM campus Morelia*. Esta red es muy similar a las redes académicas que se tienen en diversas universidades y centros de investigación de nuestro país, y en otras partes del mundo.

El sistema está basado en un *firewall* o *cortafuegos*, el cual protege a la red local de internet. La forma en que trabaja este *firewall* es mediante el filtrado de paquetes entre las dos redes: internet y la red interna, protegiendo a todas las computadoras de accesos no autorizados o que puedan suponer alguna amenaza a la seguridad.

Se inicia con los **antecedentes** que propiciaron la implementación del sistema de seguridad en la red local del *CRyA*.

Posteriormente se hace un análisis de los los **requerimientos** que se deben cumplir para mantener operando el sistema exitosamente.

Mas adelante se muestra el **diseño** propuesto junto con las consideraciones de hardware y software.

Enseguida se describe el proceso de **implementación** y el establecimiento de las políticas se seguridad adoptadas.

En la siguiente sección se muestra la forma de **operación** del programa *IPFilter* y los comandos que se emplearán para establecer los accesos y bloqueos de paquetes.

Posteriormente viene la etapa de **programación** del sistema, ahi se muestran los archivos de configuración y la forma en que se pone en operación el sistema.

Mas adelante se tiene una etapa de **pruebas**, las cuales sirven para diagnosticar el funcionamiento del sistema y hacer las adecuaciones pertinentes.

Al final se dan algunas **recomendaciones** sobre como se puede prevenir posibles fallas y como actuar en caso de alguna **emergencia** que afecte al sistema.

2. Antecedentes

La red donde se instaló el *filrewall* está conformada por estaciones de trabajo *Sparc* de *Sun Microsystems* con sistema operativo *Solaris*, computadoras personales (PCs) con sistema operativo *Linux*/*Microsoft Windows* y equipos *Mac*.

Se tomó la decisión de implementar un sistema de seguridad a raíz de un incidente donde varias computadoras sufrieron ataques severos y ocasionaron fallas en la operación y de negación de servicios.

El *firewall* está operando en una estación de trabajo *Ultra Sprac 10* con Solaris 9 y el programa *IPFilter*. Este *software* permite hacer un bloqueo de paquetes mediante el ciertas reglas de acceso y filtrado, las cuales aislan lógicamente a las computadoras locales de Internet.

3. Requerimientos

Los requerimientos básicos que debe cubrir este sistema son principalmente el proveer un acceso seguro y confiable a las diversas aplicaciones que cotidianamente se usan en la red, tales como acceso a bases de datos, consulta de información en línea, transferencia de grandes archivos, correo electrónico, audio y video en línea y todas las facilidades que ofrece internet. Así como también el poder compartir y acceder a los diversos recursos y servicios de cómputo que se ofrecen en la red local en forma sencilla y segura.

Los siguientes puntos cubren los requerimientos estipulados:

- Protección general a las computadoras, sin acceso directo desde internet
- Acceso controlado a los servidores de uso público
- Transparencia para el acceso a internet desde la red local
- Facilidad para el acceso desde internet a servicios ofrecidos en la red local
- Facilidad de implementación
- Facilidad para actualizar o modificar las reglas e filtrado
- Robustez, confiabilidad y reduncancia del sistema
- Bajo costo

4. Diseño

En esta sección se describen las consideraciones más importantes en el diseño de *firewalls* y en general de sistemas de seguridad.

La primera es referente a la política de seguridad de la organización donde se instalará el *firewall*. La configuración y el nivel de seguridad potencial será distinto para una organización que le interese bloquear todo el tráfico del exterior hacia su red interna (abriendo únicamente algunos servicios como el *web* y el correo) de otra que le interese bloquear servicios de salida hacia internet para que los usuarios no se distraigan navegando en la red.

Una segunda decisión importante en la etapa de diseño es el nivel de monitorización, redundancia y control deseado. Una vez definida la política a seguir, hay que decidir como se va a implementar el *firewall*, indicando básicamente que se va a permitir y que se va a rechazar. Para esto existen dos aproximaciones generales: Una *restrictiva* y una *permisiva*, en la primera rechazamos todo lo que explicitamente no se permita y en la segunda permitimos todo lo que explicitamente no se permita. Evidentemente la primera es la más aceptable en cuanto a seguridad, sin embargo, no siempre es posible implementarla ya que existen factores no técnicos que estarían en desacuerdo con esta decisión.

Finalmente una tercera consideración es la económica. Esto generalmente está en función de lo que queremos proteger. Un *firewall* puede implicar desde un fuerte desembolso para la organización, hasta prácticamente nada cuando se emplea algún sistema basado en *software libre*, salvo el tiempo empleado por los administradores del sistema.

Una vez que se han tomado las tres consideraciones anteriores, es necesario recordar un principio básico: *mínima complejidad* y *máxima seguridad*. Cuanto más simple sea el sistema, menos servicios ofrezca, más fácil será su mantenimiento y por tanto mayor su seguridad.

Con lo anterior en mente, estamos listos para plantear el tipo de *firewall*, las políticas y su ubicación física dentro de la red de cómputo.

Respecto de las políticas se seguridad, se requiere que el acceso de internet hacia la red local esté controlado, únicamente se darán acceso a servicios bien identificados, esto es, se adoptará una política restrictiva. Para el acceso de la red local hacia internet no se tendrán restricciones, salvo las que se observen necesarias, esta parte tendrá una política permisiva.

Respecto del esquema de red con que se cuenta actualmente, lo más conveniente y fácil de implemetar es un *firewall* a nivel lógico que haga el filtrado de paquetes. Para ello, es necesario el uso del sistema *NAT* (*Network Address Translation* - Traducciones de Direcciones de Red). Bajo este sistema, es necesario asignar direcciones IP no homologadas a todas las computadoras que estarán detrás del *firewall*. En la figura 1 se muestra un esquema de interconexión del *firewall* en la red local.

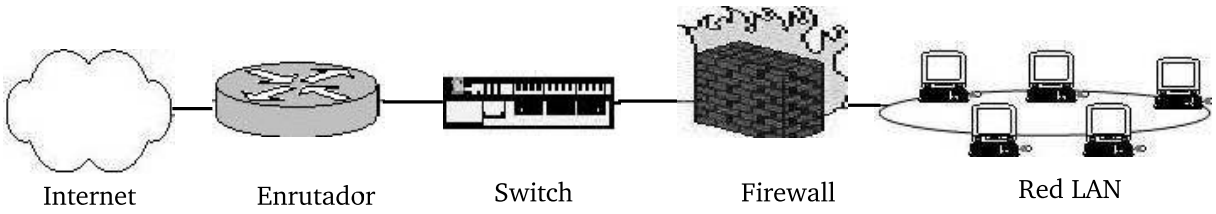

Figura 1: Interconexión del *firewall* en la red local.

4.1. Consideraciones de *hardware*

Debido a que la función que realiza el *firewall* no demanda demasiado poder de procesamiento ni capacidad de almacenamiento, se optó por una computadora *UltraSparc 10* a 400 MHz con 256 Mb de *RAM*, 2 discos duros de 9 Gb y 2 tarjetas de red 10/100 Mbps. Estas computadoras han tenido muy buen desempeño y robustez, lo que las hace una excelente opción para este trabajo.

4.2. Consideraciones de *software*

El *firewall* está basado en *software*, por lo que resulta de gran importancia evaluar y elegir el mas adecuado para los requerimientos establecidos. Se revisaron varios programas dedicados al filtrado de paquetes, entre los programas evaluados están el *Iptables*, el *SunScreen* de *Sun Microsystems*, y el *IPFilter*. De este útimo se obtuvo el código fuente y la documentación necesaria para compilarlo, instalarlo y configurarlo. La compilación se hizo con *GCC* para *Solaris 9* y se generó un paquete en formato *pkg*, listo para ser instalado en la computadora *firewall*.

5. Implementación

5.1. Proceso de instalación

El proceso de instalación del *firewall* inicia con la carga del sistema operativo a la computadora. La documentación recomienda fuertemente que se haga una instalación mínima, esto con la finalidad de que se instale la menor cantidad de programas, así también para que disminuyan las posibilidades de falla debido a huecos de seguridad en el sistema operativo. En este caso se instaló únicamente el sistema *core* del *Solaris 9*. Esta instalación incluye lo mínimo necesario para que funcione la computadora, y no es posible compilar programas ni correr aplicaciones gráficas.

Por lo general las computadoras *Sparc* vienen con una sola interfaz de red, por lo que fue necesario agregarle una segunda tarjeta de red. También se hizo una modificación en la memoria *nvram* de la computadora para que reporte direcciones *MAC* (dirección física) diferentes para cada tarjeta, se modificó la variable *local-mac-address?=true*. Otras modificaciones en la *nvram* fueron hechas para dar cierto respaldo y redundancia en el funcionamiento de esta computadora. Se agregó un segundo disco de arranque (*boot*), para que en caso de que falle el principal se pueda iniciar del secundario. Se modificó la variable *boot-device=disk disk2 net*, para en caso de emergencia se pueda arrancar del disco 2.

5.2. Adecuaciones hechas al sistema operativo

Las adecuaciones consisten básicamente en poner a punto el sistema operativo, cuidando de algunos detalles importantes. Uno de ellos es la instalación de los parches al sistema operativo, ya que la falta de ellos representa un gran hueco de seguridad para el sistema. Estos parches se obtienen del sitio oficial de *Sun: sunsolve.sun.com* en el directorio /pub/patches. Posteriormente se descomprime el archivo con el programa *unzip* y se instalan con el *script* install_cluster.

Otro punto relacionado con la seguridad está en el programa *inetd*, el cual se recomienda desactivarlo. El archivo de arranque es (S72inetsvc), por lo que se puede borrar o cambiar los permisos para que no se ejecute al arrancar la computadora. También se recomienda revisar los demás servicios del directorio /etc/rc2.d, quitar todos los servicios que no se requieran y solo dejar activos los que sabemos que si se necesitan.

Modificar el archivo /etc/system para proteger de posibles *buffer overflow* (desbordamiento de memoria)

```
set noexec_user_stack=1
set noexec_user_stack_log=1
```

Modificar el archivo /etc/init.d/inetinit

```
ndd -set /dev/ip ip_respond_to_echo_broadcast 0
ndd -set /dev/ip ip_forward_directed_broadcast 0
ndd -set /dev/ip ip_respond_to_timestamp 0
ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
ndd -set /dev/ip ip_forward_src_routed 0
ndd -set /dev/ip ip_ignore_redirect 1
```

Habilitar el *ip_forwarding*, esto es requerido para que el sistema haga el reenvío de los paquetes entre ambas redes. El comando para habilitarlo es *ndd -set /dev/tcp ip_forwarding 1*.

Para activarlo al arranque de la computadora se pone en el *script* /etc/init.d/ipforward. *Ipforwarding* debe ser arrancado después de *ipf* /etc/rc2.d/S65firewall e *inet* /etc/rc2.d/S69inet en /etc/rc2.d/S70ipforward.
Configurar el sistema de bitácora en el archivo /etc/syslog.conf

```
#---------------------------------syslog.conf---------------------------------
# IP Filter
# Log to local
local0.info;local0.err;local0.debug     /var/log/ipflog
#
# Log to a dedicate syslog server
local0.info;local0.err;local0.debug ifdef('LOGHOST',/var/log/ipflog, @loghost)
#-----------------------------------------------------------------------------
```

Una vez que la computadora está funcionando se configuran las tarjetas de red, una interfaz hacia la red local y la otra hacia internet, en este caso hacia el enruteador físico (*Cisco Router*). Se verifica que la computadora tenga acceso hacia las 2 redes, la pública y la privada. Posteriormente instalar el *software* que va a hacer el filtrado de paquetes: el *IPFilter*.
El programa *IPFilter* instalado es la versión *3.4.33pre2* y se obtuvo del sitio:
http://coombs.anu.edu.au/IPFilter/.

Este programa se descomprime con el comando *gunzip* y se extraen los archivos con el comando *tar: gunzip ipfilter-version.tar.gz ; tar xvf ipfilter-version.tar*.

Después posicionarse en el directorio donde se extrajo el programa y se ejecuta el comando: *make solaris* (La documentación recomienda no usar el *make de* GNU, usar el /usr/ccs/bin/make. Una vez que terminó de compilarse el *ipfilter* hay que cambiarse al directorio *SunOS5*, ejecutar el comando *make package*. Este último crea el paquete en formato para *Solaris*, listo para instalarse en cualquier computadora *Sparc* con la misma versión del sistema operativo. Este paquete se transfiere a la computadora *firewall* y se agrega al sistema con el comando *pkgadd -d ipf.pkg*, responder a las preguntas y dejar que se instale el nuevo paquete. Se verifica que se haya instalado mediante el comando *pkginfo -l ipf*:

```
   PKGINST:  ipf
      NAME:  IP Filter
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  3.4.33pre2
    VENDOR:  Darren Reed
      DESC:  This package contains tools for building a firewall
  INSTDATE:  Jan 05 2004 19:45
     EMAIL:  darrenr@pobox.com
    STATUS:  completely installed
     FILES:       92 installed pathnames
                  13 shared pathnames
                   6 linked files
```

```
  26 directories
  11 executables
2061 blocks used (approx)
```

Por *default* el paquete se instala en el directorio /opt/ipf/ y la configuración se hace en los archivos /etc/opt/ipf/ipf.conf y /etc/opt/ipf/nat.conf. Adicionalmente hay que configurar un archivo de arranque del firewall en /etc/init.d/firewall.

5.3. Establecimiento de las políticas de acceso

Para esta red, es requerimiento mantener cerrados todos los accesos desde el exterior y únicamente se dejan abiertos algunos puertos específicos para ciertos servicios como el correo electrónico, el *web*, el *ssh*, el *ftp* y otros.

El acceso de la red local al exterior de momento no está restringido, pero se ha considerado aplicar las mismas políticas que en la interfaz externa, solo agregando acceso a algunos servicios adicionales como el *DNS, real audio, IRC* y otros. Es más seguro aplicar una política restrictiva y únicamente abrir los que se requieran, que ir cerrandolos conforme nos dan problema o se vuelven un riesgo de seguridad. Como una medida adicional de seguridad se mueven los servidores de acceso público detrás del *firewall*, tanto el correo, el *web*, así como el que da acceso interactivo mediante *secure shell* (ssh). Quedando pendiente únicamente el servidor de *ftp anónimo*.

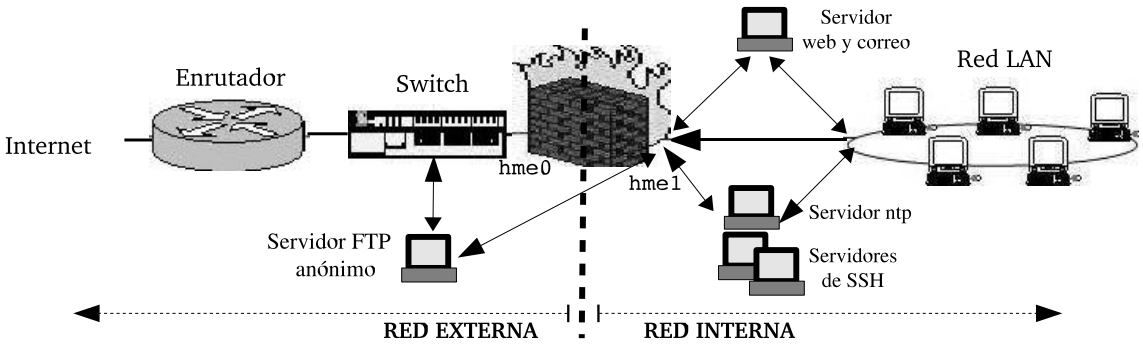

Figura 2: Esquema general de interconexión y servicios disponibles.

Interfaces de red en el firewall: *hme0 y hme1*

- hme0 interfaz externa, IP 132.248.81.bbb

  Por *default* está cerrado todo el acceso externo hacia hme0, lo único permitido son los servicios listados en la tabla 1. También por *default* está permitido todo el acceso de la red interna hacia internet.

- hme1 interfaz interna, IP 192.168.2.bbb

  Únicamente está permitido el acceso desde 132.248.81.bbb dirigido a la dirección 192.168.2.0, lo demás está bloqueado. Así mismo únicamente está permitido el acceso desde 192.168.2.0 y dirigido a 132.248.81.bbb , lo demás está bloqueado.

Tabla 1: Servicios disponibles a través del firewall.

servicio	**alias**	**IP externa**	**servidor interno**	**puerto**
correo	mail.astrosmo.unam.mx	132.248.81.bbb	192.168.2.ccc	25
web	www.astrosmo.unam.mx	132.248.81.bbb	192.168.2.ccc	80,443
ntp	ntp.astrosmo.unam.mx	132.248.81.bbb	192.168.2.fff	123
imap	mail.astrosmo.unam.mx	132.248.81.bbb	192.168.2.ccc	143*,993
pop3	mail.astrosmo.unam.mx	132.248.81.bbb	192.168.2.ccc	110*,995
ssh	ssh.astrosmo.unam.mx	132.248.81.bbb	192.168.2.[ddd,eee]	22
ftp actual	ftp.astrosmo.unam.mx	132.248.81.aaa	132.248.81.aaa	20, 21
ftp futuro	ftp.astrosmo.unam.mx	132.248.81.bbb	192.168.2.aaa	20,21

* Estos puertos solo están disponibles desde la red interna.

6. Operación del *IPFilter*

IPFilter es un programa que maneja el filtrado de paquetes, lo cual significa que permite o rechaza el flujo de paquetes entre 2 redes de acuerdo a las reglas predefinidas. Maneja además la traducción de direcciones IP (*Network Address Translation - NAT*), lo cual es usado para direcciones no homologadas (192.168.) y así salir a internet a través de una sola dirección IP. Otra función es la referente al manejo de balanceo de carga a través de un *Pool* de direcciones. Así mismo es fácil implementar un servidor *Proxy* transparente mediante el *IPFilter*. Todas estas características, además de su facilidad en la administración hacen que este programa sea bastante empleado en los *firewall* en sistemas *Unix* (*HP-UX, IRIX, FreeBSD, NetBSD, Solaris*, etc).

6.1. Configuración de las reglas de filtrado de paquetes

La configuración del *IPFilter* se hace en 2 archivos, el ipf.conf y el nat.conf.
Las reglas en *IPFilter* llevan la siguiente sintaxis:

```
Accion [in | out] opciones
```

Las acciones y opciones más comunmente usadas en el ipfilter se muestran en las tablas 2 y 3.

Tabla 2: Acciones comunes del IPFilter.

Acción	**Función**
block	Evita que el paquete pase a través del filtro
pass	Permite que el paquete pase a través del filtro
log	Registra en bitácora el paquete, ya sea que pase o no a través del filtro
in/out	Se refiere a si un paquete entra o sale

Ejemplo:

```
block in log quick from 192.168.0.0/16 to any group 100
```

Tabla 3: Opciones comunes del IPFilter.

Opción	**Función**
quick	Se ejecuta la regla inmediatamente, sin consultar las demás reglas
on *interfaz*	Aplica la regla a alguna interfaz específica
proto	Se refiere a un protocolo en particular
from/to/all/any	Se aplica para la dirección fuente, la dirección destino y el puerto
head *número*	Crea un nuevo grupo de reglas
group *número*	Agrega la regla al grupo número X en lugar del grupo por default (0)

6.2. Configuración de las reglas del NAT

El *NAT* establece reglas de *mapeo* que traducen direcciones IP fuente y destino a otras direcciones *homologadas o no-homologadas*, también puede redireccionar tráfico de un puerto a otro, manteniendo la integridad del paquete.
El *NAT* usa la siguiente sintaxis:

```
comando interfaz parametros
```

Las acciones y opciones más comunmente usadas en el nat se muestran en las tablas 4 y 5.

Tabla 4: Acciones comunes del NAT.

Comando	**Función**
map	Mapea una dirección o red a otra dirección IP o red
rdr	Redirecciona los paquetes de una dirección IP-Puerto a otra dirección IP-Puerto
hme0	Es el nombre de la interfaz, puede variar

Tabla 5: Opciones comunes del NAT.

Opción	**Función**
ipmask	Designa la máscara de red
dstipmask	Designa la dirección a la que ipmask es trasladada
mapport	Designa los protocolos rcp, udp o tcp/ucp, asi como un intervalo de puertos

Ejemplo:

```
map hme0  192.168.2.0/24 -> 0/32  portmap tcp/udp auto
```

7. Programación del sistema

7.1. Archivos de configuración

La forma mas sencilla de establecer las reglas de filtrado y redireccionamiento en *IPFilter* (y otros firewalls) es a través de archivos de configuración. Los archivos de configuración del *IPFilter* están en /etc/opt/ipf/ipf.conf y /etc/opt/ipf/nat.conf. Los *scripts* de arranque están en /etc/init.d/firewall y /etc/init.d/ipforward.

7.1.1. ipf.conf

En el archivo ipf.conf se establecen las reglas para acceso o filtrado de los paquetes que llegan al *Firewall*. Para activar las reglas se usa el comando *ipf -f ipf.conf*, esto está incluido en el *script* de inicio.

A continuación se muestra parte del archivo ipf.conf:

```
#----------------------------------ipf.conf-----------------------------------
# Se identifican las 2 interfaces de red
# hme0 -  interfaz externa
# hme1 -  interfaz interna
# Se agrupa para darle mayor rapidez al filtrado
# grupo 100 entrante en  hme0
# grupo 150 saliente en  hme0
# grupo 200 entrante en  hme1
# grupo 250 saliente en  hme1
#-----------------------------------------------------------------------------
#interfaz hme0
#-----------------------------------------------------------------------------
# groupo 100
# Previene el  "localhost spoofing "
block in log quick from 127.0.0.1/32 to 192.168.0.0/24 group 100
block in log quick from any to 127.0.0.1/8 group 100
#Inicia apertura de puertos...
#Se deben agregar las entradas correspondientes en el archivo nat.conf para que
#opere correctamente el acceso.
#
#puerto abierto para el ssh.
# Permite el acceso a todos los paquetes provenientes de cualquier direccion
# dirigidos hacia las computadoras internas con IP  ddd y eee
pass in log quick proto tcp/udp from any to 192.168.2.ddd port = 22 keep \\
state group 100
pass in log quick proto tcp/udp from any to 192.168.2.eee port = 22 keep \\
state group 100
# Puertos abiertos para el correo y web
# Permite el acceso a todos los paquetes provenientes de cualquier direccion
# dirigidos hacia la computadora con IP ccc.
pass in log quick proto tcp/udp from any to 192.168.2.ccc port = 25 keep \\
state group 100
pass in log quick proto tcp/udp from any to 192.168.2.ccc port = 80 keep \\
state group 100
# Puerto abierto para el "ping"
# Permite el acceso desde cualquier fuente y para cualquier destino
pass in log quick proto icmp from any to any keep state keep frags group 100
# Abrir los demas puertos que se requieran...
# Permite pasar todos los paquetes provenientes de la red interna hacia internet
pass in log quick proto tcp/udp from 192.168.2.0/24 to 132.248.81.bbb
# Bloquea todo lo que no esta explicitamente permitido
block in log proto tcp/udp from any to 132.248.81.bbb
```

```
#------------------------------------------------------------------------
# grupo 150
# Permite que salgan paquetes icmp hacia internet (ping)
pass out log quick proto icmp from any to any keep state keep frags group 150
# Permite que salgan todos los paquetes tcp/udp hacia internet
pass out log quick proto tcp/udp from any to any keep state keep frags group 150
#------------------------------------------------------------------------
# interfaz hme1
#------------------------------------------------------------------------
# grupo 200
# Permite pasar los paquetes de la red interna hacia cualquier direccion
pass in log quick from 192.168.2.0/24 to any group 200
# Permiten pasar los paquetes de la interfaz externa hacia la red interna
pass in log quick from 132.248.81.bbb to 192.168.2.0/24 group 200
#------------------------------------------------------------------------
# groupo 250
# Permite pasar los paquetes de la red interna hacia a la interfaz externa
pass out log quick from 192.168.2.0/24 to 132.248.81.bbb group 250
# Permite pasar los paquetes de la interfaz externa hacia la red interna
pass out log quick from 132.248.81.bbb to 192.168.2.0/24 group 250
#------------------------------------------------------------------------
```

7.1.2. nat.conf

En el archivo nat.conf se establecen las reglas para hacer la traducción de direcciones de red y el redireccionamiento de direcciones IP-Puerto. Este es su contenido:

```
#---------------------------------nat.conf-------------------------------------
# Se hace el NAT para que las computadoras de la red interna salgan con una
# unica direccin (la de la intefaz externa). El manejo de puertos es automatico
map hme0  192.168.2.0/24 -> 0/32  portmap tcp/udp auto
map hme0  192.168.2.0/24 -> 0/32
#
# Se hace el redireccionamiento de los servicios abiertos en el archivo ipf.conf
# Se redirecciona el ssh de la interfaz externa hacia las computadoras internas
# con IP ddd y eee. Observese la configuracin redundante.
rdr hme0 0.0.0.0/0 port 22 -> 192.168.2.ddd port 22 tcp round-robin
rdr hme0 0.0.0.0/0 port 22 -> 192.168.2.eee port 22 tcp round-robin
# Se redirecciona el correo electrnico a la computadora con IP ccc
rdr hme0 0.0.0.0/0 port 25 -> 192.168.2.ccc port 25
# Se redirecciona el servicio de web y web seguro a la computadora con IP ccc
rdr hme0 132.248.81.bbb/32 port 80 -> 192.168.2.ccc port 80
rdr hme0 132.248.81.bbb/32 port 443 -> 192.168.2.ccc port 443
# Se redireccionan los servicios pop e imap seguros a la computadora con IP ccc
rdr hme0 132.248.81.bbb/32 port 993 -> 192.168.2.ccc port 993
rdr hme0 132.248.81.bbb/32 port 995 -> 192.168.2.ccc port 995
...
#------------------------------------------------------------------------
```

7.1.3. firewall

El archivo firewall es el que activa el *ipf* e *ipnat* y se ejecuta siempre que se inicia la computadora, también se puede ejecutar al momento de hacer algún cambio en los archivos de configuración y se quiera actualizar las reglas.

```
#----------------------------firewall----------------------------------------
#!/bin/sh
cd /etc/opt/ipf || exit 1
log() {
 test -x "$LOGGER" && $LOGGER -p info "$1"
}
add_addr() {
 addr=$1
 nm=$2
 dev=$3
 ( ifconfig $dev | egrep  "inet +${addr} " ) ||
   {
     echo "$dev: $addr"
     ifconfig $dev $addr alias
   }
}
IPF="/usr/sbin/ipf"
IPNAT="/usr/sbin/ipnat"
LOGGER="/usr/bin/logger"
ndd -set /dev/ip ip_forwarding 1
add_addr 132.248.81.bbb 255.255.255.0 hme0
add_addr 192.168.2.bbb 255.255.255.0 hme1
add_addr 127.0.0.1 255.0.0.0 lo0
log "Activating firewall script generated Wed Feb 11 13:19:26 2004 CST by root"
$IPF -Fa
$IPNAT -C
$IPF  -I -f ipf.conf
$IPNAT  -f nat.conf
$IPF  -s
/sbin/kldstat -n ipl.ko > /dev/null 2>&1 || $IPF -E
#----------------------------------------------------------------------------
```

7.1.4. ipforward

Este script activa a desactiva el IPForwarding para que nuestro firewall pueda redireccionar los paquetes que recibe y entregarlos a las computadoras detrás de el. Este programa se ejecuta al iniciar la computadora ya que es indispensable para que funcione el firewall.

```
#---------------------------ipforward----------------------------------------
#!/bin/sh
case "$1" in
           start)
           echo "Activating IP Forwarding..."
```

```
            /usr/sbin/ndd -set /dev/tcp ip_forwarding 1
                 ;;
            stop)
            echo "De-activating IP Forwarding..."
            /usr/sbin/ndd -set /dev/tcp ip_forwarding 0
                  ;;
                  *)
            echo "Usage: $0 (start|stop)" >&2
            exit 1
                  ;;
                  esac
exit 0
#-------------------------------------------------------------------------
```

8. Pruebas

8.1. Pruebas iniciales de funcionamiento

Una vez que se ha instalado y configurado todo, se procede a hacer algunas pruebas sencillas para verificar que el sistema opere bien. Un punto muy importante a considerar verificar que el *ipf* esté cargado en el sistema y que realice el *IP forwarding*.
Si se requiere instalar el servidor *web* detrás del *firewall* se tienen 2 opciones:

Una opción es instalar un servidor de *DNS (Domain Name Server - Servidor de Nombres de Dominio)* interno, ahi se da de alta dirección IP interna del servidor con el nombre válido de internet, además se requiere configurar todos los clientes para que usen por default este servidor de *DNS*, de esta forma se pueden redireccionar adecuadamente al servidor *web*.

La otra opción consiste en instalar un programa llamado *rinetd*, este tiene la función de redireccionar las peticiones IP que se hagan al servidor *web* a una computadora que este detrás del *firewall*. Aparte del *web* también se pueden redireccionar los servicios de correo,*ssh, imap y pop*. El *rinetd* está disponible en *http://www.boutell.com/rinetd/*, sin embargo se requiere una versión modificada para *Solaris*, esta puede ser obtenida de *http://duksta.org/code/rinetd/rinetd-0.61-with-Solaris-patch.tar.gz*. La compilación e instalación es muy sencilla, una vez compilado se copia el ejecutable en /usr/sbin/rinetd o en algún otro directorio y se ejecuta directamente sin parámetros. Por *default* usa el archivo de configuración /etc/rinetd.conf. El contenido de este archivo es el siguiente:

```
---------------rinetd.conf-------------------
1  132.248.81.bbb 22 192.168.2.ddd 22
2  132.248.81.bbb 25 192.168.2.ccc 25
3  132.248.81.bbb 80 192.168.2.ccc 80
4  132.248.81.bbb 443 192.168.2.ccc 443
5  132.248.81.bbb 993 192.168.2.ccc 993
6  132.248.81.bbb 995 192.168.2.ccc 995
7  logcommon
8  logfile /var/log/rinetd.log
-------------------------------------------
```

La línea **1** indica que cuando se reciba una petición de *ssh* en la interfaz externa (puerto 22), se redirecciona a una computadora interna, también al puerto 22.

De la línea **2** a la **6** se reciben peticiones de correo, web, web seguro, imap seguro y pop seguro respectivamente, estas peticiones se redireccionan a una misma computadora de la red interna, usando los puertos por *default*.

Las líneas **7** y **8** indican que registe la bitácora de los accesos a estos servicios.

Para asegurarse de que arranque al iniciar la computadora se agrega un *script* de inicio en /etc/rc3.d/S90rinetd.

8.2. Pruebas periódicas

Algunas de las pruebas que se hacen en forma periódica consisten en verificar las políticas de acceso, esto se hace mediante el barrido de puertos desde una computadora externa. El programa que se ha usado es el *nmap*, el cual es un excelente *software* para el *escaneo* de puertos. También hay servidores que ayudan a revisar el funcionamiento del *firewall*, uno de ellos esta en *http://scan.sygate.com*. Otras pruebas consisten en verificar internamente que el *ipfilter* este haciendo bien su función de filtrado y de mapeo de direcciones (*NAT*). Para ver las reglas de filtrado de paquetes que están activas en el *kernel* se usa el comando *ipfstat -io*.

```
pass out log quick on lo0 from any to any
pass out log quick proto icmp from any to any keep state keep frags group 150
pass out log quick proto tcp/udp from any to any keep state keep frags group 150
pass in log quick proto tcp/udp from any to 192.168.2.ddd/32 port = 22 keep
state group 100
pass in log quick proto tcp/udp from any to 192.168.2.eee/32 port = 22 keep
state group 100
pass in log quick proto tcp/udp from any to 192.168.2.ccc/32 port = 25 keep
state group 100
pass in log quick proto tcp/udp from any to 192.168.2.ccc/32 port = 80 keep
state group 100
pass in log quick proto tcp/udp from any to 192.168.2.ccc/32 port = 443 keep
state group 100
pass in log quick proto tcp/udp from 192.168.2.0/24 to 132.248.81.bbb/32
block in log proto tcp/udp from any to 132.248.81.bbb/32
...
```

Como se observa, la salida de este comando es muy similar a las reglas establecidas en el archivo ipf.conf

Para ver las reglas del *NAT* activas en el *kernel* se usa el comando *ipnat -l*

```
List of active MAP/Redirect filters:
map hme0 192.168.2.0/24 -> 0.0.0.0/32 portmap tcp/udp auto
map hme0 192.168.2.0/24 -> 0.0.0.0/32
rdr hme0 0.0.0.0/0 port 22 -> 192.168.2.ddd port 22 tcp round-robin
rdr hme0 0.0.0.0/0 port 22 -> 192.168.2.eee port 22 tcp round-robin
rdr hme0 0.0.0.0/0 port 25 -> 192.168.2.ccc port 25 tcp
```

```
rdr hme0 132.248.81.bbb/32 port 80 ->  192.168.2.ccc port 80 tcp
rdr hme0 132.248.81.bbb/32 port 443 -> 192.168.2.ccc port 443 tcp
rdr hme0 132.248.81.bbb/32 port 993 -> 192.168.2.ccc port 993 tcp
rdr hme0 132.248.81.bbb/32 port 995 -> 192.168.2.ccc port 995 tcp

List of active sessions:
RDR 192.168.2.ccc   25    <- -> 132.248.81.bbb   25    [132.248.8.178 3307]
MAP 192.168.2.ccc   41544 <- -> 132.248.81.bbb   27700 [130.57.169.17 113]
MAP 192.168.2.fff   22813 <- -> 132.248.81.bbb   60881 [204.152.184.203 53572]
MAP 192.168.2.ccc   41542 <- -> 132.248.81.bbb   27698 [216.144.33.42 113]
MAP 192.168.2.ggg   3150  <- -> 132.248.81.bbb   29878 [132.248.64.250 53]
...
```

Nuevamente la salida del comando es similar a las reglas establecidas en el archivo nat.conf.

Las bitácoras las registra el programa *ipmon*, el comando para activarlo y guardar un registo en el archivo ipflog es *ipmon -a /var/log/ipflog*. La siguiente es una muestra de lo que se obtiene:

```
02/12/2004 13:00:13.233330 hme0 @150:5 p 65.54.252.230,25 -> 192.168.2.ccc,41853
 PR tcp len 20 52 -A K-S IN
02/12/2004 13:00:13.233381 hme1 @150:5 p 65.54.252.230,25 -> 192.168.2.ccc,41853
 PR tcp len 20 52 -A K-S OUT
02/12/2004 13:00:13.233850 hme1 @150:5 p 192.168.2.ccc,41853 -> 65.54.252.230,25
 PR tcp len 20 799 -AP K-S IN
02/12/2004 13:00:13.233894 hme0 @150:5 p 132.248.81.bbb,27505 -> 65.54.252.230,25
 PR tcp len 20 799 -AP K-S OUT
```

Lo que se obtiene con este comando es un estado en tiempo real del intercambio de paquetes, tanto de la red interna hacia internet como en el sentido inverso. Observese la dirección fuente, destino y los puertos involucrados.

8.3. Actualización de las reglas

Para actualizar las reglas existen básicamente 2 formas, una es agregarla en línea y la otra es la que se mostró agregándolas en los archivos de configuración ipf.conf y nat.conf. Se muestra la primera forma.

Para agregar una regla al programa de filtrado se usa el siguiente comando desde el *shell*:

```
echo "block in on hme1 proto tcp from 10.1.1.1/32 to any" | ipf -f -
```

Para agregar una regla al programa de *NAT* se usa el siguiente comando desde el *shell*:

```
echo "map hme1 192.168.2.0/24  132.248.81.bbb/32" | ipnat -f -
```

Para quitar las reglas del programa de filtrado se usa el comando:

```
ipf -Fa
```

Para quitar las reglas de *NAT* se usa el comando: ipnat -C

La opción de agregarla a través de un archivo es editando el archivo correspondiente y cargando el *ipf* e *ipnat* nuevamente. Lo mejor es mediante el *script* de inicio para que se borren las reglas actuales y se carguen las nuevas. Los comando que incluye el *script* son:

```
ipf -Fa (borra las relgas de filtrado)
ipnat -C (borra las reglas del NAT)
ipf -I -f /etc/opt/ipt/ipf.conf (lee las nuevas reglas de filtrado
del archivo ipf.conf)
IPNAT  -f /etc/opt/ipf/nat.conf (lee las nuevas reglas del NAT)
```

8.4. Actualización del *software*

Las actualizaciones del *software* consisten básicamente en instalar los *parches* al sistema operativo, los cuales se liberan aproximadamente cada semana. El sitio para obtener este paquete de parches es *http://sunsolve.sun.com*, ahi localizar el archivo correspondiente, para nuestro caso es el 9_Recommended.zip, descomprimirlo con el comando *unzip* y luego irse al directorio 9_Recommended, ahi revisar el archivo CLUSTER_README que describe el contenido del paquete de parches. Posteriormente usar el *script* install_cluster para hacer la instalación. Se tiene la intención de establecer un procedimiento para hacer estas actualizaciones en forma sistematizada.

9. Recomendaciones y emergencias

Debido a que el *firewall* es un punto de enlace muy crítico en la interconexión de la red local a internet, se ha previsto algunos planes de contingencia que podrían minimizar el impacto de una falla de este sistema.

Un punto muy importante en el diseño del *firewall* es la respuesta a emergencias, aqui podemos detectar 2 situaciones de falla. Una está en el *software*, a consecuencia de una mala configuración o algún hueco de seguridad en el sistema operativo lograran entrar al *firewall*, para este caso hemos previsto el uso de un pequeño *switch/firewall* que sustituya a la computadora *Sparc* en lo que se corrige la falla, esta posibilidad tratamos de minimizarla con la instalación de los parches al sistema operativo y revisión de las reglas de filtrado.

El otro punto de falla está directamente relacionado con el *hardware* de la computadora, para esta situación se tiene un disco de respaldo actualizado en la misma computadora, en caso de que fallara el disco principal se puede arrancar la computadora con el disco secundario, conservando toda la configuración del original. Este tipo de fallas una vez detectado tomaría muy pocos minutos corregirla, basta con rearancar con el disco 2, desde el *prompt*:

```
ok boot disk2
```

boot disk2 es el comando para que inicie del disco 2.

También se puede desconectar el disco primario y volver a encender la computadora, ya está configurado el sistema para que arranque automáticamente del disco 2 cuando no encuentra el disco 1.

Para el caso de que fallara algún otro componente o todo el sistema, se cuenta con una computadora preconfigurada de similares características en espera, la cual se adecuaría rápidamente para tomar el lugar del *firewall*. Este tipo de falla tomaría un poco mas de tiempo, pero no mas que unas cuantas horas.

10. Bibliografía

- **Seguridad**

  Chapman D. Brent, Cooper Simon, Zwicky Elizabeth D. Building Internet Firewalls, 2nd Edition, June 2000

  Antonio Villalón Huerta
  http://andercheran.aiind.upv.es/toni/personal/

  Seguridad en Unix y Redes, versión 2.1

  http://es.tldp.org/Manuales-LuCAS/SEGUNIX/unixsec-2.1-html/

- **IPTables**

  Rusty Russell Linux 2.4 Packet Filtering HOWTO, 2002/01/24
  http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html

- **SunScreen**

  http://wwws.sun.com/software/securenet

- **IPFilter**

  http://coombs.anu.edu.au/ipfilter

  Brendan Conoboy *synk@swcp.com*, Erik Fichtner *emf@obfuscation.org*

  IPF-HowTo *http://www.obfuscation.org/ipf/ipf-howto.txt*

  Phil Dibowitz *phil@ipom.com* IP Filter FAQ *http://www.phildev.net/ipf/*

  Documentación de *Sun* en línea

  http://docs.sun.com Solaris 10 System Administrator Collection, System Administration Guide: IP Services IP Security Solaris IP Filter

- **Rinetd**

  http://www.boutell.com/rinetd/

- **Solaris 9**

  http://www.sun.com/solaris