Seguridad en Redes de Computadoras Usando GNU/Linux

Luis Gerardo de la Fraga

Sección de Computación. Departamento de Ingeniería Eléctrica
CINVESTAV-IPN
Av. Instituto Politécnico Nacional 2508. 07300 México, D.F.
E-mail: fraga@cs.cinvestav.mx

Resumen

Se expondrán los mecanismos que hemos usado en la Sección de Computación del CINVESTAV para la seguridad y mantenimiento de nuestros laboratorios de cómputo: a través de cortafuegos y zonas militarizadas (redes con direcciones IP privadas), el arranque del sistema vía red (usando PXE ó Etherboot) y usando un sistema de archivos distribuido (NFS y AFS). Estos mecanismos nos han permitido optimizar el uso de los recursos en la red: facilita el mantenimiento de la misma red, el uso de discos duros y la implantación de políticas de respaldo.

Índice

1. Fundamentos de TCP/IP

1.1. La historia de Internet

Allá por 1960, los transistores reemplazaron los tubos de vacío reduciendo tanto el tamaño como el costo de las computadoras, haciéndolas, al mismo tiempo más poderosas. Pero Internet, como la mayoría de las demás creaciones humanas, es el resultado de la necesidad humana. Mientras tanto, la guerra fría alcanzó su temperatura más baja en 1962 (el año de la crisis de los misiles en Cuba) y una guerra nuclear parecía inminente. De este modo el Departamento de Defensa de E.E.U.U. se enfrentó a un problema: ¿cómo mantenerse comunicados en caso de un ataque nuclear por parte del enemigo? Así, surgió la necesidad de una red de comunicación que operara aún si la mayoría de sus enlaces y nodos fueran destruidos durante un ataque nuclear [1]. Fue entonces cuando un grupo de científicos trabajaron juntos para que las computadoras pudieran comunicarse. A principio de los años 60, la idea flotaba entre diversas instituciones americanas, como el Intituto de Tecnología de Massachussetts (MIT, por las siglas en inglés de Massachussetts Institute of Technology, una de las universidades más prestigiosas del mundo) y la corporación RAND. Paul Baran, un investigador de RAND concibió entonces la idea de una red distribuida, autónoma y capaz de recibir, transmitir y enrutar la información, que posteriormente se desarrolló en lo que hoy conocemos como Internet. En esta idea de red de comunicación, cada mensaje se quebraba en piezas de tamaño definido, y cada pieza sería transmitidos como un paquete individualmente direccionado. Estas piezas encontrarían su camino a través de la red hacia el destinatario, por cualquier ruta que estuviera accesible, brincando de un nodo a otro hasta llegar a su destino final. De este modo, si un nodo era destruido, los paquetes de información encontrarían su ruta alternativa en la red. Al llegar a su destino final, estas piezas de información se reensamblarían a su posición original, recuperando el mensaje que se quería mandar.

La idea de una red distribuida, así como la idea de "switcheo de paquetes" ó "conmutación de paquetes" (desmantelar cada mensaje y posteriormente ensamblarlo de nuevo para formar el mensaje original) se consideran como las aportaciones más importantes de Baran en el desarrollo de Internet. Uno de los colaboradores de ARPA, Leonard Kleinrock, entonces un estudiante de doctorado en el MIT conceptualizó la tecnología de "switcheo de paquetes" y publicó un *paper* [1] sobre ello en 1961. El Pentágono, a través de su Agencia de Proyectos de Investigación Avanzada (ARPA en sus siglas en inglés) financió la puesta en marcha de una prueba práctica. ARPA fue creada en respuesta al primer satélite artificial Sputnik, elaborado por la URSS y su objetivo consistía en mantener el liderazgo tecnológico estadounidense. En 1969, el año que el hombre llegó a la Luna, se abría el primer nodo de la red ARPANET, en la Universidad de California en Los Angeles [2].

El segundo nodo de la red ARPANET fue en el Intituto de Investigaciones de Stanford (IIS), donde trabajaba Douglas Engelbart en un proyecto sobre "ampliación del intelecto humano". Engelbart había inventado un poco antes el ratón, usado ahora por todas las computadoras, y se preocupaba por el trabajo en colaboración a través del hipertexto. No era un visionario aislado: en el MIT, J.C.R. Licklider ya discutía en 1962 su concepto de "Red Galáctica": un conjunto de computadoras interconectadas para dar acceso a almacenes de datos [2].

De modo que esta red empezó a servir para algo realmente revolucionario: para comunicar personas mas que computadoras. En 1969 apareció en la Universidad de California en Los Ángeles el sistema de RFC (Request for Commentaries: petición de comentarios), que permitía a todos los participantes en el proyecto opinar sobre las temas técnicos (aunque además de estos comentarios florecieron pronto discusiones sobre ciencia ficción). La cultura llegaba pronto al nuevo medio: en 1971 Michael Hart creó el Proyecto Gutenberg, para crear y difundir gratuitamente textos electrónicos (el estándar ASCII databa de 1968). En 1972 fecha de la demostración pública de la red3, apareció el primer programa de correo electrónico, que pronto se convirtió en una de las aplicaciones más usadas: tres años después ya se discutía el problema de cómo bloquear el "correo basura" (spam).

Mientras tanto, el primitivo proyecto ARPANET se preparaba para unirse con otras redes: satelitales (el primer satélite comercial se había lanzado en 1962), de radio terrestre, y de otros tipos, siempre y cuando compartieran la "conmutación de paquetes". Robert Kahn introdujo esta "arquitectura abierta" en 1972. Es en 1983 cuando se considera que nació realmente la Internet, al separarse la parte militar y la civil de la red. En ese momento ya la compartían 500 servidores (computadoras interconectadas). En el mismo año se

[1] Así se le llama a un artículo publicado en una revista internacional de prestigio

creó el sistema de nombres de dominios (.com, .edu, etc., más las siglas de los países), que prácticamente se ha mantenido hasta ahora. En 1984 William Gibson novelaba el nuevo mundo y acuñaba el término "ciberespacio". Al año siguiente se forjaba Well, la primera comunidad comercial de usuarios [2].

ARPANET desapareció como tal en 1989, pero muchas instituciones (de la NASA al Departamento de Energía) ya habían creado sus propias redes que podían comunicarse entre sí. El número de servidores en la red superaba los 100,000. Ese mismo año, Tim Berners-Lee, investigador en el centro europeo CERN de Suiza, elaboró su propuesta de un sistema de hipertexto compartido: era el primer esbozo de la WWW. Como el ARPANET veinte años atrás, su propósito era poner en comunicación a los científicos.

En 1992 con más de un millón de servidores en la red se creó la Internet Society, la "autoridad" de la red. Nacía como el lugar donde pactar los protocolos que harían posible la comunicación. Se trataba de una coordinación técnica, que no intervenía en los nacientes problemas de libre expresión: acababan de crearse la Fundación de Frontera Electrónica (Electronic Frontier Foundation), defensora de los "ciberderechos", y el más famoso sistema abierto de criptografía: Pretty Good Privacy [2].

Con la extensión de las computadoras personales y el lanzamiento del primer navegador de la WWW popular, Mosaic, en 1993, ya había llegado el momento de "surfear en la Web" (la expresión se registró por primera vez ese mismo año) Un chiste de Peter Steiner en New Yorker proclamaba: "En Internet, nadie sabe que eres un perro". En 1994 se abre el primer ciberbanco. En 1997 ya hay 17 millones de servidores en la red. A partir de aquí las estadísticas se nublan: el tremendo crecimiento de la red, unido a la autonomía de su funcionamiento, hacen que grandes zonas de sus contenidos estén en la penumbra: según datos de 1999 el conjunto de los grandes buscadores de páginas en la Malla Mundial sólo conoce el contenido de menos del 50 % de la red. La última iniciativa, Internet 2 [3], propone crear un espacio aparte y de más calidad en las comunicaciones para instituciones de investigación [2]

Un resumen gráfico del desarrollo de Internet se muestra en la tabla 1

Fecha	Suceso [1]
1957	La URSS lanzó Sputnik
1966	Experimentación con conmutación de paquetes en ARPA
1968	Primera Red de Switcheo de Paquetes
1969	Nació ARPANET
1972	Primera demostración pública de correo electrónico sobre ARPANET
1973	Kahn y Cerf presentaron un artículo sobre Internet en la Primera Conexión Internacional de Internet
1975	Se fundó Microsoft
1976	Se fundó Apple
1979	Se creó UseNet
1980	Comienzo de la fase experimental con TCP/IP
1981	Inclusión de nuevos nodos cada 20 días
1983	ARPANET emigró a TCP/IP. ARPANET se dividió en ARPANET y MILNET. Microsoft introduce Windows
1984	Internet excedió 1,000 huéspedes. William Gibson escribió "Neuromancer". Se introdujó el Servidor de Nombres de Dominio (DNS)
1986	Creación de la parte troncal (backbone) de la NFSnet
1987	Internet excedió 10,000 huéspedes
1988	Un gusano atacó 6,000 de los 60,000 huéspedes de Internet
1989	Internet excedió 100,000 huéspedes
1990	ARPANET se desmanteló. Comenzó Archie [a]
1991	Se creó WAIS. Se creó el Gopher [b]. NSF descarta la prohibición comercial.
1992	Internet excede 1 millón de huéspedes. Se introduce el concepto "Web" creado por Tim Berners-Lee
1993	MOSAIC se desarrolló por Marc Andreesen. InterNIC se fundó por NSF
1995	Privatización de la parte troncal de Internet

Cuadro 1: Resumen del desarrollo de Internet

[a] Archie fue uno de los primeros servicios en Internet. Permitía la búsqueda de archivos en la red

[b] El gopher fue otro servicio de Internet que permitía la búsqueda de archivos

1.2. La ARPANET

La ARPANET original consistió de cuatro computadoras huésped, una en la Universidad de California en los Ángeles (UCLA), el Instituto de Investigaciones de Stanford, la Universidad de California en Santa Bárbara y la Universidad de Utah. Posteriormente, la red de ARPANET fue reemplazada por la NSFnet que

culminó en lo que hoy es Internet.

Esta pequeña red, usando el *Protocolo de Control de Red* (NCP, del inglés Network Control Protocol) proporcionó a sus usuarios la habilidad de entrar a un huésped remoto, imprimir en una impresora remota y transferir archivos. Ray Tomlinson, un ingeniero de la compañía BBN, creó en 1971 el primer programa de correo electrónico.

ARPANET funcionaba a partir del principio de "switcheo de paquetes" y estaba basada en un conjunto de pequeños computadores interconectados llamados procesadores de mensajes con interfaz (IMPs). Estos IMPs, son los precursores de los modernos dispositivos de enrutamiento [4].

En su segundo año de operatividad, sin embargo, algo extraño sucedió. Los usuarios de ARPANET habían convertido la red en una oficina de correos electrónica de alta velocidad subvencionada federalmente. La mayor parte del tráfico de ARPANET no era el proceso de datos a largas distancias. En vez de eso, lo que se movía por allí eran noticias y mensajes personales. Los investigadores estaban usando ARPANET para colaborar en proyectos e intercambiar notas sobre sus trabajos. La gente tenía sus propias cuentas personales en las computadoras de ARPANET y sus direcciones personales de correo electrónico. No es que sólo utilizaran ARPANET para la comunicación de persona a persona, pero había mucho entusiasmo por esta posibilidad – mucho más que por la computación a larga distancia.

Eso no pasó mucho antes del invento de las listas de distribución, una técnica de emisión de información por ARPANET mediante la cual un mismo mensaje se podía enviar automáticamente a una gran cantidad de subscriptores. Es interesante que una de las primeras listas de distribución masivas se llamara "Amantes de la Ciencia Ficción" (SF-LOVERS). Discutir sobre ciencia ficción en la red no tenía nada que ver con el trabajo y eso enfadaba a muchos administradores del sistema de ARPANET, pero eso no impediría que la cosa siguiera.

Durante los 70s, ARPANET creció. Su estructura descentralizada facilitó la expansión. Contrariamente a las redes estándar de las empresas, la red de ARPA se podía acomodar a diferentes tipos de computadoras. En tanto una máquina individual pudiese hablar el lenguaje de conmutación de paquetes de la nueva y anárquica red, su marca, contenidos e incluso su propietario eran irrelevantes [5]. El éxito de la ARPANET fue en si el catalizador para la investigación en redes dando como resultado un protocolo de emergencia: el TCP/IP, el cual se estableció firmemente en 1980. La naturaleza descentralizada de ARPANET y la disponibilidad sin costo de programas basados en TCP/IP permitió que ya en 1977, otro tipo de redes no necesariamente vinculadas al proyecto original, empezaran a conectarse. En 1983, el segmento militar de ARPANET decide separarse y formar su propia red que se conoció como MILNET. ARPANET completó su transición al TCP/IP en 1983, y en 1990 dejó de ser la espina dorsal de la red Internet [4].

1.3. Switcheo de paquetes

Una de las ideas brillantes de Baran [6] fue dividir los mensajes en "bloques de mensaje" antes de enviarlos a través de la red. Cada mensaje debería enviarse separadamente y reunirse para completar el mensaje cuando fueran recibidos en su destino. Un británico llamado Donald Davies de forma independiente divisó un sistema muy similar, pera le llamo a los bloques de mensage como "paquetes", un término que fue adoptado eventualmente en vez del término bloques de mensaje de Baran.

Los paquetes permitieron una vía muy eficiente para transmitir datos. Las comunicaciones de datos son por naturaleza erráticas. La información es enviada en ráfagas, no en forma continua. Las líneas tradicionales de comunicación emplean línea dedicadas; las redes telefónicas son un buen ejemplo: cuando se hace una llamada, una línea se dedica para esa llamada, nadie más puede usar esa línea mientras se está realizando la llamada. Si existe una pausa en la conversación, no se envían datos, pero la línea se mantiene en uso y no disponible. Esto representa un desperdicio en la capacidad de comunicación (la "capacidad de comunicación" se conoce como *ancho de banda*).

Baran visionó a red con nodos autónomos que podrían actuar como interruptores ruteando paquetes de un nodo a otro hasta su destino final. Los nodos se diseñarían para guardar y enviar rápidamente, esquema que le llamó "ruteo de papa caliente". Cuando un nodo recibe un paquete, el nodo lo almacena; entonces determina la mejor ruta para su destino, y lo envia al siguiente nodo en el camino (de la mejor ruta). Usando computadores digitales como nodos, el proceso de ruteo podría hacerse muy rápidamente permitiendo transmisiones en tiempo real. Las computadoreas podrían usar estadísticas, puestas al día de forma constante, de la red y cada uno de sus nodos para determinar la mejor ruta en cualquier momento. Si hubiese un problema con algún nodo (o si hubiese sido destruido) los paquetes podrían rutearse alrededor de él. Es método de actualizar contantemente la información de la red para el rureo se conoce también como *ruteo dinámico*.

En la ARPANET, varios años después, cuando Larry Roberts estaba comenzando a trabajar en la AR-

PANET, él había escuchado de las ideas de Baran. Roberts no diseñó una red para uso en tiempo de guerra, sino para facilitar la comunicación en los investigadores de la ARPA y para permitirles el uso eficientes de recursos remotos. Pero las idea de Baran afectaron a Roberts. Se adoptaron las ideas de Baran sobre una red distribuida y con un esquema de switcheo de paquetes, y Baran de convirtió en un consultante informal del projecto ARPANET.

1.4. TCP/IP, su historia

En 1974, justo cuatro años después del nacimiento de la ARPANET, Vinton Cref y Robert Kahn inventaron el *Protocolo de Control de Transmisión* (TCP, del inglés *Transmission Control Protocol*). En 1974 publicaron el proyecto sobre el TCP. TCP fue diseñado para ser independiente de cualquier computadora o red. De este modo, ninguna computadora o red es esencial, ya que TCP/IP se adapta perfectamente a cualquier software o hardware. Debido a esta increíble capacidad de adaptación a cualquier medio, TCP/IP fue implantado en la ARPANET solo tres años después de la publicación del primer artículo sobre éste, osea en 1977. En aquel entonces ARPANET funcionaba a base de NCP (Network Control Protocol). Con el tiempo, TCP/IP reemplazó a NCP, debido a que NCP no era capaz de manejar eficientemente el enorme tráfico que empezaba a producir la red [7]. Entre otras razones se encuentran la fácil mantenibilidad y el bajo costo de implantación de TCP/IP. Los ajustes posteriores determinaron la incorporación del apéndice IP, por Internet Protocol en 1978, convirtiendo TCP en lo que hoy conocemos como TCP/IP. Un tiempo después ARPANET ya no estaba sola. Entidades estatales y académicas estadounidenses y europeas se sumaban al mundo de las redes. De este modo, se tenía que escoger un solo protocolo de red para que todas las redes pudieran trabajar juntas. Se escogió TCP/IP, debido a sus cualidades que ya mencionamos arriba. De esta manera, TCP/IP proveía un puente tecnológico que unía las pequeñas redes alrededor del mundo [8].

En 1982, rendido ante la evidencia de la popularidad del protocolo y de la fuerza de las conexiones a la red, ARPA decidió desclasificar el TCP/IP y además dispuso que fuera de uso obligatorio para todas aquellas redes conectadas a ARPAnet [7]. Así, para cualquier computadora dentro de ARPANET ó conectada a ARPANET, era obligada a cambiarse al protocolo TCP/IP dentro de un plazo de unos cuantos meses. Debido a las facilidades de este protocolo, muchos de los usuarios lo hicieron con éxito. En el año 1983, los usuarios empezaron a llamar a ARPANET y sus afiliados como Internet, y en ese mismo año el cambio en el lenguaje se hace oficial. Fue entonces cuando nació la Internet [8].

1.5. Capas y protocolos de TCP/IP

TCP e IP manejan juntos el flujo de datos, tanto el que entra como el que sale, en una red. IP pone paquetes en la red de forma indiscriminada, TCP se encarga de asegurar que los paquetes estén allí. TCP es responsable de: el “apretón de manos” (handshaking), la administración de los paquetes, del control de flujo y del manejo y detección de errores. TCP/IP es el ambiente que maneja todas estas operaciones y coordinarlas con los huéspedes remotos. TCP/IP esta realizado en cuatro capas, como puede verse en la Fig. 1, en vez de las siete capas del modelo OSI [9]. A las capas de la Fig. 1 comúnmente se le llama *pila* de TCP/IP. El modelo de capas nos sirve para hacernos una idea conceptual de como trabaja TCP/IP y como funcionan los programas en red.

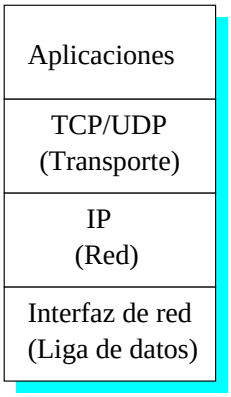

Figura 1: El modelo de capas de TCP/IP

La diferencia primaria entre el modelo de capas OSI y el de TCP/IP es que la capa de *transporte* no garantiza la entrega todas las veces. TCP/IP ofrece el *Protocolo de Datagramas para el Usuario* (UDP), que es un protocolo más simplificado, en el que todas la pila de TCP/IP realizan trabajos específicos o corren las aplicaciones.

La capa de *aplicación* consiste de SMTP, FTP, NFS, NIS, LPD, telnet y login remoto (rlogin). Todas estas aplicaciones son familiares a la mayoría de los usuarios de Internet.

La capa de *transporte* consiste de TCP y UDP, donde los paquetes son entregados sin casi un chequeo, y en la recepción se garantiza la entrega.

La capa de *red* está realizada con los siguientes protocolos: ICMP, IP, IGMP, RIP, OSPF y EGP para el ruteo de paquetes. No nos preocuparemos de ellos porque todos ellos son de bajo nivel y esotéricos.

La capa de *liga* consiste de RIP y RARP, los cuales manejan la transmisión de paquetes.

TCP, el Protocolo de Control de Transmisión, provee una entrega fiable del flujo y el servicio de conexión a las aplicaciones [10]. TCP usa *acuses de recibo* (en inglés, *acknowledgments*) y es capaz de retransmitir paquetes cuando sea necesario. TCP es un protocolo "orientado a conexiones" o "fiable". De forma simple, TCP tiene ciertas características que aseguran que los datos arriven al huésped remoto en una forma intacta. La operación básiza de esto yace en el "apretón de manos de tres vías" inicial de TCP, que será descrito en los siguientes tres pasos.

Paso 1 Se envía un paquete de sincronización (SYN) y el Número de Secuencia Inicial (ISN, Initial Sequence Number)

El huésped A desea establecer una conexión con el huésped B. El huésped A envia entonces un paquete solitario al huésped B con el bit de sincronización (SYN) puesto, que anuncia la nueva conexión, y un Número de Secuencia Inicial (ISN) el cual permitirá hacer el seguimiento de paquetes enviados entre los huéspedes:

Huésped A —— SYN(ISN) ⟶ Huésped B

Paso 2 Permite al huésped remoto responder con un acuse de recibo (ACK). El huésped B responde a la requisa enviando un paquete con el bit de sincronización (SYN) puesto y con el bit de acuse de recibo ACK puesto. Este paquete de regreso al huésped que realiza la llamada contiene no solamente el número de secuencia de la respuesta del cliente, sino también el Número Inicial de Secuencia más uno (ISN+1) para indicar que el paquete remoto ha sido correctamente recibido como parte del acuse de recibo y que está esperando para la siguiente transmisión:

Huésped A ← SYN(ISN+1)/ACK — Huésped B

Paso 3 Se completa la negociación enviando un acuse final al huésped remoto. En este punto el huésped A envía de regreso un paquete ACK final y el número de secuencia para indicar una recepción satisfactoria y que la conexión está completa y los datos pueden fluir ahora.

Huésped A —— ACK ⟶ Huésped B

El proceso entero de conexión sucede en milisegundos en ambos lados, independientemente acusando de recibo cada paquete a partir de este punto. Este apretón de manos asegura una conexión "fiable" entre los huéspedes y es por ello que TCP se considera un protocolo "orientado a conexión". Sólo los paquetes de TCP exhiben este proceso de negociación. Esto no sucede con los paquetes de UDP, los cuales se consideras "no fiables" y no intentan corregir los errores ni negociar una conexión antes del envio a un huésped remoto.

1.6. El encabezado de TCP

El encabezado que usa TCP se muestra en la Fig. 2. Los números representan las posiciones de los bits, de 0 a 32, por lo que cada renglón constra de 32 bits.

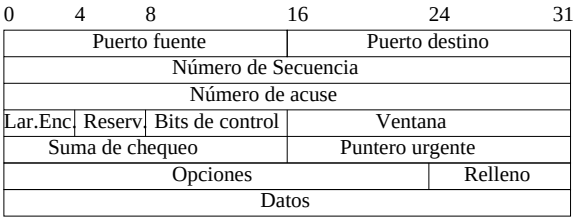

Figura 2: El encabezado de TCP

El *puerto fuente* y el *puerto destino* son dos números de 16 bits que indican el puerto fuente y destino, respectivamente.

El uso de campos de *número de secuencia* y de *número de acuse* ya fueron explicados en la subsección 1.5 y sirven para realizar el apretón de manos del protocolo.

La *compensación de fragmento* indica el final del encabezado, por lo tanto también el inicio de los datos.

El número *ventana* indica el número de octetos que manda el que envía. Comienza con el paquete en el campo *número de acuse*.

El campo *Lar.Enc.* indica el largo del encabezado y el campo *Reserv.* indica bits que están reservados y no se usan normalmente.

La información de los otros campos viene tratada en [9]

1.7. El encabezado de IP

El encabezado del protocolo IP es mostrado en la Fig. 3.

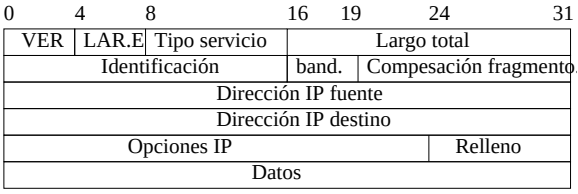

Figura 3: El encabezado de IP

El campo *VER* indica la *versión* del protocolo. Actualmente se usa la versión 4, experimentalmente se tiene la versión 6 que se usará en algunos años. El campo *LAR.E* indica el *largo del encabezado*. La máquina que recibe debe saber cuando debe de parar para leer el encabezado y cuando los datos comienzan. El *tipo de servicio* generalmente no se usa, este campo indica la prioridad del paquete, números más grandes indican una mayor prioridad. El *largo total* insica el largo del paquete en bytes, no puede ser mayor a 65,535 bytes o será tratado como un paquete corrupto por el receptor.

La información más importante para nuestas necesidades en este trabajo en la dirección IP, tanto fuente como destino.

La información de los otros campos viene tratada en [9]

2. Redes usando TCP/IP

En la figura 4 se muestra una red compuesta de cinco huéspedes, en este caso cada huésped es una computadora, interconectada con cable telefónico (conocido técnicamente como *par trenzado*. Vamos a aprender como configurar los huéspedes para permitir la interconexión. La red de la Fig. 4 está directamente conectada a Internet. Generalmente la conexión de intranets como la que se presenta en la Fig. 4 no es directa; es necesario usar un modem para realizar la conexión a través de un proveedor de Internet. La red de este ejemplo usa un concentrador (aunque podría ser también un etherswitch).

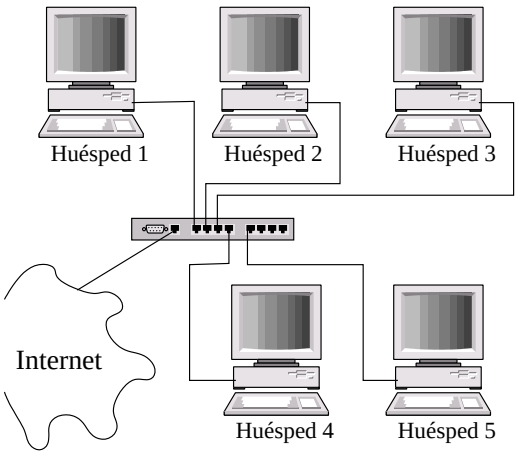

Figura 4: Esquema básico de una red

TCP/IP define una *interfaz* abstracta a través de la cual se puede accesar al hardware, esto a su vez es un mecanismo que oculta la diversidad de equipo que puede usarse en un ambiente de red [11]. Esta interfaz ofrece un conjunto de operaciones que es el mismo para todos los tipo de hardware y que trata básicamente con el envio y recepción de paquetes.

Por ejemplo, en el sistema GNU/Linux, las interfaces ethernet (las que ofrece una tarjeta de red) tienen nombres como *eth0* y *eth1*. Las interfaces PPP (que se usan para conectarse con un modem) tienen nombres como *ppp0* y *ppp1*.

Antes de usar una interfaz en una red, se debe de asignarle una dirección IP que sirve como su identificación cuando se comunican en el resto del mundo. Esta dirección es diferente del nombre de la interfaz mencionado en el párrafo anterior; si se compara la interfaz con una puerta, la dirección es como la placa de identificación pegada a ella.

Otros parámetros pueden ponerse para el dispositivo, tal como el tamaño máximo de los datagramas que pueden ser procesados por una pieza particular de hardware, el cual es conocido como la Unidad de Transferencia Máxima (Maximum Transfer Unit, MTU). Otros atributos se conocerán posteriormente. Afortunadamente, la mayoría de los atributos tienen valores por defecto funcionales.

2.1. Beneficios de usar una red TCP/IP

TCP/IP permite plataformas-entrelazadas o administración de redes heterogéneas. Por ejemplo una red de Windows NT podría contener una computadora de Unix o Macintosh o hasta redes mixtas. TCP/IP también tiene las siguientes características:

- Buena recuperación de las fallas
- Habilidad de añadir redes sin interrumpir los servicios ya existentes.
- Manejo de alto porcentaje de errores
- Independencia de la plataforma
- Bajos gastos indirectos de información.

Debido a que originalmente TCP/IP fue diseñado por propósitos relacionados al Departamento de Defensa de Estados Unidos, lo que ahora llamamos características fueron de hecho requisitos de diseño. La idea detrás de “Buena recuperación de las fallas” fue que si una parte de red fuera dañada durante un ataque, las piezas de red restantes deben seguir funcionando adecuadamente. Lo mismo aplica para, la capacidad de añadir nuevas redes, sin interrupción a los servicios ya existentes. La habilidad de manejar gran porcentaje de errores fue implantado para que si un paquete de información se pierde al recorrer una ruta, habría un mecanismo que asegurara que éste llegará a su destino mediante otra ruta. Independencia de plataforma

significa que las redes y los clientes pueden ser Windows, Unix, Macintosh o cualquier otra plataforma o combinación de ellas. La razón por la cual TCP/IP es tan eficiente son sus gastos indirectos bajos. Desempeño es la clave de cualquier red. TCP/IP no tiene una contraparte en su simplicidad y rapidez.

2.2. Números binarios y decimales

En base dos ó números binarios, el valor representado por "1" es determinado por su posición. No es diferente de la base diez que todos conocemos, en la cual el primer numero desde la derecha enumera unidades, el segundo desde la derecha enumera decenas, el tercero centenas, y así hasta el infinito. Mientras el sistema decimal provee 10 dígitos (de 0 a 9) para representar diferentes valores, el sistema binario solo ofrece dos dígitos validos: 0 y 1. Su posición, al igual que en el sistema decimal, determina el valor que representa. La posición que esta hasta la derecha, en términos decimales, representa 2. La siguiente posición a la izquierda 4, la siguiente 8, etc. Cada posición vale 2 veces más que su vecina derecha. El valor decimal de un numero binario se calcula sumando los valores decimales de los dígitos que tienen 1 en su posición. Matemáticamente, cada octeto de una dirección IPv4 (hay 4 de ellos) puede tener un valor máximo de 225 en el sistema decimal. Un número binario equivalente a 225 consiste de 8 bits, con todos los bits 1. Un ejemplo de la relación entre números decimales y binarios:

Dígito	8	7	6	5	4	3	2	1
Binario	1	1	1	1	1	1	1	1
Valor decimal del dígito	128	64	32	16	8	4	2	1

Como puede observarse, cada bit en la dirección binaria tiene "1" en su posición. Por esto, el valor decimal de este numero binario puede calcularse sumando los valores de las 8 columnas: $128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 225$.

La siguiente tabla muestra de la conversión de un número binario a uno decimal. En este ejemplo el quinto numero desde la derecha es 0. Esta posición representa el valor de 16. De esta manera, el valor de este numero binario es 16 unidades menor que 255: $128 + 64 + 32 + 8 + 4 + 2 + 1 = 239$.

Dígito	8	7	6	5	4	3	2	1
Binario	1	1	1	0	1	1	1	1
Valor decimal del dígito	128	64	32	16	8	4	2	1

Esta relación entre números binarios y decimales es la base de la arquitectura de direcciones IP. Recuerde que hay 4 octetos binarios en cada dirección IPv4, incluyendo subredes enmascaradas. Por eso es necesario entender la relación entre estos sistemas básicos, la conversión del uno al otro, antes de estudiar distintas maneras de implantar dirección de IP.

2.3. Direcciones IP

El protocolo de red IP entiende las direcciones como números de 32 bits. Esta convención es para la versión 4 (IPv4) que será tratada en todo el curso. A cada máquina debe asignársele un número único en el ambiente de la red. Existen algunos intervalos de números IP que se han reservado para usarse en el diseño de intranets (ó redes privadas). Estos intervalos están listados en la tabla 5. Sin embargo, para sitios de Internet, los números eran asignados hace ya algunos años, por una autoridad central, el *Centro de Información de la Red* (NIC, *Network Information Center*). Actualmente los números que se usarán son asignados por el mismo proveedor de Internet al que se le compra la conectividad IP.

Las direcciones IP se dividen por legibilidad en cuatro números de ocho bits, llamados *octetos*. Por ejemplo, luna.computacion.universidad.mx tiene la dirección 0x954C0C04, el cual se escribe como 149.76.12.4. Este formato se refiere frecuentemente como *notación decimal puntuada*. De esta forma cada byte es convertido en un número decimal (0-255), despreciando los ceros a la izquierda a menos que el número en sí sea cero.

Otra razón para esta notación es que una dirección IP se puede dividir en un número de *red*, la cual está contenida en los primeros octetos, y un número de *huésped*, que está en los restantes octetos. Cuando de requieren números IP y se les pide al NIC, este no asigna un número por cada huésped que se planea usar. En vez de ello se asigna un número de red y se permite asignar todas las direcciones IP válidas dentro del intervalo del número de huéspedes sobre su propia red, de acuerdo al diseño propio que se tenga. Al número de bits que comparten todas las direcciones de una red se le llama máscara de red (netmask), y su papel es determinar qué direcciones pertenecen a la red y cuáles no. Esto puede verse con el ejemplo mostrado en la tabla 2

Dirección IP	192.168.120.21
Máscara de red	255.255.255.0
Número de red	192.168.120.
Número de huésped	.21

Cuadro 2: Ejemplo de la división de una dirección IP

Cualquier dirección a la que se aplique una operación AND de bits con su máscara de red, revelará la dirección de la red a la que pertenece. La dirección de

red es por tanto siempre el menor número de dirección dentro del intervalo de la red y siempre tiene la porción de huésped codificada toda con ceros.

Por razones administrativas, durante el desarrollo inicial del protocolo IP se formaron, de forma arbitraria, algunos grupos de direcciones como redes, y estas redes se agruparon en las llamadas *clases*. Estas clases proporcionan un cierto número de redes de tamaño estándar que pueden ser reservadas. Los intervalos reservados pueden verse en la tabla 3.

Clase de red	Máscara de red	Direcciones de red
A	255.0.0.0	1.0.0.0 - 126.255.255.255
B	255.255.0.0	128.0.0.0 - 191.255.255.255
C	255.255.255.0	192.0.0.0 - 223.255.255.255
Multicast	240.0.0.0	224.0.0.0 - 239.255.255.255

Cuadro 3: Clases de direcciones IP

El número de huéspedes que permite cada clase es:

Clase A . La porción de red está contenida en el primer octeto. Esta clase provee una porción de huésped de 24 bits, permitiendo alrededor de 1.6 millones de huéspedes por red. Esta clase fue diseñada para redes extremadamente grandes.

Clase B . El número de red está en los primeros dos octetos. Esta clase permite 16,320 redes con 65,024 huéspedes cada una. Esta red fue diseñada para redes de tamaño moderado a grandes.

Clase C . El número de red está contenido en los primeros tres octetos. Esta clase permite cerca de dos millones de redes con 254 huéspedes cada una. Esta clase fue diseñada para permitir cientos de redes de tamaño pequeño.

En el ejemplo dado anteriormente, para la dirección IP 149.76.12.4, la dirección de **luna**, se refiere al huésped 12.4 de la red clase B 149.76.0.0.

No todos los números se permiten en la porción del huésped [12]. Los octetos 0 y 255 están reservados para usos especiales. Una dirección donde todos los bits de la porción del huésped son 0 se refiere a la red, y una dirección donde todos los bits de la parte del huésped son 1 se llama una *dirección de difusión* (broadcast). Ejemplo, para la tabla 2:

La dirección de difusión es una especial a la que escucha cada máquina en la red además de a la suya propia. Esta dirección es a la que se envían los datagramas si se supone que todas las máquinas de la red lo deben recibir. Ciertos tipos de datos, como la información de encaminamiento y los mensajes de aviso son transmitidos a la dirección de difusión para que cada huésped en la red pueda recibirlo simultáneamente.

Dirección IP	192.168.120.21
Máscara de red	255.255.255.0
Número de red	192.168.120.
Número de huésped	.21
Dirección de Red	192.168.120.0
Dirección de Difusión	192.168.120.255

Cuadro 4: La tabla 2 con la dirección de red y de difusión

Hay dos estándares usados comúnmente al respecto de la dirección de difusión. El más ampliamente aceptado es el de usar la dirección más alta posible en la red. En el ejemplo de la Tab. 4 sería 192.168.120.255. Por alguna razón, otras estaciones han adoptado la convención de usar las direcciones de red como direcciones de difusión. En la práctica no importa mucho cual se use, pero hay que asegurarse de que cada máquina en la red está configurada con la misma.

Otras direcciones de red reservados para usos especiales son la 0.0.0.0 y 127.0.0.0. La primera es llamada *ruta por defecto* y la segunda es la *dirección propia* (loopback).

La red 127.0.0.0 esta reservada para el tráfico IP local en el huésped propio. Usualmente la dirección 127.0.0.1 será asignada a una interfaz especial en el huésped, la *interfaz propia*, la cual actúa como un circuito cerrado. Cualquier paquete IP manejado por esta interfaz será regresado a ella misma tal como si fuese recibido desde alguna otra red. Esto permite desarrollar y probar el software de red aún si no se tiene una red “real”. La red propia también permite usar software de red sobre un huésped aislado.

Algunos intervalos de direcciones para cada clase se han dejado fuera y se han asignado para direcciones “privadas”. Estas direcciones se han reservado para ser usadas en redes privadas y no son ruteadas hacia Internet. Estas son usadas por las empresas para construir sus propias intranets, pero se usan aún en redes muy pequeñas. Estas direcciones reservadas se presentan en la tabla 5

Clase	Máscara de red	Direcciones de red
A	255.0.0.0	10.0.0.0
B	255.255.0.0	172.16.0.0 - 172.31.0.0
C	255.255.255.0	192.168.0.0 - 192.168.255.0

Cuadro 5: Direcciones reservadas para intranets

De la tabla 5 se desprende que hay una red reservada clase A, 16 redes reservadas clase B y 256 redes reservadas clase C.

Para instalar un nuevo huésped en una red IP existente se debe contactar con los administradores de la red y preguntarles por la siguiente información:

- Dirección IP del huésped
- Dirección IP de la red
- Dirección IP de broadcast
- Máscara de red IP
- Dirección del encaminador (router)
- Dirección del Servidor de Nombre de Dominio (DNS)

Se debería configurar entonces el dispositivo de red del huésped con esos detalles. No pueden inventarse y esperar que la configuración funcione.

Para construir una nueva red propia que nunca conectará con Internet, esto es, si está construyendo una red privada y no tiene intención de conectar nunca esa red a Internet, entonces puede elegir las direcciones que quiera. De todas maneras, por razones de seguridad y consistencia, se deben de usar las direcciones reservadas presentadas en la tabla 5.

2.4. Subredes

Una subred [13] es un medio para tomar una sola dirección de red IP y localmente particionarla de forma que esta sola dirección IP pueda ser usada realmente en varias redes locales interconectadas. Recuerde que un número de red IP solo puede usarse sobre una sola red.

La palabra importante aquí es *localmente*: para todo el mundo fuera de las máquinas y las redes físicas cubiertas por la red IP puesta como subred, nada ha cambiado – es solo una red IP–. Esto es muy importante: hacer una subred es una configuración local que es invisible al resto del mundo.

2.4.1. ¿Por qué se usan las subredes?

Las razones detrás de las subredes vienen de las primeras especificaciones de IP, donde solo unos pocos sitios estaban ejecutando números de red clase A. Una red clase A permite millones de huéspedes conectados.

Si todas la computadoras IP en un sitio grande tuviesen que estar conectadas a la misma red, resultaría obviamente tanto en un tráfico enorme como un problema de administración: tratar de manejar tal bestia enorme podría ser una pesadilla y la red podría (casi con certeza) colapsar bajo la carga de su propio tráfico (se saturaría).

Entrando a las subredes: la direcciones de la red IP clase A podrían dividirse para permitir su distribución a través de varias (si no es que muchas) redes separadas. También la administración de cada red separada puede delegarse fácilmente. Esto permite tener redes pequeñas, manejables, que puedan establecerse, quizás usando tecnologías de red diferentes. Hay que recordar que no se pueden mezclar Ethernet, Token Ring, FDDI, ATM, etc., sobre la misma red física. Sin embargo, las diferentes tecnologías si pueden interconectarse.

Otras razones para la realización de subredes son:

- La distribución física del sitio puede crear restricciones (el largo de los cables, por ejemplo) en términos de cómo la infraestructura física puede conectarse, requiriendo múltiples redes. Las subredes permiten realizar esto en un ambiente IP usando un solo número de red IP. Esto es de hecho de realización muy común entre los Proveedores de Internet, los cuales tienen tienen que dar conectividad permanente a clientes con redes locales con números IP estáticos.
- El tráfico de red es lo suficientemente alto para causar caídas significantes. Partiendo la red usando subredes, el tráfico que es local en un segmento de red puede mantenerse local, reduciendo el tráfico total y acelerando la conectividad de la red sin requerir más ancho de banda.
- Los requerimientos de seguridad bien pueden dictar que diferentes clases de usuarios no compartan la misma red, ya que el tráfico sobre una red puede siempre ser interceptado por un usuario experimentado. Las subredes proveen una forma de mantener el departamento de mercadotecnia fuera del fisgoneo de tráfico de red del departamento de Investigación y Desarrollo (o a los estudiantes fuera del fisgoneo sobre la red de administración).
- Se cuenta con equipos que usan tecnologías de red incompatibles y que es necesario interconectar.

2.4.2. Cómo realizar una subred de un número de red IP

Una vez que se ha decidido poner subredes en el número de red que se tenga, ahora ¿cómo ha de realizarse esto? De forma general tienen que realizarse los siguientes pasos (que luego se explicarán en detalle):

- Poner la conectividad física (cables de red e interconexiones, tales como ruteadores).
- Decidir que tan grande/pequeña se necesita cada subred en términos del número de dispositivos que se conectarán a ellas, esto es, cuantos números IP útiles se requieren para cada segmento individual.

- Calcular la máscara de red y las direcciones de red apropiadas.
- Asignar a cada interfaz sobre la red su propia dirección IP y la máscara de red apropiada.
- Configurar las rutas sobre los ruteadores y las compuertas apropiadas, las rutas y/o rutas por defecto sobre los dispositivos de red.
- Probar el sistema y arreglar los problemas.

Para propósitos de ejemplo, se considerará que se crearán subredes sobre un número de red clase C: 192.168.1.0. Esto provee hasta un máximo de 256 interfaces conectadas, más el número de red obligatorio (192.168.1.0) y la dirección de difusión (192.168.1.255).

2.4.3. Poniendo la conectividad física

Se debe de instalar la infraestructura correcta de cableado para todos los dispositivos que se deseen interconectar para alcanzar la distribución física.

También será necesario tener un mecanismo para interconectar varios segmentos (ruteadores, convertidores, etc.).

2.4.4. Tamaño de la subred

Existe un compromiso entre el número de redes que se pueden crear y los números IP "perdidos".

Cada red IP individual tienen dos direcciones que ni pueden usarse como direcciones para una interfaz (ó huésped), el número de red IP en sí mismo y la dirección de difusión. Cada subred tiene estas dos direcciones que no pueden usarse:su propio número de red y dirección de difusión, además de direcciones válidas en el intervalo proveído por la red IP que se quiera dividir en subredes.

De esta forma, haciendo subredes de una dirección IP en dos subredes separadas existen ahora dos direcciones de red y dos direcciones de difusión, incrementando las direcciones "perdidas"; crear cuatro subredes crea ocho direcciones que no pueden usarse; etc.

De hecho, la subred útil más pequeña conste de solo cuatro números IP:

- Dos números IP para las interfaces, una para la interfaz del ruteador sobre esta red y otro para la interfaz del huésped sobre la red.
- Un numero de red.
- Una dirección de difusión.

Para qué se quería crear una red tan pequeña ya es otra pregunta. Con solamente un huésped sobre la red, cualquier comunicación en red debe ir hacia otra red. Sin embargo, este ejemplo sirve para mostrar las leyes de disminución que se aplican a las subredes.

Por principio, solamente se puede dividir una número de red IP en 2^n (donde n es menor en uno que el número de bits de la parte del huésped del número de red IP que se esté manejando) subredes de igual tamaño (sin embargo, se pueden hacer subredes de una subred ó combinar subredes)

Para ser realistas sobre el diseño de una red propia, se requiere el número mínimo de redes locales separadas que sea consistente con las restricciones de administración, físicas, de equipo y seguridad.

2.4.5. Cálculo de la máscara de subred y los números de red

La máscara de red es la que realiza toda la magia local de dividir una red IP en varias subredes.

La máscara de red para un número de red IP sin subredes es simplemente un número de red que tiene todos los bits de red puestos a '1' y todos los bits del huésped puestos a '0'. Para las tres clases de redes IP, las máscaras de red estándar son:

- Clase A (8 bits de red) : 255.0.0.0
- Clase B (16 bits de red): 255.255.0.0
- Clase C (24 bits de red): 255.255.255.0

La forma de que una subred opera es tomar prestado uno o más de los bits del huésped disponibles y hacer que las interfaces localmente interpreten estos bits prestados como parte de los bits de red. De manera que para dividir un número de red en dos subredes, podríamos tomar prestado un bit del huésped poniendo a uno el bit apropiado en la máscara de red del primer bit del huésped. Para una red clase C, resultaría una máscara de red de 11111111.11111111.11111111.10000000, ó 255.255.255.128

Para la red de clase C, por ejemplo, 192.168.1.0, estas son algunas de los opciones que tenemos para realizar subredes:

Redes	Huéspedes/red	Máscara de red
2	126	255.255.255.128 (ff.ff.ff.10000000)
4	62	255.255.255.192 (ff.ff.ff.11000000)
8	30	255.255.255.224 (ff.ff.ff.11100000)
16	14	255.255.255.240 (ff.ff.ff.11110000)
32	6	255.255.255.248 (ff.ff.ff.11111000)
64	2	255.255.255.252 (ff.ff.ff.11111100)

En principio, no hay ninguna razón para seguir el camino explicado para realizar la subred, donde los bits de las máscara de red son adicionados en el bit del huésped más significante hacia el bit del huésped menos significante. Sin embargo, si no se realiza de esta manera, los números IP resultantes seguirán una secuencia bastante extraña. Esto lo hace extremadamente difícil, para nosotros los humanos, decidir cual subred pertenece un número IP, ya que nosotros no somos tan buenos para pensar en binario (las computadoras, por otro lado, se manejan igual de bien en cualquier esquema).

Una vez que se ha decidido por la máscara de red apropiada, se tienen que resolver las direcciones de red y de difusión, y el intervalo de números para cada una de las redes. Considerando solamente un número de red clase C, y listando solo la parte final (la porción de huésped) se tiene:

Máscara	Subredes	Red	Difusión	MinIP	MaxIP	Huéspedes	Huéspedes totales
128	2	0	127	1	126	126	
		128	255	129	254	126	252
192	4	0	63	1	62	62	
		64	127	65	126	62	
		128	191	129	190	62	
		192	255	193	254	62	248
224	8	0	31	1	30	30	
		32	63	33	62	30	
		64	95	65	94	30	
		96	127	97	126	30	
		128	159	129	158	30	
		160	191	161	190	30	
		192	223	193	222	30	
		224	255	225	254	30	240

Como puede verse, existe una secuencia muy definida de estos números, lo cual los hace muy fácil de checar. La desventaja de las subredes también puede verse, ya que se reduce el número total de direcciones de huésped disponibles al mismo tiempo que se incrementa el número de subredes.

Con toda esta información, ya se está en la posición de asignar números de huéspedes, números de redes IP y máscaras de red.

3. Redes con el Sistema GNU/Linux

En este capítulo vamos a aprender a cómo implantar una red.

Empezaremos con dos ejemplos: uno muy sencillo, para conector solo dos huéspedes y otro ejemplo para realizar una red que conecte varias computadoras, esto es, vamos a realizar intranets.

Conoceremos las reglas básicas para realizar un cableado estructurado.

Y en este capítulo también conoceremos las reglas básicas para el diseño formal de una Intranet.

Los ejemplos de configuración de las intranets se harán para el sistema GNU/Linux, para la distribución de Red Hat [14], sin embargo no debe ser mayor problema la realización de las configuraciones de las interfaces de red para otro sistema operativo.

3.1. Configuración de las interfaces de red

Recordemos (del capítulo 2), TCP/IP define una *interfaz* abstracta a través de la cual se puede accesar al hardware, esto a su vez es un mecanismo que oculta la diversidad de equipo que puede usarse en un ambiente de red [11].

En una máquina con el sistema operativo GNU/Linux, si tuviésemos dos tarjetas de red ethernet, a cada interfaz que “ve” TCP/IP se le identifica como *eth0* y *eth1* (que son *nemónicos* a *ethernet-0* y *ethernet-1*).

La identificación de las interfaces de red, esto es, que el sistema asigne los nombres eth0 y eth1 para las dos interfaces de red instaladas en el sistema, se realiza de forma automática al instalar el sistema operativo. Esto es posible por el programa *kudzu*. Kudzu es la herramienta de configuración y autodetección de hardware, originalmente se introdujo en la distribución Linux de Red Hat versión 6.1. Este programa detecta cambios en la configuración del hardware del sistema y nos da la opción de adicionar o remover dispositivos. Este se ejecuta por defecto cada vez que se reinicia la computadora [15].

Los datos principales que debemos conocer para la configuración de nuestra red son los siguientes:

1. Nombre del huésped.
2. Su dirección IP
3. La máscara de red
4. La dirección de difusión
5. La dirección de la puerta

Y algunas cosas más necesitamos configurar, como por ejemplo un servidor de nombres. Para nuestros ejemplos, usaremos la *dirección de difusión* por defecto (que corresponde al número IP más alto de nuestra red. Por ejemplo, para la red 10.1.1.0/255.255.255.0, la dirección de difusión sería la 10.1.1.255). La *dirección de la puerta* no es necesario por ahora. Esta se utiliza para interconectar redes.

Para las redes que vamos a diseñar es conveniente que usemos los números IP asociados a las redes privadas y mostrados en la tabla 6. Estos número son filtrados por todos los ruteadores sobre Internet y facilitaría la conexión futura de la intranet a Internet.

Clase	Máscara de red	Direcciones de red
A	255.0.0.0	10.0.0.0
B	255.255.0.0	172.16.0.0 - 172.31.0.0
C	255.255.255.0	192.168.0.0 - 192.168.255.0

Cuadro 6: Direcciones reservadas para intranets

Claro que podríamos usar un esquema de subredes para configurar nuestra intranet. Pero para facilitarnos las cosas vamos a utilizar máscaras del tipo 255.255.255.0, lo que nos permitirá conectar a lo más 254 huéspedes (recordemos que las direcciones IP con terminación 0 y 255 están reservadas).

Resulta obvio que podríamos subdividir una red clase A ó B, con los números reservados en la tabla 6, en redes clase C (esto se logra fácilmente usando una máscara 255.255.255.0).

Para los ejemplos que veremos vamos a usar la red 192.168.10.0/255.255.255.0.

3.1.1. Ejemplo de configuración de una interfaz de red

Supongamos que queremos asignar el número IP 192.168.10.2/255.255.255.0 a una máquina con su tarjeta de red instalada. La máquina la llamaremos "luna". Para ello editamos el archivo

```
/etc/sysconfig/network
```

con el contenido

```
NETWORKING=yes
HOSTNAME=luna
```

Y los datos para IP se editan en el archivo:

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

donde el nombre de archivo `ifcfg-eth0` indica que es el archivo de configuración para la interfaz eth0. Para la interfaz ethN, el archivo deberá llamarse `ifcfg-ethN`, esta N se sustituye por el número de la interfaz.

El contenido del archivo `ifcfg-eth0` para asignar el número IP 192.168.10.2/255.255.255.0 deberá ser:

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.10.2
NETMASK=255.255.255.0
NETWORK=192.168.10.0
BROADCAST=192.168.10.255
```

Aunque no es estrictamente necesario agregar el número de red y el de difusión (BROADCAST), lo ponemos por claridad.

Para que los cambios tengan efecto llamamos al comando:

```
/etc/rc.d/init.d/network stop
/etc/rc.d/init.d/network start
```

ó simplemente como:

```
/etc/rc.d/init.d/network restart
```

Los comandos arriba presentados son en realidad *scripts* que detectan la información (usando algunos otros scripts) en el archivo `ifcfg-eth0` y mandar a llamar al comando *ifconfig* y *route*. La ventaja de hacerlo como es presentado aquí es que los cambios se hacen permanentes a la vez que se mantienen cuando se reinicia la computadora.

Con los datos presentados arriba, directamente se puede llamar al comando *ifconfig* y *route* de la siguiente manera:

```
ifconfig eth0 192.168.10.2 \
         netmask 255.255.255.0 \
         broadcast 192.168.10.255
route add -net 192.168.10.0
route add default gw 192.168.10.1
```

pero los cambios solo serán temporales. El símbolo "\" al final de la primera línea indica que se continua la línea (todo es una solo línea, pero se dividió en dos para que cupiese en el ancho del documento). La última línea adiciona la ruta para todos los paquetes que alcanzan Internet a través de la dirección de la puerta 192.168.10.1. La penúltima línea pone la ruta para la red de la máquina cliente [11].

3.2. Conexión de dos computadoras por medio de sus tarjetas de red

En la Fig. 5 vemos la que podríamos llamar "una red mínima": son dos computadoras que queremos comunicar a través de las tarjetas de red y un cable par trenzado.

El cable debe ser un cable nulo (o cable cruzado). La manera de hacer este cable ya se ha mencionado en el segundo programa. Si A es el primer conector RJ-45 y B el segundo. El cable nulo se realizar conectando 1A-3B, 2A-6B, 3A-1B y 6A-2B; los primeras dos conexiones deben ser con un solo par trenzado y

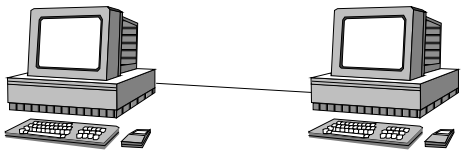

Figura 5: “La red mínima”

al igual que las dos últimas (si no lo hacemos así no servirá para una red a 100 Mbs).

Debemos ahora dar nombre a los dos computadoras y asignar sus números IP. A una computadora la llamaremos “luna” y a otra “sol”. Le asignaremos los IPs 192.168.10.2 y 192.168.10.3, respectivamente. Recordemos que la red que estamos usando es la 192.168.10.0/255.255.255.0. Ambas direcciones las configuramos con se ha planteado en la subsección 3.1.1.

Ahora vamos a probar que la red funciona. Desda la máquina “luna” hacemos un *ping* a la máquina “sol”:

```
ping 192.168.10.3
```

Si todo lo hemos realizado bien, la salida debe ser algo como:

```
$ ping 192.168.10.3
PING 192.168.10.3 from 192.168.10.2 : 56(84) bytes of data.
Warning: time of day goes back, taking countermeasures.
64 bytes from 192.168.10.3: icmp_seq=0 ttl=255 time=262 usec
64 bytes from 192.168.10.3: icmp_seq=1 ttl=255 time=169 usec
64 bytes from 192.168.10.3: icmp_seq=2 ttl=255 time=148 usec
64 bytes from 192.168.10.3: icmp_seq=3 ttl=255 time=143 usec
64 bytes from 192.168.10.3: icmp_seq=4 ttl=255 time=153 usec

--- 192.168.10.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.143/0.166/0.262/0.040 ms
```

Oprimimos las teclas control-D para terminar.

Para que podamos “conocer” las máquinas por nombre, tenemos que editar sus archivos `/etc/hosts` correspondientes; por ejemplo, para nuestra configuración, este archivo debe contener:

```
127.0.0.1     localhost.localdomain localhost
192.168.10.2  luna
192.168.10.3  sol
```

De esta forma ya podemos lanzar el comando *ping* con el nombre de la computadora (sol) en vez de su número IP (192.168.10.3) desde “luna” así:

```
ping sol
```

3.3. Conexión en red de varias computadoras

En la Fig. 6 se muestra el diagrama de la red que vamos a realizar ahora virtualmente. En la figura observamos que vamos a conectar cinco computadoras, y

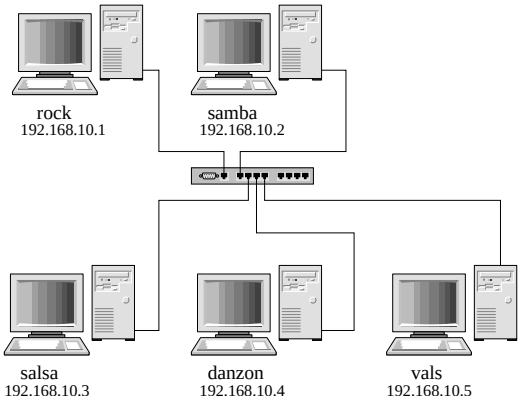

Figura 6: Red de cinco computadoras.

a diferencia de la red vista en la Sec. 3.2 (y mostrada en la Fig. 5) en donde conectamos dos computadoras con un cable nulo, necesitamos ahora un etherswitch o un concentrador. Un etherswitch es mejor si vamos a tener una comunicación intensa entre las máquinas. Los cable para conectar cada computadora al etherswitch son cables normales: si A es un conector RJ-45 y B es el otro conector RJ-45, el cable normal se construye conectando los pines 1A-1B, 2A-2B, estos dos deben ser un solo par trenzado; se conecta también 3A-3B y 6A-6B y deben ser otro par trenzado. Se recomienda conectar los otros dos pares a los conectores RJ-45.

3.3.1. Hardware necesario

Resumiendo, para construir la red mostrada en la Fig. 6, las consideraciones de equipo necesario son:

1. Las cinco computadoras
2. Cinco tarjetas de red 100 Mbps, con conectores RJ-45
3. Un etherswitch (o un concentrador) con al menos cinco puertos (uno para cada computadora)
4. Cinco cables *normales* de red (que involucran cinco cables y 10 conectores RJ-45)

3.3.2. Software necesario

Si en cada computadora estuviese instalado el sistema operativo GNU/Linux, el software requerido para funcionamiento en red ya viene incluido. Tendríamos que checar que hay un manejador disponible para las tarjetas de red. Esto se confirma pagando la máquina, poniendo una tarjeta de red en ella y volviéndola a encender. Automáticamente el software *kudzu* encontraría la tarjeta y la configuraría (creando el dispositivo de red *eth0*).

Si tuviésemos alguno de los *sabores* de Windows instalado en algunas de las máquinas que queremos meter en la red, tendríamos que insertar la tarjeta de red, configurar el manejador de la tarjeta y verificar que tenga el software para TCP/IP asociado a la tarjeta de red.

3.3.3. Configuración de la red

Podríamos usar una subred de tamaño 8 (ocho). Pero siguiendo con nuestros ejemplos utilizaríamos la red 192.168.10.0/255.255.255.0 (esto es, una red clase C).

Los datos necesario para cada máquina son:

1. Número IP: De la 192.168.10.1 a la 192.168.10.5, como se muestra en la Fig. 6.
2. Máscara de red: 255.255.255.0
3. Red: 192.168.10.0
4. Difusión (por defecto): 192.168.10.255

Todas las interfaces de red se configurarían como fue explicado en la Sec. 3.1.

Si todo lo hemos realizado correctamente, podremos hacer hacer la prueba de que la red funciona haciendo *pings* entre todas las máquinas.

Sería necesario agregar la lista de todas los nombres de las máquinas en el archivo `/etc/hosts`, que quedaría como:

```
127.0.0.1    localhost.localdomain localhost
192.168.10.1 rock
192.168.10.2 samba
192.168.10.3 salsa
192.168.10.4 danzon
192.168.10.5 vals
```

Se ha adoptado poner nombres de “músicas” a las máquinas. Resulta simpre divertido tratar de asignar nombres a todas las computadoras.

Tener una lista de nombres para más máquinas claramente se vuelve impráctico. Esto se resolvería con un Servidor de Nombres de Dominio (DNS en inglés, Domain Name Server).

Algunos de los servicios más importantes que podríamos poner a nuestra red (como el compartir una impresora, o compartir espacio en disco o compartir archivos) serán tratados en el siguiente capítulo.

3.4. Ruteo entre varias redes

Para realizar la comunicación entre varias redes o subredes (esto se conoce como *ruteo*) se puede usar una computadora personal con dos (ó más) interfaces de red.

Para efectuar el ruteo se tiene que habilitar el *traspaso IP* (en inglés, IP Forwarding) dentro del núcleo del sistema operativo. En un sistema con Linux RedHat se puede hacer editando el archivo `/etc/sysctl.conf`:

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
```

esto es, se pone un “1” en la opción `net.ipv4.ip_forward`. Para que el cambio tenga efecto se relevanta el servicio de red con el comando:

```
/etc/rc.d/init.d/network restart
```

Existen otras dos formas de activar el traspaso de IP (y se podría escoger alguna de ellas):

1. Realizando la instrucción siguiente en algún script:
   `echo 1 > /proc/sys/net/ipv4/ip_forward`
2. Editando el archivo /etc/sysconfig/network y agregando la línea
   `FORWARD_IPV4=true`

Para propósito de explicar el ruteo, veremos el siguiente ejemplo: supóngase que se ha decidido dividir en cuatro subredes la red clase C 192.168.1.0 con la que contamos (cada subred tendrá 62 números IP usables para sus huéspedes o interfaces de red dentro de ella). Sin embargo, dos de esas subredes se combinarán en una sola red más grande, resultando en tres redes físicas:

Red	Difusión	Máscara	Huéspedes
192.168.1.0	192.168.1.63	255.255.255.192	62
192.168.1.64	192.168.1.127	255.255.255.192	62
192.168.1.128	192.168.1.255	255.255.255.128	124

Nota: La redes de que las dos últimas redes tengan sólo 124 direcciones usables es que este es en realidad una *super red* formada con dos subredes. Los huéspedes en las otras dos redes interpretarán el número 192.168.1.192 como la dirección de red de la subred que ‘no existe’. De forma similar, esas redes interpretarán el número 192.168.1.191 como la dirección de difusión de la subred ‘no existente’.

Por lo tanto, si se usan los números 192.168.1.191 ó 192 como direcciones para huéspedes sobre la tercera red, entonces las máquinas conectadas en las dos redes más pequeñas no serán capaces de comunicarse con ellas.

Lo anterior también ilustra un punto importante al trabajar con subredes: la direcciones útiles están determinadas con las subredes *más pequeñas* del espacio total de direcciones [13].

3.4.1. Las tablas de ruteo

Vamos a suponer que tenemos una máquina con Linux que se usará como ruteador para la red de nuestro ejemplo. La máquina deberá tener tres interfaces de red para las tres intranets y, posiblemente, una cuarta interfaz para conectarse a Internet (la cual debería sser su ruta por defecto).

Vamos a suponer también que la computadora con Linux usa las direcciones IP disponibles más pequeñas de cada subred, para la interfaz a cada red. Esto resultaría en la siguiente configuración para las interfaces de red:

Interfaz	Dirección IP	Máscara
eth0	192.168.1.1	255.255.255.192
eth1	192.168.1.65	255.255.255.192
eth2	192.168.1.129	255.255.255.128

El ruteo que esto podría establecer sería:

Destino	Puerta	Máscara	Interfaz
192.168.1.0	0.0.0.0	255.255.255.192	eth0
192.168.1.64	0.0.0.0	255.255.255.192	eth1
192.168.1.128	0.0.0.0	255.255.255.128	eth2

Es aqui donde vemos que el parámetro de la *puerta* (gateway, en inglés) toma sentido: sirve para rutear, o comunicar diferentes redes.

Ahora, sobre cada una de las subredes, los huéspedes podrían configurarse son su propio número IP y la máscara de red (la apropiada para cada subred en particular). Cada huesped debe de declarar la máquina Linux como su puerta (gateway), especificando la dirección IP de la máquina Linux por su interfaz sobre la red en particular.

4. Configuración de una Puerta

Vamos a conocer como instalar una *puerta) (*gateway) en inglés) para el sistema GNU/Linux, específicamente para la distribución de RedHat para los núcleos versión 2.4. Una puerta es una máquina con dos (o más) interfaces de red y que sirve para conectar dos (o más) redes diferentes. Aquí se recomienda su uso para interconectar intranets (o redes con direcciones IP no válidas), ó para conectar intranets a Internet. Esta última aproximación solo require unas cuantas (dos, cuatro, ocho a lo más) IP válidas para conectar cientos de máquinas (clientes) a Internet. El mismo esquema para realizar una puerta se usa para configurar cortafuegos con Linux.

Recordemos, las direcciones IP *no válidas* son las especificadas en el RFC1918 [16, 17, 18] para diseñar redes privadas o intranets. Estas son $10.*.*.*$, $172{,}16.*.*$ a $172{,}31.*.*$ y $192{,}168.*.*$. Todos los ruteadores actuales filtran estas direcciones por lo que no se pueden accesar estas direcciones desde Internet.

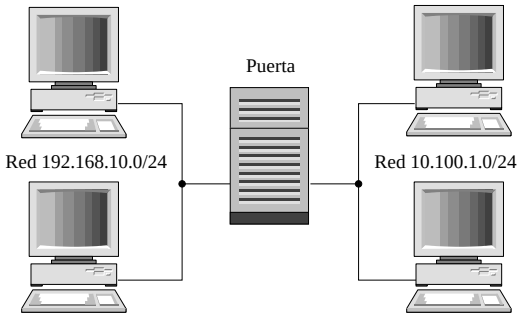

Figura 7: Esquema de una puerta básica

4.1. Una puerta simple

En la Fig. 7 se presenta el esquema de una puerta básica. Esta puerta es una máquina con pocas características, como 486, pentium, 32M en RAM y un disco de 1GB (puede hacerse hasta en un disquete) con el sistema Linux básico instalado (sin el sistema X y ningún servidor). La puerta conecta a las dos redes 192.168.10.0/24 y 10.100.1.0/24 y nos sirve para que una computadora dentro de una red pueda “ver” a otra computadora situada en la otra red. El esquema de la Fig. 7 solo tiene dos computadoras dentro de cada red y no se representa los etherswitches para conectar a todas las computadoras dentro de cada red.

La puerta básica de la Fig. 7 necesita tener dos tajetas de red instaladas. En los kernels 2.4 y en la distribución de RedHat de Linux, el programa *kudzu* detecta automáticamente éstas tarjetas al arranque del sistema.

La configuración para la puerta básica solo requiere las parámetros de red para sus dispositivos de red y habilitar el transpaso de paquetes entre las redes, como fue explicado en la sección anterior. Considerando que la puerta de la Fig. 7 tiene dos tarjetas ethernet, podrían ponerse para cada una los siguientes parámetros

eth0: IP: 192.168.10.254, Máscara: 255.255.255.0

eth1: IP: 10.100.1.254, Máscara: 255.255.255.0

de esta forma, el valor del parámetro de la puerta (*gateway*) para todas las máquinas dentro de la red 192.168.10.0 será 192.168.10.254 y para las máquinas dentro de la red 10.100.1.0 tendrá el valor 10.100.1.254. Por supuesto el etherswitch para la red 192.168.10.0 tendrá que conectarse a la tarjeta asignada a la interfaz eth0 y lo mismo para el etherswitch para la red 10.100.1.0 que tendrá que conectarse a eth1.

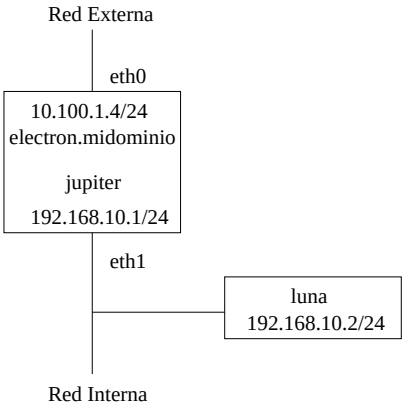

Figura 8: Esquema de uso de una puerta

4.2. Configuración de la puerta con IP-tables

La conección de laboratorios de cómputo, salas de cómputo, etc., en donde es necesario tener acceso a Internet, esto es, en un escenario donde necesitemos instalar de unos pocos a cientos de *clientes* de Internet, *no es necesario* asignar direcciones IP válidas a cada máquina, o instalar un servidor DHCP que ofrezca direcciones IP válidas. Con una máquina Linux podemos interconectar estos clientes formando una intranet y poner esta máquina como una *puerta* para que todos los clientes "vean" Internet. Este esquema también se conoce como *traducción de direcciones de red*.

El esquema aquí presentado no funciona directamente si se quiere poner una máquina para ofrecer información a Internet. Esto es un *servidor* y utiliza un esquema diferente que puede revisarse en la siguinte sección, en la pág. 22.

Para más información se recomienda leer los HOWTO's [16, 17] y los artículos [18, 19, 20, 18, 21].

El esquema que se va a usar se presenta en la Fig. 8. Una máquina Linux tiene dos tarjetas ethernet. Una de las interfaces, eth0, está conectada a Internet (aunque en la figura se muestra una dirección no válida para este dispositivo por razones obvias de seguridad, eth0 debe tener asignada una dirección IP válida) y la otra, eth1, se usa para conectar la máquina con la red privada.

En la Fig. 8, la puerta conoce como "electron.midominio" desde Internet y como "júpiter" desde la red interna.

Debemos de configurar las intertfaces de red de la puerta y de los clientes, información que puede extraerse de la Fig. 8. Aquí falta un detalle: el valor para el parámetro de las puerta (gateway) para los clientes (la máquina "luna" en la Fig. 8 es un ejemplo de un cliente) debe ser la dirección 192.168.10.1.

Por supuesto, debe se levantarse el traspaso de paquetes entre las interfaces de red como fue explicado en la sección anterior (en la pág. 3.4).

4.2.1. Un cliente

A la máquina cliente "luna" (ver Fig. 8) se le instaló una tarjeta de red (eth0) y se editó el archivos de configuración de red /etc/sysconfig/network con algo como

```
NETWORKING=yes
HOSTNAME="luna"
```

Y el archivo /etc/sysconfig/network-scripts/ifcfg-eth0 con la información:

```
DEVICE=eth0
IPADDR=192.168.10.2
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=static
GATEWAY= 192.168.10.1
```

Y también, para que se reconozcan los cambios en la configuración, hay que tirar y levantar los servicios de la red con la siguiente instrucción:

```
/etc/rc.d/init.d/network restart
```

4.2.2. Configuración de IP-tables

Iptables es tanto un comando para introducir reglas para filtrar paquetes como el módulo del núcleo (kernel) que lo permite realizar. Todo la distribución se conoce como *netfilter*. Netfilter es el sistema compilado dentro del núcleo que permite realizar ciertas cosas en la pila de IP con los módulos cargables (iptables es uno de ellos) que permite realizar operaciones sobre los paquetes [22].

Resulta difícil controlar las reglas que se introducen al núcleo si se realiza una a una. Por ello lo mejor es hacer un script que introduzca al núcleo todas las reglas a la vez. El siguiente script realiza la configuración para crear una puerta habilitando la traducción de direcciones de red.

```
#!/bin/sh
# Configuración de una ZM con iptables

PATH=/sbin
LOOPBACK_INTERFAZ=lo
INTERFAZ_EXT=eth0
IPADDR=10.100.1.4/32
REDLOCAL=10.100.1.0/24
#
INTERFAZ_INT=eth1
REDINTERNA=192.168.10.0/24
UNIVERSO=0.0.0.0/0

# Limpiamos las reglas actuales
iptables -F
iptables -F -t nat
```

```
# Quitamos cadenas propias
iptables -X
##---------------------------------
# Establecer la política por defecto
#      entrada DENEGADA
#      FORWARD DENEGADA
#      salida  DENEGADA
##---------------------------------
iptables -P INPUT   DROP
iptables -P FORWARD DROP
iptables -P OUTPUT  DROP
##--------
# LOOPBACK
# Tráfico sin límite en la interface "loopback"
iptables -A INPUT -i $LOOPBACK_INTERFAZ -j ACCEPT
iptables -A OUTPUT -o $LOOPBACK_INTERFAZ -j ACCEPT

##--------
# TRASPASO (FORWARDING) ó traducción de direcciones

# Bloquea los paquetes falsificados, o "exagerados,"
# que pasen a través de la puerta
iptables -A FORWARD -i $INTERFAZ_INT \
     -s ! $REDINTERNA -j DROP

# Permite a todos los paquetes internos salir de la red
iptables -A FORWARD -m state --state NEW,ESTABLISHED \
    -i $INTERFAZ_INT -s $REDINTERNA -j ACCEPT

# Permite que regresen los paquetes asociados con
# las conexiones de arriba
iptables -A FORWARD -m state --state ESTABLISHED,RELATED \
    -i $INTERFAZ_EXT -s ! $REDINTERNA -j ACCEPT

# Todo el tráfico interno es enmascarado externamente
iptables -A POSTROUTING -t nat -o $INTERFAZ_EXT \
    -j MASQUERADE
```

Para que los cambios sean permanentes en el sistema, primero se ejecuta el script anterior y luego se salvan las reglas en un archivo con los siguientes comandos:

```
cd
iptables-save > iptables
cp iptables /etc/sysconfig
```

Otra forma es cambiar el script para que tenga el formato de un archivo de servicio de arranque en el directorio /etc/init.d y activarlo con el programa `chkconfig`.

4.3. Dos intranets con acceso a Internet

Un ejemplo interesante que se puede realizar con pequeños cambios del script presentado en la subsección anterior es el siguiente: se puede realizar una puerta que de acceso a Internet a dos intranets; se pueden poner algunas reglas para que se pueden conectar algunas, o todas, las computadoras entre una intranet y la otra –que se realizará más adelante cuando estudiemos a los cortafuegos–. El esquema de esta idea se presenta en la Fig. 9

En el diagrama de la Fig. 9, la interfaz eth0 de la puerta debe tener asignada una dirección IP válida, pero se presenta una dirección inválida solo por motivos de seguridad. El script para configurar la puerta puede

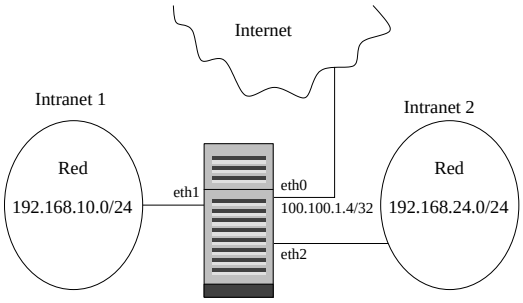

Figura 9: Esquema de una puerta usada para conectar dos intranets a Internet

realizarse a partir del script de la subsección anterior, solo se deben duplicar las reglas para cada intranet, y recordemos que el valor del parámetro de la puerta (gateway) para todas la máquinas dentro de la intranet 1 será la dirección IP de la interfaz eth1 de la puerta. Para las máquinas dentro de la intranet 2, el valor del parámetro de la puerta será la dirección IP de la interfaz eth2 de la puerta.

5. Consideraciones Básicas de Seguridad en Redes

En 1985, la Universidad de Carnegie Mellon, en los E.E.U.U., ganó una licitación para establecer el *CERT Coordination Center* con el apoyo monetario del Departamento de Defensa de los E.E.U.U. [23].

El CERT/CC es un centro para reportar todos los problemas de seguridad en Internet. Su personal provee recomendaciones y respuestas coordinadas a compromisos de seguridad, identifica intento de actividad de intrusión, trabaja con otros expertos en seguridad para identificar soluciones a problemas de seguridad y disemina información hacia toda la comunidad. El CERT/CC también analiza vulnerabilidades en productos, publica documentos técnicos y presenta cursos de entrenamiento.

El CERT/CC está dentro de Instituto de Ingeniería de Software de la Universidad de Carnegie Mellon, en Pittsburgh, E.E.U.U.

El CERT/CC [24] publica que un sitio ideal en seguridad debe contar con:

1. Estar al día en parches
2. Usar cortafuegos
3. Debe monitorearse la red.
4. Deben deshabilitarse los servicios y características que no son necesarios

5. Tener un software de antivirus instalado, configurado y actualizado.

6. Una política para la realización de respaldos.

7. Un equipo entrenado y con capacidad de respuesta a incidentes.

La seguridad de un sistema en red es un tema extenso. Aquí vamos a revisar con cierto detalle los puntos 2, 3 y 4, con soluciones basadas en el sistema GNU/Linux. Estos tres puntos pueden considerarse de vital importancia en una red que cuenta con uno o varios servidores y que está conectada a Internet.

En una red pequeña, sin conexión a Internet, los puntos 2, 3 y 4 pueden considerarse importantes aunque no de vital importancia. Sin embargo, deben de considerarse si nuestra red sufre compromisos de seguridad desde los mismos usuarios de nuestra red. Típicamente los compromisos de seguridad provienen de Internet, donde alguna persona, muy preparada y con muchos conocimientos, podría ingresar sin permiso a nuestras computadoras. Si nuestra red no está conectada a Internet, entonces no deberíamos angustiarnos por estos puntos, aunque sí preocuparnos si nuestros usuarios están muy preparados y podrían instalar programas que pueden comprometer la seguridad de la red.

Así que además de los sistemas aquí presentados es necesario contar con un equipo de trabajo dedicado a la seguridad de la red. Y nuestra experiencia es que se puede pensar en tener una red segura si se tiene un equipo de trabajo que lleve a cabo las ideas de seguridad.

Aquí se van a desarrollar los puntos siguientes:

1. Estar al día en parches

2. Tener un software de antivirus instalado, configurado y actualizado.

3. Una política para la realización de respaldos.

4. Un equipo entrenado y con capacidad de respuesta a incidentes.

Los otros tres puntos se desarrollarán en secciones subsecuentes:

- Usar cortafuegos.

- Monitoreo de la red.

- Deshabilitar los servicios y características innecesarios.

5.1. Parches actualizados

Todo el software de un centro de cómputo conectado en red necesita estar actualizado en parches. Un parche es una parte corregida del código de un programa. En el mundo de GNU/Linux generalmente se tienen los parches en código fuente, o en los sitios de Internet de los distribuidores de Linux (checar por ejemplo www.redhat.com). En el mundo de las sistemas Windows y Mac generalmente se presentan los parches como un nuevo código ejecutable. Un parche corrige un problema o un fallo de seguridad del programa que se trate. Es muy importante bajar los parches de sitios de Internet que sean confiables, ya que podríamos instalar un programa que transmita un virus o un gusano y comprometa la seguridad de toda nuestra red.

Cada distribución de GNU/Linux viene con más de 12,000 páginas de documentación. Las distribuciones comerciales de GNU/Linux, tales como Red Hat Linux, Caldera, SuSE, Mandrake, Turbo Linux y OpenLinux, ofrecen un soporte inicial para sus usuarios registrados; empresas y negocios pequeños pueden tener soporte por media de compañías de soporte comerciales. Como es un Sistema Operativo Abierto, no hay que esperar para que se lance una nueva versión del software, dado que la comunidad de usuarios de GNU/Linux fija, o arregla, muchos de los bugs (fallas del software) en horas [10].

5.2. Antivirus

El antivirus es un programa que debe tenerse con las licencias actualizadas para garantizar tener las últimas versiones del mismo. Todas nuestros sistemas Windows y Mac debiesen de tener el programa antivirus instalado, configurado y actualizado. En los sistemas Linux prácticamente no existen virus, la misma arquitectura del sistema (con la creación de usuarios y permisos de acceso a archivos) hace muy difícil la proliferación de cualquier virus.

Hay casos especiales en que un programa antivirus en Linux, para escanear virus en otros sistemas que no son Linux, tiene sentido. Por ejemplo, para la realización de un sistema que actúa como servidor de archivos para máquinas Windows (vía SAMBA). Como los virus de los archivos no pueden infectar el sistema Linux pero si a los clientes de Windows, el escanear los archivos desde el mismo servidor resuelve el problema de virus para todos los clientes.

5.3. Respaldos

Uno de los métodos infalibles para luchar contra agujeros de seguridad, fallas de los sistemas, desastres

físicos, virus, etc., es realizar respaldos (backups). Generalmente se deben de respaldar la información que es crucial para el sistema, como las bases de datos y los archivos de los usuarios. No es necesario guardar la información de sistema operativo ya que debemos de disponer de un medio (CDROM) para su reinstalación. En una máquina Linux/UNIX es importante realizar el respaldo de la información configurable del sistema que se encuentra bajo el directorio `/etc`. Los dispositivos de almacenamiento que tenemos disponibles son:

1. Discos duros
2. Discos compactos
3. Discos de video digital (DVDs)
4. Cintas magnéticas

Los primeros se convierten en una opción dado al abaratamiento y la gran capacidad de los discos duros actuales. Se puede realizar un "espejeo" de disco o una técnica que se conoce como RAID [25]. RAID es el acrónico en inglés de *Redundant Arrays of Inexpensive Disks* (ó Arreglo Redundante de Discos Baratos). La idea básica de RAID es combinar múltiples discos, independientes y pequeños, en un arreglo de discos en el cual su rendimiento excede de un solo disco grande y caro. De forma adicional este arreglo de discos es visto por la computadora como un solo disco lógico. La técnica RAID incluye también una tolerancia a fallos por medio del almacenamiento de información redundante en los discos.

Los discos compactos pueden ser una opción, la limitante son los 740MB de información que puede almacenarse en un CD. Claramente la opción con los DVD (que pueden almacenar hasta 4.7GB de información), que si disminuyen su precio pueden ser una clara opción para almacenar datos en ellos. La clásica opción que se tiene son las cintas magnéticas, que almacenas gigas y gigas de bytes en ellos, aunque su desventaja es que pueden ser muy lentas y su ventaja es que dan la mejor razón de precio/megabyte de información almacenada.

5.4. Personal entrenado y con capacidad de respuesta a incidentes

Algunos empresarios o responsable de entidades públicas consideran a el mantenimiento y a la seguridad de una red como actividades secundarias. Ejemplo de ello es que piensen que cualquier persona puede hacerse cargo del mantenimiento y la seguridad. O más aún, que cambien entre distinto personal, ya sea calificado o no, para la realización de estas dos importantes actividades. Hay que tomar en cuenta que quien da vida a una red es tanto sus usuarios, como el administrador (o administradores) de la red. El material humano es lo más importante para la buena realización de cualquier actividad. Para la administración y seguridad de una red es indispensable contar con personal entrenado y capaz para llevar a cabo las ideas de seguridad, que por ejemplo, conoceremos en este capítulo.

6. Virus, Gusanos, Troyanos y dos Sistemas de Lucha Contra Ellos.

Un virus de computadora es un programa que se replica a sí mismo, contiene código que explícitamente hace copias de sí mismo y que puede "infectar" otros programas modificándolos o cambiando su ambiente de forma que una llamada a un programa infectado implica una llamada a una posible copia evolucionada del virus [26].

Mucha gente usa vagamente el término "virus" para cubrir cualquier lista de programas que intenta ocultar su posible función maliciosa e/o intenta diseminarse en tantas computadoras como sea posible, aunque algunos de estos programas pueden llamarse de forma más correcta como "gusanos" ó "caballos de Troya". También hay que tener cuidado en considerar lo que es un "programa" que un virus puede infectar, puede incluir algo que no es obvio –¡no debe suponerse demasiado sobre lo que un virus puede o no puede hacer!–

Estas "bromas pesadas" del software son muy serias; pueden propagarse más rápido de lo que pueden ser detenidas, y aún el daño más pequeño podría implicar un daño la escenario de vida que tenemos. Por ejemplo, en el contexto del sistema de soporte de vida de un hospital, algún virus que "simplemente" detiene una computadora y despliega un mensaje hasta que se presione cualquier tecla, podría ser fatal. Más aún, aquellas personas que crean virus no pueden detener su diseminación, aún si lo quisieran. Esto requiere un esfuerzo concertado de los usuarios de computadoras para que sean cuidadosos con los virus, más que continuar con la ambivalencia que ha permitido que los virus se vuelvan un problema.

Los virus de computadora son de hecho un caso especial de algo que se conoce como "lógica maliciosa" o "malware". [26].

Un *gusano* de computadora es un programa auto contenido (o puede ser un conjunto de programas), que es capaz de diseminar copias funcionales de él mismo o de sus segmentos a otros sistemas de computadoras (generalmente vía conexiones de red).

Al contrario de un virus, un gusano no necesitan estar hospedado en algún programa. Existen dos clases de gusanos: gusanos de huésped y gusanos de red.

Los *gusanos de huésped* están enteramente contenidos en la computadora donde se están ejecutando y usan únicamente la red para copiarse a otras computadoras. Los *gusanos de red* consisten de muchas partes (que se llaman *segmentos*), cada una ejecutándose en máquinas diferentes (y posiblemente realizando acciones diferentes) y estas partes usan la red para propósitos de comunicación. La propagación de un segmento de una máquina a otra es solamente uno de esos propósitos.

Un *caballo de Troya* es un programa que hace algo que no está documentado por lo que programador hizo, pero que algunos usuarios no podrían aprobar si lo supiesen. Para algunas gentes, un virus es un caso particular de un caballo de Troya, debido a que es capaz de diseminarse a otros programas (esto es, los convierte también en caballos de Troya). Acorde a otros, un virus que no hace un daño deliberado (más que replicarse) no es un troyano. Por último, sin tomar en cuenta las definiciones, mucha gente usa el término *troyano* para referirse solo al malware que no se replica, de esta manera el conjunto de troyanos y el conjunto de virus son disjuntos.

En máquinas grandes, servidores y estaciones de trabajo, los problemas de seguridad se deben a agujeros de seguridad del Sistema Operativo o a troyanos. Los virus son un problema generalmente de PCs.

Con un programa se escaner para GNU/Linux se podrían construir dos sistemas de protección de virus:

1. Un servidor de archivos para máquinas Windows
2. Un escaner de correo electrónico

El primer sistema puede visualizarse en la Fig. 10. En la figura se nota un servidor de disco que puede realizarse con GNU/Linux y servidor disco a los tres clientes usando SAMBA (este realiza el protocolo para poder comunicarse con los sistemas Windows). GNU/Linux es inmune a los virus hechos para Windows, esto es, con el servidor se garantiza que no desaparecerán los archivos si un virus o gusano destruye el sistema de archivos (podrían destruir el sistema de archivos de los clientes solamente), pero para evitar que un virus se transmita de un cliente a otro por nuestro sistema es necesario pone un escaner de virus en el servidor de disco. Así, se podría poner un script que cheque algún archivo de virus cuando dicho archivo se ha dejado de utilizar.

Otro sistema interesante se presenta en la Fig. 11 y es explicado en detalle en [27]. La idea de este sistema es de usar un escaner de virus para detectar si un correo electrónico viene o no infectado con una virus.

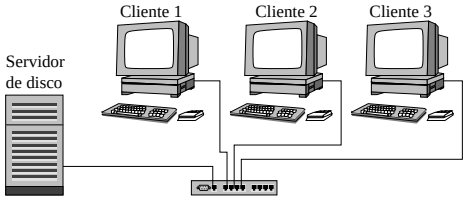

Figura 10: Esquema de un servidor de disco.

Una de las principales puertas de entrada de los virus es a través del correo electrónico recibido por Internet. Este sistema se podría instalar en un cortafuegos o en otra máquina dedicada al sistema. El sistema funciona, en grandes rasgos, de la siguiente manera, el correo que proviene de Internet es guardado primero dentro de un directorio dentro del sistema (el cuadro señalado como *correo entrante* en la Fig. 11). Con un *script* (un programa que es interpretado con instrucciones del sistema operativo, es decir, no debe de compilarse) que se ejecuta cada cinco minutos (por ejemplo) se separa cada correo y los posible archivos que vienen empotrados con ellos; se analizan con el escaner los archivos empotrados (en inglés, *attachments*) y si se detecta un virus, el correo se almacena en una parte del sistema llamado *cuarentena* y se envía un correo de advertencia al que lo envió. Si el correo está libre de virus, se reenvía al servidor de correo normal.

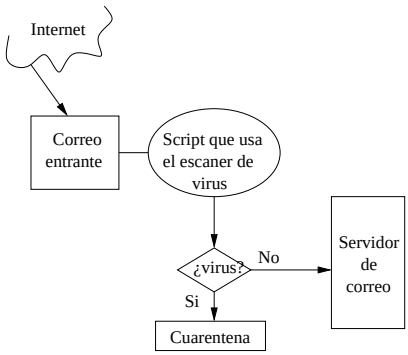

Figura 11: Esquema de un sistema para detección de virus en el correo electrónico.

7. Cortafuegos

Un cortafuegos es una combinación de hardware y software usado para realizar una política de seguridad para controlar el tráfico entre dos o más redes [24], alguna de las cuales puede estar bajo control (ejemplo, la red propia) o fuera de control (ejemplo, Internet). Un cortafuegos de red comúnmente sirve como una primera línea de defensa contra tratamientos dañinos

externos a los sistemas de cómputo propios, redes e información crítica. Los cortafuegos pueden usarse también para particionar las redes internas, reduciendo el riesgo de ataques dentro de la red.

El término de cortafuegos se tomó de la analogía estructural cuyo propósito es hacer más lento el avance del fuego en un edificio. En la literatura de computación [28, 29], prensa popular y vendedores de material en el mercado, el término es usado en muchos sentidos. Algunos lo usan para identificar un componente de hardware específico o un paquete de software, mientras que otros consideran la colección entera de sistemas y software empleados entre dos redes como parte de un cortafuegos.

El término *cortafuegos* generalmente se usa como un adjetivo que modifica a un nombre (tal como sistema, hardware, software ó producto) para hacer la referencia clara. En este trabajo de usará el termino *cortafuegos* como un nombre, cuyo significado general es el concepto de un mecanismo tecnológico para el reforzamiento de la política de seguridad de una red.

Artículos, libros y otras referencias que cubren la evaluación, selección y configuración de tecnologías de cortafuegos son comunes en la actualidad [30, 31, 32, 33].

Los sistemas conectados a Internet son vulnerables a los intentos de acceso no autorizado por parte de usuarios ajenos. Esta práctica suele consistir en intentar introducirse en el sistema, o bien en interceptar información de usuarios remotos conectados al sistema en cuestión, así como modificar información, negar el servicio y abusar de éste. El cortafuegos es un forma de protección contra dichos ataques.

Para construir un cortafuegos primero necesitamos entender como se construyen *listas de acceso*, que serán explicadas a continuación. Para completar un cortafuegos se necesita además levantar una *puerta* (que ya se explicó en capítulos anteriores), así una puerta más listas de acceso nos resulta en un cortafuegos.

7.1. Filtrado de paquetes

Cada servicio que provee un servidor, por ejemplo DNS, WEB, correo electrónico, acceso a través de SSH, etc., se identifica por un *número de puerto*. Así, el DNS va sobre el puerto 53, el WEB en el puerto 80, el correo electrónico sobre el 25 y un servidor de SSH va sobre el puerto 22. La lista de servicios y los puertos asociados a cada servicio pueden consultarse en el archivo `/etc/services` en cualquier distribución de GNU/Linux. Todos los servicios abajo del puerto 1024 se consideran privilegiados y son usados por los principales servicios disponibles en Internet. Puertos arriba del 1024 se consideran no privilegiados y pueden usarse para realizar servicios propios. Los números de puertos asignados a los servicios, que vienen especificados en el archivo `/etc/services`, son de uso estándar; aunque cualquier administrador de red podría asignar otros números de puerto a sus servicios (y sus usuarios deberían de conocer esos números de puertos para poder usar los servicios en puertos no estándar).

Además de los números de puerto, cada servicio va sobre los protocolos TCP y/o UDP. Aquí recordemos que los que se conoce como “TCP/IP” en realidad son un conjunto de protocolos, el básico es IP y sobre de él están TCP, UDP y ICMP. ICMP es el protocolo multicapa que fue diseñado para facilitar el control, prueba y funciones de manejo dentro de una red IP. Las aplicaciones de Internet están sobre los protocolos TCP y UDP. TCP es el protocolo altamente confiable y UDP es simple. TCP es complejo, UDP eficiente y mejor para entregar datagramas. UDP se dice que no es confiable porque no posee ninguno de los mecanismos de confiabilidad de TCP: éste da acuse de recibo por cada datagrama recibido, resecuencia los datagramas recibidos si están fuera de orden y requisa retransmisiones para un paquete recibido que esté dañado. En otras palabras, no se garantiza que un datagrama de UDP alcance intacto su destino.

Para transmitir un paquete de datos, se agregan unos identificadores al inicio del paquete; estos identificadores conforman un *encabezado*. Juntos el encabezado y los datos forman un *datagrama*. El datagrama de IP, en su encabezado, lleva la dirección IP de la fuente y el destino del paquete. El datagrama de TCP, ó de UDP, llevan además el número de puerto del servicio, tanto del puerto fuente como puerto destino.

Ahora bien, el tipo de cortafuegos que viene interconstruido en el núcleo del sistema GNU/Linux es por *filtrado de paquetes*, lo que quiere decir que cada datagrama se bloquea o se deja pasar en base a la información de los encabezados en el datagrama: por dirección IP (fuente o destino) y/ó por el número del puerto del servicio (fuente o destino).

Se le llama *regla* a un línea de texto que describe como se bloquean ciertas direcciones IP y/o servicios. Un conjunto de reglas forman una *lista de acceso*.

7.2. Uso de cortafuegos

Como se ve en la Fig. 12, un cortafuegos se puede usar para proteger una red local. Todas las máquinas, o servidores, dentro de esta red forman una *zona desmilitarizada* (ZDM) [34]. Este cortafuegos puede no tener asignada un número IP, entonces se le conocería como *cortafuegos transparente* dado que no es nece-

sario que se conozca su dirección IP desde Internet.

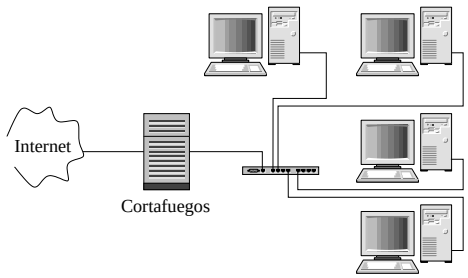

Figura 12: Un cortafuegos usado para proteger una red

Para administrar remotamente un cortafuegos transparente es necesario construir una red alternativa, ya que no se puede accesar en la propia debido a que no tiene dirección IP. Para ello hay que poner otra tarjeta de red en el cortafuegos y en la máquina desde donde se administrará remotamente (esto sería crear otra red, la red más simple, como ya sabemos).

Otro uso de un cortafuegos es el de dividir una red local, creando una red interna con dirección IP no válidas. Todas las máquinas dentro de esta red se lo conoce como *zona militarizada* (ZM). Las máquinas dentro de la zona militarizada pueden acceder a Internet pero ninguna máquina en Internet puede alcanzarlas. Eso hace que los servidores de la red local deban estar dentro de la zona desmilitarizada. El esquema de esta red puede verse en la Fig. 13.

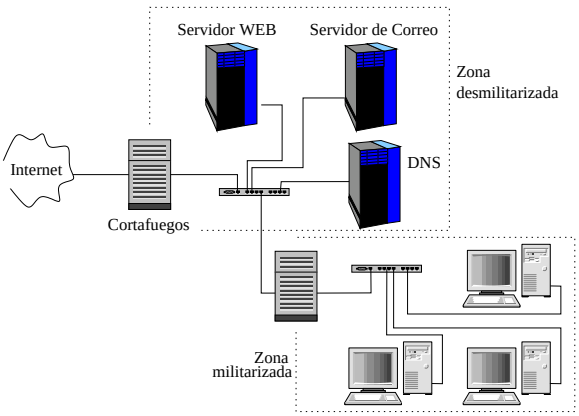

Figura 13: Usos de un cortafuegos. Los servidores dentro de la red ZDM tienen direcciones IP válidas y los clientes en la ZM tienen IP inválidas.

El tener los clientes dentro de una ZM, como se muestra en la Fig. 13 –los clientes tienen direcciones IP inválidas [2]– presenta las siguientes ventajas:

1. Ninguna máquina desde Internet podrá accesar a nuestras máquinas clientes,

2. Los clientes no podrán instalar ningún servidor propio (no será visto desde Internet).

3. Cada dirección IP válida tiene un costo, con este esquema reducimos el número de IP válidas para realizar nuestra red, solo requerimos las IP para el cortafuegos y los servidores.

Este esquema también se lo conoce como Traducción de Direcciones de Red (TDR).

Para realizar una red segura los reglas deben habilitar la salida de los los servicios WEB y ssh (puertos 80 y 22). De esta forma clientes podrán accesar Internet a través de su navegador y a servidores que tengan la comunicación encriptada del SSH. Los clientes también deben tener acceso a los servicios de POP3 e imap para el servidor local de correo electrónico. Cualquier otro servicio debe ser denegado.

Las máquinas que se han usado para los cortafuegos con TDR de la Fig. 13, en la Sección de Computación del CINVESTAV, han sido máquinas con procesador Pentium, de 190 MHz a 300 MHz, y con 64MB en RAM. Esto a sido suficiente para mantener a alrededor de 40 usuarios, aunque no se han realizado pruebas exhaustivas de cuantos usuarios podría sostener una configuración de hardware dada.

7.3. Construcción de un cortafuegos

Se pueden usar tanto *IP-Chains* [16] como *IP-Tables* [35, 36] para la realización de los cortafuegos en un sistema GNU/Linux. IP-Chains es la herramienta para los núcleos en la versión 2.2.x e IP-Tables es la herramienta para las versiones 2.4.x del núcleo de GNU/Linux.

Crear unas lista de acceso, o un conjunto de reglas, para los cortafuegos es un arte. Como tal, se realizan mejor las cosas entre más cortafuegos se configuren y prueben. En [10] se presentan varias configuración para distintos cortafuegos. Esta referencia es un buen punto de partida para aprender a configurar cortafuegos.

Existen también varias herramientas gráficas para la creación de reglas para cortafuegos (checar http://sourceforge.net), aunque están limitadas a ciertos servicios básicos y no pueden usarse en un caso general.

7.4. Protección de una red inalámbrica

Poner un *punto de acceso* (PA) para una red inalámbrica se podría convertir en un gran agujero de

[2] Direcciones IP *inválidas* son las especificadas en el RFC1918 [16, 17, 18] para diseñar redes privadas o intranets, y son las recomendadas para usarse cuando se experimenta con redes. Estas direcciones son 10.∗.∗.∗, 172,16.∗.∗ a 172,31.∗.∗ y 192,168.∗.∗.

seguridad. Un PA es como un etherswitch, donde todos los usuarios de la red se conectan para tener acceso a la red. Aquí el problema es que como no hay cables, cualquiera podría entrar a nuestra red. La zona donde puede tener recepción un PA es de un radio de 100 metros. Si no ponemos seguridad podrían acceder a nuestra red, o usar nuestra plataforma para acceder a Internet, nuestro vecinos, el personal de la empresa de junto, gente en la calle, etc.

La primera barrera de seguridad es habilitar la entrada al PA a las tarjetas de red *registradas*. Esta es una opción de configuración que presentan todos los PAs: se registran las direcciones MAC de cada tarjeta de red inalámbrica de nuestros usuarios. Esta barrera no es cien por ciento segura, algún hacker la podría romper.

La encriptación que viene por defecto en los AP es muy débil. El protocolo de encriptación se le conoce como WEP y ya existen programas en Internet que pueden romperla (podrían adivinar nuestras claves de acceso, por ejemplo).

Se puede usar un cortafuegos para poner nuestras redes inalámbricas y un servidor de autenticación para reconocer a los usuarios de estas redes. Este esquema se lo conoce como *punto caliente* viene desarrollado en [37] y un esquema se muestra en la Fig. 14. En esta referencia realizan el punto caliente con un software libre que le han llamado NoCat.

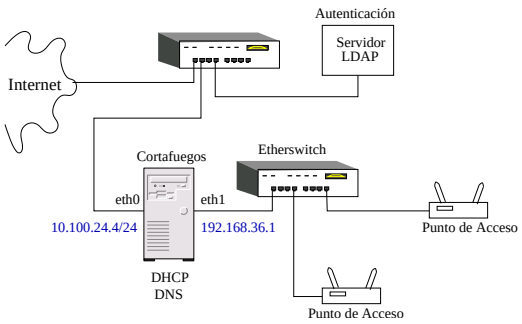

Figura 14: Esquema de un punto caliente realizado con NoCat

Para la realización del punto caliente mostrado en la Fig. 14, se deben instalar un servidor de nombres (DNS) y un DCHP (para asignar direcciones IP dinámicas a los clientes) en la misma máquina del cortafuegos. Además aquí se ha realizado un zona militarizada para todos los usuarios de la red inalámbrica. La dirección IP del cortafuegos en la Fig. 14, la 10.100.24.4/24, es una dirección no válida pero debe suponerse que es una dirección válida (para que sea visto desde Internet). En NoCat puede realizarse de muchas formas el servidor de autenticación, en la Sección de Computación del CINVESTAV se han probado las opciones de tener en archivos simples los usuarios y sus contraseñas y también poner estos datos dentro de un servidor LDAP [38] dentro de nuestro servidor WEB. Se usa SSL ó WTLS para acceder al servidor LDAP, lo que vuelve la comunicación encriptada entre el cortafuegos y el servidor LDAP, por lo que las comunicaciones no pueden ser “escuchadas” por extraños.

8. Monitoreo de la Red

Actualmente no puede verse una red como un equipo estático al que ponemos a funcionar y nos olvidamos de él. Esto es debido a que los ataques a las redes se ha vuelto algo común por lo grande que es la Internet, o aún podemos tener ataques en una intranet por los mismos usuarios. Por estas razones necesitamos contar con una herramienta activa, que nos diga que paquetes están pasando por nuestra red en un instante y tomar las medidas necesarias si la red está sufriendo un ataque.

Un esquema para el monitoreo de la red presentada en la Fig. 13, en la Pág. 23, es mostrado en la Fig. 15. Aquí se aprovecha la situación de los dos cortafuegos para realizar el registro de lo que está pasando por la red.

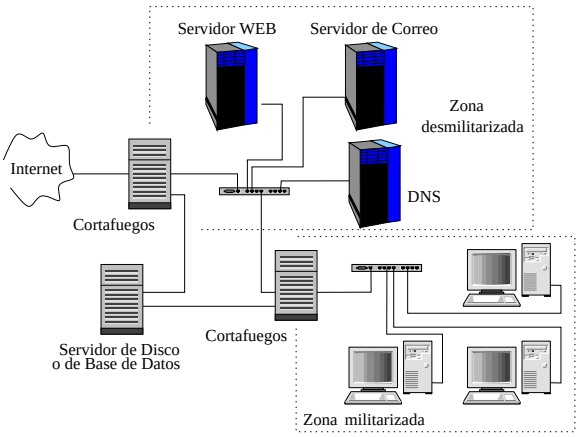

Figura 15: Esquema para realizar el monitoreo de una red con zonas desmilitarizadas y militarizadas: se aprovechan los cortafuegos.

El registro de los paquetes en la red de la Fig. 15 puede realizarse de dos formas:

1. Activando el registro de auditoría para los paquetes que son rechazados en los cortafuegos

2. Activando el registro de todos los paquetes que pasan por la red.

Para la primera opción suponemos que los paquetes que permitimos circular por la red son "buenos". Aunque podría darse el caso, en un ataque que se conoce como *denegación de servicio* que precisamente en un servicio activado se presenten miles de peticiones, lo que congestiona de tráfico la red y la vuelve al mismo tiempo invisibles (por no estar disponible) a todos. Así, con la primera opción tenemos el registro de cosas que están pasando por la red y que no están permitidas. En el segundo esquema podemos llevar estadísticas de todos los paquetes que están circulando por nuestra red, aunque no podemos ver su contenido. Para llevar el registro del contenido de los datagramas podría usarse una tercera opción que sería instalar un paquete como *snort* [10] en los cortafuegos, esto podría hacer viable el análisis de los mismo datos y no solo la cantidad de ellos; aunque esta última solución sería gran demandante de recursos de cómputo. Un esquema que desarrolla esta tercera opción ha sido presentado en [39].

En ambas soluciones todavía es necesario contar con alguna herramienta que analice los datos y nos lo presente de forma amigable, tal vez en una página HTML en nuestro servidor WEB.

El esquema de monitoreo podría extenderse para que detectase automáticamente los abusos y mal funcionamiento de la red. Esta es aún un área activa de investigación en la que ya se tienen resultados parciales [39], aunque no definitivos.

9. Deshabilitar los Servicios y Características Innecesarios

Tener un servicio o programa que no se usa para nada importante es algo a lo que no se le da mantenimiento, o no se toma en cuenta, y entonces puede ser un objetivo de ataque desde Internet o aún desde nuestra propia intranet. En una máquina con el sistema GNU/Linux hay que borrar las ligas correspondientes en el directorio de los niveles de ejecución en `/etc/rc.d/`. Esto puede hacerse más sencillo usando el comando `chkconfig` que viene en la distribución de RedHat y sirve para administrar esas ligas; por ejemplo, si no se requiere el servidor de DNS, hay que realizar algo como

```
/sbin/chkconfig --del named
```

y esto nos evita checar todas las ligas en los subdirectorios dentro de `/etc/rc.d/`. Y claro hay que detener la ejecución del servicio actual, para el ejemplo se haría

```
/etc/rc.d/init.d/named stop
```

Para checar que se dejó de ejecutar debemos revisar los procesos que actualmente de están ejecutando

```
ps -ef | more
```

y finalmente, para checar el estado de los servicios que se levantar al arranque del sistema se usa el comando:

```
/sbin/chkconfig --list | more
```

Todos los servicios innecesarios deberían ser deshabilitados. En particular hay que poner especial atención en los siguientes:

- Servicios RPC: En los protocolos de Sun para Control de Procedimientos Remotos (en inglés, RPC, Remote Procedure Control) el cual esta incluido sobre virtualmente todos los sabores de UNIX, que incluye Linux, por supuesto. Estos servicios nos permiten ejecutar comandos sobre un sistema remotos vía rsh, rcp, rlogin, nfs, etc. Desafortunadamente no es un protocolo muy seguro, especialmente sobre huéspedes en una ZDM. No se deberían de ofrecer estos servicios al mundo externo. Las funcionalidades de este servicio puede ofrecerlas ssh (el Shell Seguro), el cual ha sido especialmente diseñado para reemplazar a los servicios rpc). Deben de deshabilitarse los servicios nfsd y nfsclientd con el comando `chkconfig` y comentar cualquier comandor, o de plano deshabilitar todos los servicios de xinetd, usando el mismo comando `chkconfig`. Algunos procesos locales pueden que usen el servicio rpc de portmap, por lo que hay que deshabilitarlo con cuidado: se rehabilita si algo no funciona correctamente.

- linuxconfd: Aunque no hay bugs explotables en la versión actual de linuxconf (esta es una herramienta de administración que puede accesarse remotamente), el CERT reporta que este servicio es escaneado comúnmente y puede ser usado por los atacantes para identificar sistemas con otras vulnerabilidades (ver CERT Current Activity page 7/31/2000, http://www.cert.-org/current/current_activity.html.

- sendmail: Muchas gente cree que el servicio de envío de correo, el cual está habilitado por defecto en las mayoría de las versiones de UNIX, es necesario aún en las computadoras que solo envían correo a ellas mismas (p.e., mensajes administrativos tales como los que genera la salida del Ctrl+tab al root que los envía el demonio del crontab). Esto no es así: el sendmail (o postfix, qmail, etc.) solo es necesario en ls huéspedes que entregan correo o reciben correo desde otros huéspedes.

- Telnet, FTP y POP: Estos tres protocolos tienen una característica riesgosa en común: requieren que los usuarios envíen su nombre de usuario y

su contraseña en texto claro sobre la red. Los servicios de telnet y FTP son reemplazados fácilmente con ssh y su utilidad scp para transferir archivos; el correo electrónico puede ser redirigido automáticamente o otro huésped, puede dejarse en el servidor dentro de la ZDM y leído a través de una sesión de ssh, ó puede descargarse vía POP usando una "redirección local" a ssh (esto es, puesto en una pipa a través de una sesión encriptada del Shell Seguro). Estos tres servicios son invocados por xinetd.

9.1. Ejecutar los servicios en una raíz diferente cuando sea posible

Algunos demonios (*dæmons* en inglés) tales como named, tienen un soporte específico para ejecutarse en otra raíz con el comando chroot. Esto significa que el proceso ejecutado en otra raíz usa "/" en algún otro subdirectorio y solo puede moverse dentro de él. Esta es una característica invaluable para la seguridad, si un proceso ejecutado en otra raíz es atacado o explotado de alguna manera, el atacante no será capaz de acceder a archivos fuera del subdirectorio de la nueva raíz.

En Linux, aun los procesos que no tienen un soporte interconstruido para ejecutarse en otra raíz pueden hacerlo. Por ejemplo, para ejecutar un comando imaginario `burbuja -v pciclo` en la nueva raíz `/var/burbuja`, se hace

```
chroot /var/burbuja \
 /usr/local/bin/burbuja -v pciclo
```

Nótese que, sin embargo, cualquier archivo del sistema, que necesite el proceso para que puede ser ejecutado, debe ser copia a los subdirectorios apropiados dentro de la nueva raíz. Si nuestro proceso imaginario necesita leer el archivo `/etc/passwd`, necesitamos poner una copia de este archivo dentro de `/var/burbuja/etc`. La copia no necesita contener más de la información que el proceso necesita; para el ejemplo, si `burbuja` es un servidor que solo usuarios anónimos pueden acceder, el archivo `/var/burbuja/etc/passwd` probablemente solo necesita una línea como está

```
nobody::50:50:Anonymous user::/bin/noshell
```

9.2. Ejecutar los servicios con UIDs y GIDs no privilegiadas cuando sea posible

Anteriormente los demonios solo podían trabajar si se ejecutaban como root (que es el UID [la identificación de usuario] por defecto de un proceso invocado al arranque), pero ahora muchos programa pueden ejecutarse como un usuarios no privilegiado. Por ejemplo, Postfix, el reemplazo de sendmail de Wietse Venem, usualmente se ejecuta con una cuenta especial sin privilegios de nombre postfix.

Esta característica tiene un efecto similar a chroot (y de hecho los dos se usan juntos frecuentemente). Si el proceso es atacado, el atacante obtendrá los privilegios en el nivel de acceso menores que root (y tal vez mucho menores). Se debe de tener cuidado que la cuenta sin privilegios todavía tenga los privilegios para poder hacer su trabajo.

9.3. Borrar las cuentas innecesarias

Algunas distribuciones de Linux, por defecto, presentan archivos largos de `/etc/passwd` que contienen cuentas para paquetes que no se han instalado. Se deben de comentar o borrar tales cuentas, especialmente las que incluyen shells ejecutables.

9.4. Configurar los archivos de auditoría y checarlos regularmente

Esta configuración es algo que todos deberían saber pero que frecuentemente se olvida seguir. No se puede checar auditoría que no existe y no se puede aprender algo de esos archivos si no se leen. El servicio de archivos de auditoría (*logging*) debe ser puesto en un nivel apropiado, hay que conocer donde se localizan estos archivos y como son rotados cuando son muy grandes y hay que aprender el hábito de checarlos continuamente para encontrar anormalidades.

Para analizar los archivos de auditoría contamos con el comando `grep`, que busca patrones en el texto y los imprime. O también el comando `diff` para checar diferentes versiones de archivos de configuración.

Si se tienen varios servidores en una ZDM, se puede considerar levantar los servicios de syslogd en cada servidor, o se podría levantar un solo demonio del syslogd. El demonio de syslogd puede configurarse para recibir no solo la salida de todos los procesos del sistema, sino también para recibir los datos de huéspedes remotos. Por ejemplo, si se tienen los huéspedes tintan y rollo dentro de la ZDM y se desean tener los archivos de auditoría de los huéspedes en un solo lugar, entonces se debe cambiar el archivo /etc/syslogd.conf de tintan para que contenga solamente esta línea:

```
*.*         @rollo
```

Esto provocará que el syslogd de tintan envíe sus datos no a su archivo `/var/log/messages` sino al del huésped rollo.

Esta solución también implica riesgos: si rollo ha sido atacado, no se podrán tener los registros de tin-

tan, o un atacante podría aprender lo suficiente de tintan leyendo los archivos de auditoría, y posteriormente atacarlo. Debe de sopesarse si los beneficios justifican la exposición (o cual configuración tiene más beneficios para proteger de forma efectiva nuestra red).

Referencias

[1] Jerry Honeycutt. Using internet. http://docs.rinet.ru/UsingInternet/ch01/ch01.htm.

[2] José Antonio Millán. El fruto caliente de guerra fría. noviembre 1999. http://jamillan.com/histoint.htm.

[3] Corporación Universitaria para el Desarrollo de Internet (cudi). www.internet2.edu.mx.

[4] Xavier Cufí. *Internet, Protocolos y Servicios*. http://eia.udg.es/˜atm/tcp-ip/tema_4_1_1.htm.

[5] Bruce Sterling. Pequeña historia de internet. 1992. http://www.sindominio.net/biblio/-web/telematica/hist/internet.html.

[6] Internet pioneers: Paul baran. http://www.ibiblio.org/pioneers/baran.html.

[7] John Plane Jr. Tcp/ip introduction and history. http://www.certificationcenter.net/php-web/tcphistory.php.

[8] Christos J.P. *History of Internet. A chronology: 1843 to the present*. Moschovite Group, 2001. http://www.historyoftheinternet.com/index.html.

[9] Parker and M.Sportack. 2000.

[10] G. Mourani. *Securing & Optimizing Linux: The Ultimate Solution*. July 2002. Disponible en http://www.tldp.org.

[11] O. Kirch and T. Dawson. *Linux Network Administrators Guide*. O'Reilly, 2000.

[12] Redes-en-linux-como. Disponible en www.tldp.org.

[13] R. Hart. Ip sub-networking mini-howto, 2001. Disponible en www.tldp.org.

[14] El sitio oficial de la distribución GNU/Linux de RedHat. Disponible en www.redhat.com.

[15] Inc. Red Hat. Hardware autodetection & configuration tool. http://fedora.redhat.com/projects/additional-projects/kudzu/.

[16] The Linux Documentation Project. *IPCHAINS–HOWTO*.

[17] www.tldp.org. *IP–Masquerade–HOWTO*.

[18] J.D. Blair and L. Grinzo. Connected to the net. *Linux Magazine*, 2(5):50–59, 2000. www.linux-mag.com.

[19] L. Teo. Setting up a linux gateway. *Linux Journal*, (72):86–88, April 2000. www.linuxjournal.com.

[20] P.F. Crow. The linux home network. *Linux Journal*, (72):80–84, April 2000.

[21] C. Easwaran. Linux apprentice: A heterogeneous linux/windows 95 home network. *Linux Journal*, (76):62–67, August 2000.

[22] D. Coulson. Mastering iptables. May 2001. www.linuxformat.co.uk.

[23] El Centro de Coordinación CERT para la seguridad en Internet. http://www.cert.org.

[24] A. Householder, A. Manion, L. Pesante, G.M. Weaver, and R. Thomas. Managing the threat of denial-of-service attacks. v10.0. *CERT® Coordination Center*, Oct 2001. www.cert.org.

[25] Software-RAID-HOWTO. Available in http://www.tldp.org.

[26] Virus-l/comp.virus frequently asked questions (faq) v2.00. http://www.faqs.org/faqs/computer-virus/faq.

[27] D. Jones. Building an e-mail virus detection system for your network. *Linux Journal*, pages 56–65, dec 2001.

[28] Andrew S. Tanenbaum. *Redes de Computadoras*. Prentice Hall Hispanoamericana, 1999.

[29] D. Comer. *Operating System Design, Vol II*. Prentice Hall, 2001.

[30] Marcus Goncalves. *Manual de Firewalls*. McGraw-Hill, 2001.

[31] Robert L. Ziegler. *Firewalls Linux , Guía Avanzada*. Prentice Hall, 2000.

[32] D.B. Chapman and E.D. Zwicky. *Construya Firewalls para Internet*. O´Really Associates, Inc, 1997.

[33] The Linux community's center for security. http://www.linuxsecurity.com.

[34] M. Bauer. Designing and using DMZ networks to protect internet servers. *Linux Journal*, (83):27–36, March 2001.

[35] D.Napier. Iptables/netfilter – linux's next-generation stateful packet filter. *SysAdmin*, 10(1), Dec 2001. www.samag.com.

[36] M. Bauer. Paranoid penguin: Using iptables for local security. *Linux Journal*, 2002.

[37] M. Kershaw. Linux-powered wireless hot spots. *Linux Journal*, (113), Sep 2003.

[38] Information about installing, configuring, running and maintaining a LDAP (Lightweight Directory Access Protocol) server on a linux machine. http://www.tldp.org/HOWTO/LDAP-HOWTO/.

[39] J. E. Morfín Galván. Análisis de tráfico en una LAN. Master's thesis, CINVESTAV, Marzo 2004.