Instalación y Configuración de Servidor INTERNET E INTRANET

Junio-2000

DESCRIPCION DEL CURSO. PROGRAMA.

Este curso está **dirigido** a :

- Directores de Informática y Jefes de Proyecto, o a personas que realicen funciones equivalentes dentro del mundo empresarial.
- Investigadores y Técnicos.
- Profesionales.
- Personas interesadas en el mundo de las Comunicaciones entre ordenadores y en Internet.

Describe la forma de instalar un **Servidor Internet** e **Intranet** con ejemplos para un PC con sistemas operativos Linux (UNIX) o Windows NT. En cada capítulo se expone la forma de hacerlo por cada **servicio**.

Para seguir este curso, es conveniente tener unos conocimientos básicos de Internet y sus servicios. También se recomienda tener conocimientos básicos de UNIX y de Windows NT.

La **duración** estimada es de 32 horas incluyendo teoría y prácticas. El **contenido** consta de los temas y duración siguientes:

```
1.- Introducción y Objetivos.                          [15 minutos]

Práctica de instalación de Linux.                             [1 hora ]
Práctica de configuración de servidor Linux.           [1 hora ]
Práctica de configuración de servicios en Linux.              [2 horas ]
Práctica de uso de servidor y servicios desde cliente.        [2 horas ]
```

INDICE

1. INTRODUCCION.

El impacto mundial de la red Internet es extraordinario, y se extiende de una forma espectacular en todo el mundo, revolucionando las comunicaciones con sus avanzados servicios. Las características de esta red que propiciarion su éxito se resumen en lo siguiente:

- **Fuente de información** con posibilidades ilimitadas. En ella se puede encontrar información continuamente actualizada sobre cualquier tema (ciencia, periodismo, comercio).
- **Medio de comunicación global**. Permite poner en contacto dos usuarios desde cualquier punto del mundo, de forma inmediata y sencilla (diálogos entre usuarios, videoconferencia, aplicaciones de ofertas, pedidos, consultas). Y se puede acceder a la red desde casa.
- Su **manejo** y **navegación** es muy sencilla y potente. Permite cambiar de forma inmediata de un punto a otro del globo, aunque el usuario solo necesita conocer el primer punto (mediante referencias entre distintos puntos).
- Posibilidad casi ilimitada de creación de **nuevos servicios**. Dado que utiliza un protocolo (TCP/IP) que diferencia los mensajes por servicios (puertos), y permite crear nuevos servicios sin cambiar de red, ni de equipos, ni de protocolo. En la actualidad ya están a disposición nuevos productos y **servicios** como:
  . Información tecnológica, financiera, metereológica, negocios, cultural, ocio.
  . Productos informáticos, financieros, y de cualquier otro tipo.
  . Congresos, Ferias.

Es de **gran utilidad** para:
- Empresarios que muestran sus productos en la red, realizan pedidos a proveedores y recogen información sobre nuevos productos de proveedores, y de la competencia.
- Técnicos, científicos e investigadores que deseen conocer información sobre un tema determinado, o poner a disposición de los demás los resultados propios.

Debido a su importancia, no podemos quedarnos al margen sin conocer esta tecnología de comunicaciones tan potente. Por ello surge este curso que describe la forma de instalar y configurar un **ordenador PC** como **servidor** para ofrecer servicios propios a la red Internet, o como servidor privado Intranet (sin conexión a la red mundial).

La diferencia entre **Internet** e **Intranet** estriba en que ésta última no está conectada a la red mundial, por ello los servicios solo son accedidos por un conjunto de usuarios (clientes) local. Sin embargo, todos los demás aspectos son los mismos.

1.1. Historia de Internet.

Debido a la proliferación de redes y a la **necesidad** de **compartición** de los **recursos** del **Departamento de Defensa de U.S.A.** surge la necesidad de interconectarlos mediante una **red**. Las primeras redes no podÍan alcanzar grandes dimensiones, ni grandes distancias y eran incompatibles, debido a esto surge el concepto de **WAN** (**Red de Area Extensa**).

El Ministerio de Defensa de Estados Unidos dispone en **1969** de una **red de 4 nodos**, con la que se conectan ordenadores de 4 puntos distantes de su geografÍa. Era el proyecto ARPA, y la red **ARPAnet**. Esta constituye la primera experiencia de conexión de ordenadores con posibilidad de **reencaminamiento** de mensajes. Nace el concepto de '**internetworking**'. Para estas conexiones se crea el protocolo **IP** (Internet Protocol), base para la posterior red **INTERNET**. Este protocolo consiste en enviar los datos en paquetes de mensajes con las direcciones del ordenador origen y destino, de tal forma que pueden viajar por toda la red, por múltiples caminos (reencaminamiento -routing-).

En **1973** se realiza la primera **interconexión internacional** entre Noruega y el Reino Unido.

En **1982** se crea **EUnet**, la red europea que adopta el protocolo **TCP/IP** (TCP es 'Transmission Control Protocol') utilizando ordenadores con el sistema operativo **UNIX**. El protocolo TCP define el formato de los mensajes que viajan dentro de los paquetes IP.

En **1986** nace la red **NSFNet** (NSF es National Science Foundation) para unir los ordenadores de Universidades y organismos dedicados a la investigación científica. Al principio utiliza la misma red ARPAnet, pero posteriormente crea sus propios enlaces formando la red NFSnet mediante enlaces telefónicos de **56.000 bps** (bits por segundo).

En **1987** llega a España la red EUnet a traves de la **Red IRIS**. Y en este mismo año IBM, MCI y Merit mejoran los enlaces de NFSnet, que ya se había saturado. En **1990** empieza el servicio de Internet nativo en **España**.

En **1991** comienza el **uso comercial** de Internet en **Estados Unidos**, con la **red ANS** (Advanced Netwoorks and Services) formada por IBM, Merit y MCI. En **1992** comienza el uso comercial de Internet (**EUnet**) en **España** mediante Goya como proveedor.

En **1994** existen de **20.000 a 50.000 redes** conectadas a Internet de las cuales 300 a 400 están en España. De **3 a 6 millones de ordenadores** tienen acceso, de ellos 38.000 están en España (en **Junio** de **1995** existían **30 millones** de ordenadores conectados a Internet, 250.000 en España). De **20 a 40 millones de usuarios** existen en el mundo (250.000 en España). Internet llega a más de **90 paises**, con enlaces de **90 Mbps**.

1.2. Estado Actual y Evolución.

El **crecimiento** de la red es extraordinario. Cada mes crece un 10% el número de redes, proveedores y ordenadores conectados a la red. De la misma forma el impacto tanto en la empresa como en la sociedad en general es cada vez mayor, merced a los nuevos servicios que aporta la red.
En Junio de 1995 existían 30 millones de ordenadores conectados o con acceso a la red Internet (250.000 en España), y 20.000 compañias.

El número máximo de ordenadores que pueden conectarse a la red son 4.294.967.286 ordenadores (igual al número máximo de direcciones IP).

Simultáneamente a Internet fueron creciendo otras redes con **otros protocolos** que siguen dando **servicio** a un **gran número de usuarios**. Para que estos usuarios no queden aislados de la red Internet las compañías que gestionan estas redes permiten el **acceso** de sus usuarios a los servicios de **Internet** y que los usuarios de Internet puedan comunicarse con sus usuarios. Esta función la realizan mediante ordenadores conectados a ambas redes (Internet y la red propia) denominados **pasarelas** o **puertas de acceso** ('**gateway**').

Estas redes son:

- BITNET
- Servicom
- Infovía
- Compuserve
- Fidonet
- America OnLine

En el caso de la red española **Infovía** (que utiliza el mismo protocolo que Internet), se ofrecen dos formas de acceso a los usuarios, que permiten distintas facilidades:

1. **Acceso No Identificado**. Solo permite acceder a los Proveedores de Servicios Infovía, no a Internet. Este acceso es gratuito y solo hay que pagar la llamada telefónica (055). Este servicio es similar al Videotext, y además lo sustituirá en un futuro.

2.- **Acceso Identificado**. Permite el acceso a los Proveedores Infovía y a Internet. Para tener este acceso es necesario disponer de una Cuenta de Usuario en un Proveedor Infovía. Y es este proveedor que funcionará de pasarela entre Infovía e Internet. Al coste de la llamada telefónica (055) hay que añadir el coste de la Cuenta de Usuario del Proveedor.

1.3. Tipos de Conexión.

Internet y/o Infovía.
Intranet

1.4. Oferta de servicios en España.

Telefónica Transmisión de Datos
Global One (Sprint y France Telecom)
BT Telecomunicaciones

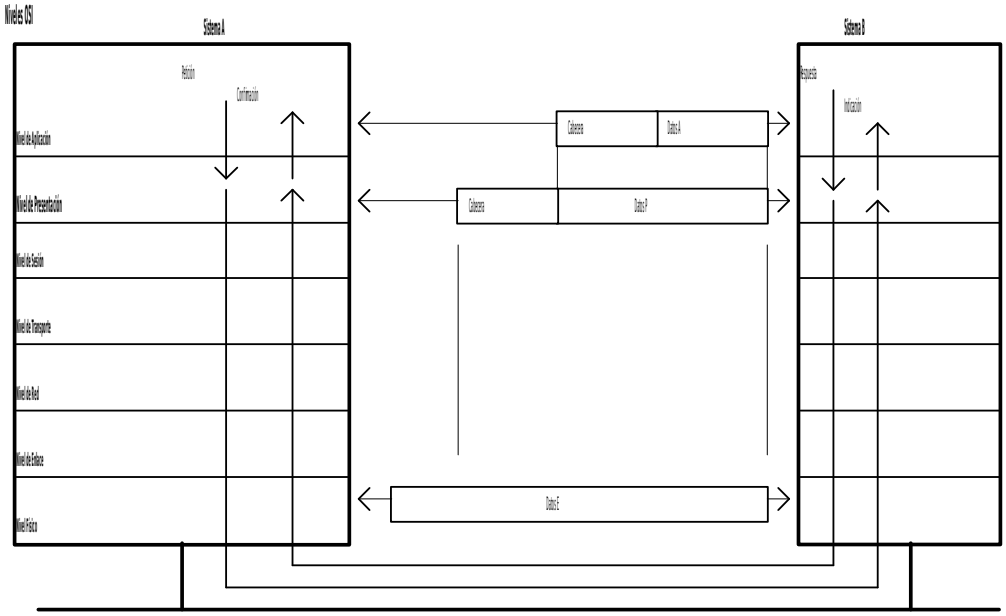
2. Familia de protocolos TCP/IP.

2.1. Introducción al modelo OSI.

El modelo OSI (Open Systems Interconnection) promovido por el organismo ISO está basado en niveles lógicos o funcionales. Cada nivel ofrece servicios al de nivel superior en los puntos de acceso al servicio (SAP).

El modelo se define mediante documentos para cada nivel, que incluyen dos partes:
- **Servicio**, definido según unas primitivas del servicio (Conexión, Desconexión, Transferencia de datos)
- **Protocolo**, especificación del formato de los paquetes para dar ese servicio.

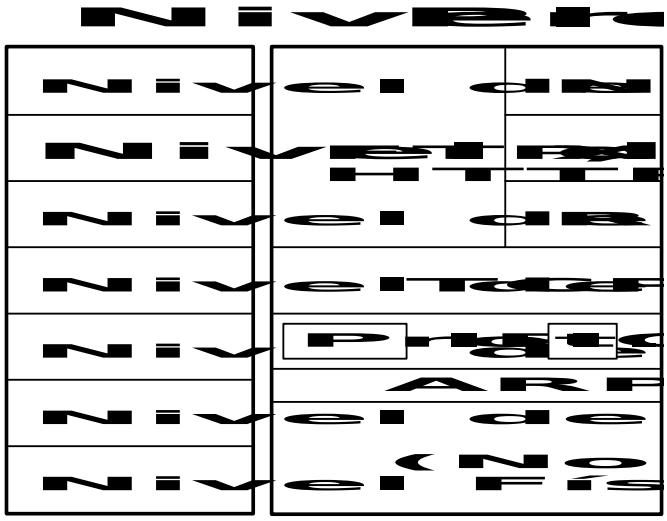
Se implementan mediante protocolos entre entidades del mismo nivel. Cada nivel introduce sus cabeceras para implementar el protocolo.



2.2. Protocolos Internet.

A mediados de los 70 DARPA (Defense Advanced Research Projects Agency) estableción una red de conmutación de paquetes para comunicar centros de investigación en los Estados Unidos.

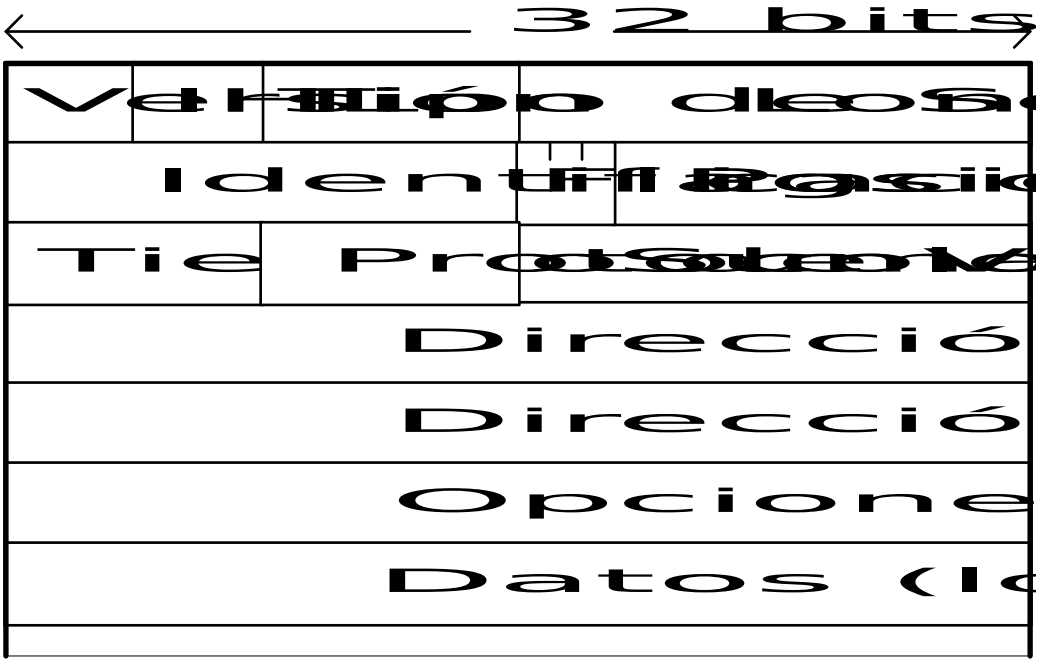
DARPA y otras organizaciones gubernamentales pensaron que era la forma de conectar centros con redes heterogéneas. Con este objetivo DARPA financió la investegación de una serie de protocolos de comunicaciones, desarrollados en la Universidad de Stanford. El resultado fue el conjunto de protocolos Internet a finales de los 70, de los que los más conocidos son TCP e IP. El conjunto se conoce como Internet Protocol Suite. La característica más sobresaliente de estos protocolos es que puden funcionar sobre cualquier tipo de redes, tanto LAN como WAN. Incluye no sólo especificaciones de bajo nivel (TCP e IP), sino también especificaciones para aplicaciones estándar, como mail, emulación de terminal y transferencia de ficheros.



Los protocolos se especifican en documentos llamados RFCs (Request for Comments). Las RFCs se publican y después son estudiadas por la comunidad Internet. Después son revisadas y publicadas como nuevas RFCs.

2.2.1. Nivel de red.

IP es principal protocolo de nivel 3. Proporciona fragmentación y reensamblado de datagramas e indicación de errores. El formato de paquetes de **IP** se muestra a continuación:



Los campos son los siguientes:

- Version. La versión IP que se usa (la actual es la 4)
- IP header length (IHL). Longitud de la cabecera del datagrama en palabras de 32 bits (lo habitual es 20 octetos => 5)
- Type-of-service. Cómo quiere el quiere el protocolo de nivel superior que IP manege los datagramas. Incluye una serie de grupos de bits:
  - PRECEDENCE del datagrama(0-7), suele ignorarse por routers y hosts
  - D bit (low delay)
  - T bit (high throughput)
  - R bit (high reliability)
  - La red subyacente puede no garantizarlos
  - Total length. Longutud total del paquete (cabecera + datos) en bytes. En la actualidad máximo son 65535 (16 bits)

Los 3 campos siguientes se refieren a la fragmentación de datagramas Las redes subyacentes tienen distintas MTU (maximun transfer unit), por ejemplo Ethernet 1500 bytes, Token Ring 4470 bytes, o incluso menores, por lo que los datagramas se fragmentan en origen, y routers intermedios, y se emsamblan en el destino.

- Identification. Identifica a un datagrama, se usa para saber que los fragmentos que llegan pertenecen al mismo datagrama

- Flags. 3 bits, los 2 inferiores controlan la fragmentación:
  - No not fragment

  - More fragments (a 0 si es el último) Es necesario porque el campo Total length se refiere a la longitud del fragmento, no del datagrama original.
- Fragment offset. Offset del fragmento actual respecto al datagrama original

Otros campos:

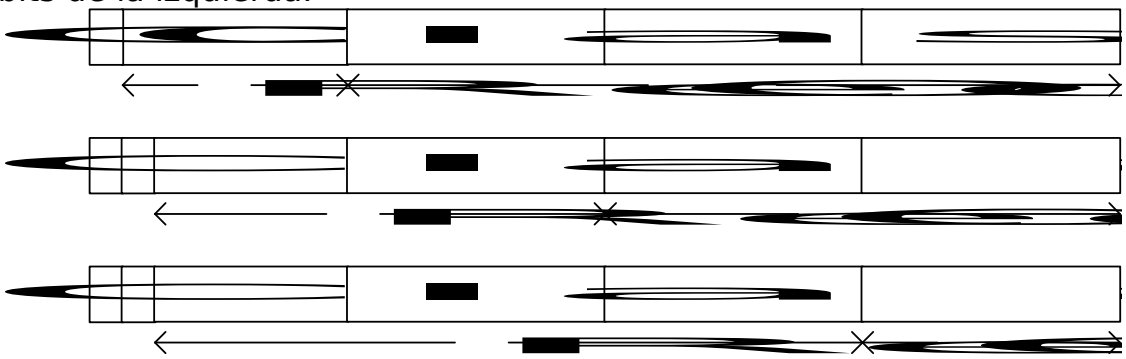
- Time-to-live. En segundos, se va decrementando gradualmente hasta llegar a 0, y entonces el datagrama se descarta. Los routers lo decrementan cuando un datagrama pasa por ellos (1 en caso normal, o más si hay congestión)
- Protocol. Indica cuál es el protocolo de nivel superior que creó el datagrama.
  - TCP
  - UDP
  - ICMP
- Header checksum. Para integridad de la cabecera (no de los datos)
- Source address
- Destination address
- Options. Generalmente no se usan, sólo para depuración y test.
- Data. Contiene datos de protocolos de nivel superior

2.2.2. Direcciones.

El esquema de direccionamiento es esencial para el routing de los datagramas IP en Internet. Una dirección IP tiene una longitud de 32 bits, dividida en 2 ó 3 partes:

- Network Address
- Subnet Address (opcional, sólo si el administrador lo decide)
- Host Address

Las longitudes de cada parte son variables. El direccionamiento IP soporta 5 clases diferentes de redes (tipos). La clase está determinada por los bits de la izquierda.



Class A

- 7 bits para la red (126 redes, la 127 está reservada)
- Para compañías muy grandes
- Imposibles de obtener

Class B

- 14 bits para la red
- 16 bits para hosts (~65000)

Class C

- 22 bits para la red
- 8 bits para hosts (254)

Class D

- Los 4 bits de mayor orden son 1110
- Clase especial reservada para multicast
- Descrita en la RFC 1112

Class E

- Los 4 bits de mayor orden son 1111
- Reservada para uso futuro

Hay algunos números que significan cosas especiales, en general:

- 1s significa “todos”, se usa para broadcast
- 0s significa “este”, se suele usar en el arranque cuando no se conoce su propia dirección
- 0s también se suele usar para referirse a una red

En particular:

0s		este host (arranque)
0s	host	host en esta red (arranque)
1s		broadcast en red local

net	1s	broadcast en red
127	-	interfaz loopback

Las direcciones IP se suelen escribir como 4 números decimales separadas por puntos: a.b.c.d. Las redes IP se pueden dividir en unidades más pequeñas, llamadas subredes (subnets). Esto se hace tomando parte de los bits de la porción del host, que ahora se usan como campo de subred.

El número de bits que se emplean para la subred es variable, y para especificarlo se emplea la máscara de subred. Las máscaras de subred usan el mismo formato que las direcciones IP, y tienen todos sus bits a 1s excepto aquellos que se refieren al campo del host.

Por ejemplo, para la red de clase A 34.0.0.0 si la máscara de subred es 255.255.0.0 entonces tenemos un campo de subred de 8 bits.

Si usamos como máscara de red 255.255.255.0, tendríamos 16 bits para el campo de subred.

2.2.3. ICMP

Encapsulado dentro de datagramas IP (datagram protocol = 1).

Para encaminar los datagramas IP, se hace salto a salto. El camino completo no se conoce a priori. En cada punto intermedio, se calcula el siguiente salto comparando la dirección de destino con la tabla de rutas de ese punto.

Sin embargo, si un datagrama no llega a su destino, IP no provee información de error hacia el origen. Esta tarea le corresponde a otro protocolo: Internet Control Message Protocol (ICMP)

ICMP realiza una serie de tareas dentro de una red Internet. La principal es reportar errores hacia el origen de un datagrama. Es sobre todo usado por los routers.

Además sirve para lo siguiente:

-Echo & reply messages: para tests de alcance
- Redirect messages: para conseguir rutas más eficaces
- Time exceeded mesagges: para informar de que un datagrama ha expirado
- Router advertisement & router solicitation mesagges: para routers directamente conectados a subredes
- Mecanismos para descubrir la máscara de subred usada.

2.2.4. ARP y RARP

Estos protocolos sirven para establecer la correspondencia entre direcciones IP y direcciones de acceso al medio. Se usan sobre determinadas redes que sirven de base a IP (por ejemplo en las LAN 802.x)

Address Resolution Protocol (ARP)

Usa mensajes de broadcast para determinar la dirección MAC (Media Access Control) que corresponde a una dirección IP. El host que se da por aludido devuelve su dirección MAC. Permite ser usado para cualquier tecnología subyacente.

Se construye una tabla con pares dir IP - dir MAC.

Reverse Address Resolution Protocol (RARP)

Usa broadcast preguntando por la dirección IP que le corresponde a una dirección MAC.

Lo usaban las estaciones sin disco para arranque para conocer su dirección IP, pero actualmente está en desuso (por ejemplo se usa BOOTP o DHCP).

2.2.5. Nivel de Transporte

Se implementa con los protocolos TCP y UDP. TCP provee un tipo de servicio orientado a conexión, mientras que UDP es “conectionless”

2.2.5.1. Transmission Control Protocol (TCP)

Da un mecanismo de transferencia de información fiable. Los datagramas IP pueden:

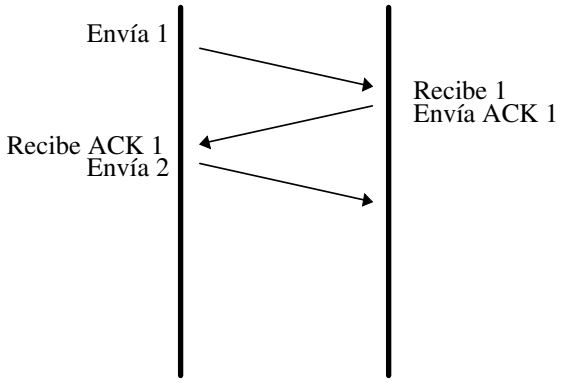
- perderse
- destruirse
- expirar
- corromperse
- duplicarse
- llegar fuera de orden

TCP evita los problemas de los datagramas IP, debido a que cumple los siguientes requisitos:

- Orientado a streams: cuando dos aplicaciones transfieren datos, éstos son recibidos como un stream de bits exactamente igual al que se ha enviado.
- Orientado a conexión: antes de empezar a transferir datos, se debe establecer una conexión. Se establece un mecanismo por el que los sistemas finales aceptan establecer un mecanismo de circuito virtual.
- Buffer de transferencia: los datos son insertados en un extremo por la aplicación, y en el otro se reciben en la misma secuencia.
- Stream no estructurado: las aplicaciones extremo se deben poner de acuerdo sobre el contenido de los datos, el servicio no provee mecanismos para marcar límites (existen mecanismos adicionales, como XDR)
- Conexión Full-Dúplex

Fiabilidad

Para lograr la fiabilidad, se utilizan mecanismos de asentimiento de paquetes (Acknowledge). El mecanismo básico es el siguiente:

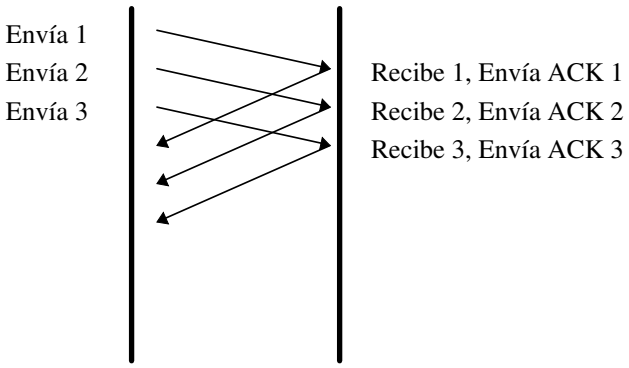


El que envía mantiene registro de los paquetes enviados e inicia un timer, si no se recibe el ACK en tiempo, reenvía el paquete. Espera un ACK antes de mandar el siguiente.

Desventajas:

- Puede haber duplicados
- Si hay retardos de transmisión es muy lento
- Sólo en una dirección

Para solucionar estos inconvenientes se usa un mecanismo de *sliding window*:



Existe una ventana de transmisión que incluye varios paquetes. Se pueden enviar paquetes que estén dentro de la ventana actual. Cuando se reciben los ACKs la ventana se desplaza.

Mecanismo bidireccional (piggybacking), los paquetes de datos contienen los ACKs. Con un tamaño de ventana grande, se eliminan los retardos de transmisión.

Conceptos TCP

Especifica el formato para datos y ACKs.

En TCP se utiliza el concepto de port, que se pueden ver como distintas colas donde el software que implementa el protocolo va dejando los paquetes.

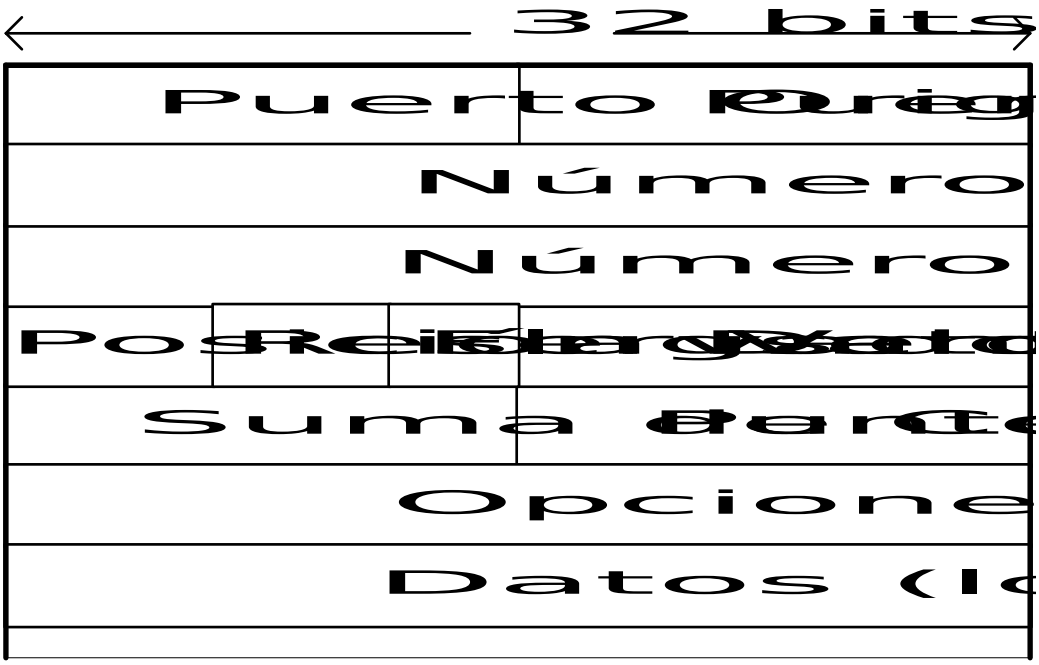
Los ports están identificados por un entero (0 - 65535).

Por ser TCP orientado a conexión, se debe establecer primero una conexión entre dos puntos finales de conexión:

(host, port)

Estructura del paquete TCP

La estructura de los paquetes TCP está alineasa en palabras de 32 bits, y es la siguiente:



- Source port & destination port: identifica los ports de los puntos origen y destino de la conexión.
- Sequence number: TCP en vez de numerar paquetes, numera bytes. Este número identifica el primer byte de datos de este paquete.
- Acknowledgment number: número de secuencia del siguiente byte que se espera recibir (por tanto acepta los bytes hasta *acknowledgment number - 1*)
- Data offset: indica el número de palabras de 32 bits que contiene la cabecera TCP.
- Reserved: para uso futuro.
- Flags: información diversa del protocolo.
- Window: especifica el tamaño de la ventana de recepción del que envía, es decir, el espacio disponible para aceptar datos. Puede variar dinámicamente.
- Checksum: de la cabecera.
- Urgent pointer: apunta al primer byte de datos urgentes dentro de la parte de datos.
- Options: especifica varias opciones TCP.
- Data: contiene los datos de los protocolos de nivel superior.

2.2.5.2. User Datagram Protocol (UDP)

UDP es un protocolo mucho más simple que TCP, y se usa en situaciones donde los mecanismos de fiabilidad de TCP no son tan necesarios.La cabecera UDP tiene sólo 4 campos:

- source port
- destination port
- length (de la cabecera)
- UDP checksum (opcional)

2.2.6. Protocolos de Aplicación

El stack de protocolos Internet, incluye una serie de protocolos de nivel superior para varias aplicaciones, como gestión de red, correo electrónico, transfrencia de ficheros ...

En la siguiente tabla se dan las aplicaciones más habituales y los protocolos que las implementan.

Application	Protocols
File transfer	FTP
Terminal emulation	Telnet
Electronic mail	SMTP
Network management	SNMP
Distributed file services	NFS, XDR, RPC, X Windows

2.3. Routing

Por el término routing se entiende el traslado de información desde una fuente hasta un destino pasando tal vez por varios puntos intermedios, realizado a nivel 3 del modelo OSI.

Por el contrario el término bridging se refiere a lo mismo, pero realizado a nivel 2.

El routing sirve fundamentalmente para la determinación de rutas óptimas entre dos puntos de una red. Para establecer que ruta es mejor que otra, se utiliza una métrica, que evalúa el coste de utilizar una ruta determinada. Este coste puede ser:

- número de saltos hasta el destino
- retardo de los enlaces
- fiabilidad
- coste económico de los enlaces

El routing se implementa con unos algoritmos, que son de distintos tipos. Estos algoritmos se pueden aplica a distintos tipos de redes (TCP/IP, DECNet, OSI, etc).

Los algoritmos de routing rellenan unas tablas con distinta información, la más habitual son pares (destino, próximo salto).

Cuando una máquina intermedia recibe un paquete, chequea su dirección de destino y la busca en su tabla de rutas, para decidir qué hacer con él.

2.3.1. Tipos de algoritmos de routing

La clasificación de los algoritmos se puede hacer en base a diferentes criterios:

Staticos vs Dinámicos

En realidad los estáticos no son verdaderos algoritmos de routing, puesto que las tablas de rutas son puestas por el administrador de la red, por lo que más bien se podría hablar de mecanismos de routing. Las rutas no cambian a menos que el administrador las cambie. Se utiliza el comando:
route add destination gateway count

Funciona bien en entornos con una topología que no sufre modificaciones frecuentes.

Los dinámicos por el contrario son capaces de configurar las rutas de las máquinas sin tener que introducirlas a mano, y son capces de reaccionar a cambios de topología.

Single-Path vs Multipath

Algunos algoritmos avanzados permiten mantener múltiples caminos hacia una misma ruta, siendo posible multiplexar el tráfico entre ellas.

Planos vs Jerárquicos

En los algoritmos planos todos los routers están al mismo nivel, y cualquiera puede enviar paquetes a otro.

En los jerárquicos, los routers forman una estructura de árbol, de forma que un paquete viaja hacia los routers de mayor rango, y de éstos hacia los de rango menos hasta alcanzar su destino.

Host-Intelligent vs Router-Intelligent

Algunos algoritmos de routing asumen que la fuente del paquete conoce toda la información de rutado para llegar a su destino. En ese caso el router actúa simplemente haciendo forward del paquete hacia el siguiente salto. También se conoce como source routing.

Otros algoritmos actúan como si los hosts no supieran nada sobre rutas. En estos algoritmos los routers determinan el siguiente salto en base a sus propios cálculos.

Intradominio vs Interdominio

Según funcionen sólo dentro de dominios o entre dominios distintos.

Link State vs Distance Vector

Los algoritmos **Distance Vector** funcionan intercambiando información con los routers vecinos a intervalos regulares. Se caracterizan por:

- Se intercambia la tabla de rutas completa.
- Los nodos contienen sólo una información parcial de la red.
- Pasa un cierto tiempo hasta que la red se estabiliza.
- Gasta poca CPU y las tablas ocupan poco.

Los algoritmos **Link State** (también conocidos como shortest path first) envían la información de sus tablas cuando cambia el estado de un enlace. Se caracterizan por:

- Se intercambia información parcial de la tabla de rutas.

- Cada nodo construye un mapa completo de la red.
- Calcula el mejor camino para cada destino.
- Aconsejable en redes estables.
- Son muy estables y de rápida convergencia
- Gastan muchos recursos (CPU y memoria)

2.3.2. Routing Information Protocol (RIP)

Es un protocolo:

- dinámico
- single-path
- plano
- intradominio
- vector-distancia

Es un protocolo desarrollado por Xerox y que fue usado por el conjunto de protocolos Xerox Network Systems (XNS).

La implementación para UNIX y TCP/IP empezó a distribuirse en 1982 con Berkeley Software Distribution (BSD), y se conoce como RIP. Está especificado en la RFC 1058 (1988).

RIP también se ha adoptado por otros productos de diversos fabricantes, como AppleTalk, Novell, 3Com y Banyan.

Formato de la tabla de rutas

Cada entrada en la tabla de rutas de RIP incluye información sobre:

- destino
- siguiente salto hasta ese destino
- métrica asociada
- timers

Destination	Next hop	Distance	Timers	Flags
Network A	Router 1	3	t1, t2, t3	x, y
Network B	Router 2	5	t1, t2, t3	x, y
Network C	Router 1	2	t1, t2, t3	x, y
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

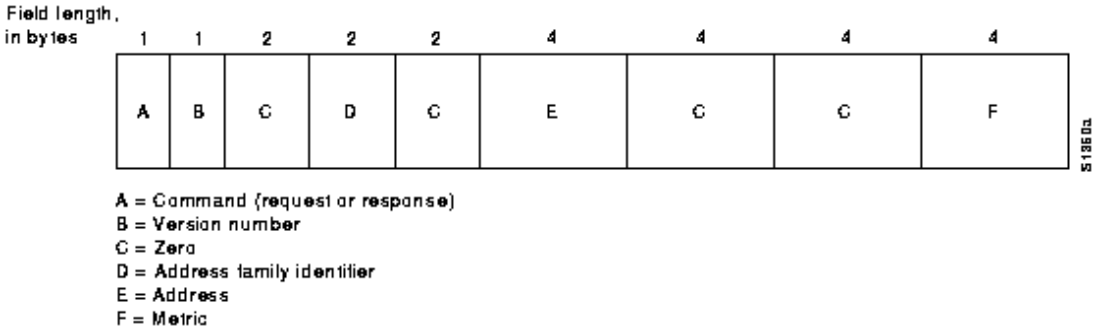
S13G0a

RIP sólo mantiene información sobre la mejor ruta hacia un destino. Cuando hay información sobre una ruta mejor, ésta reemplaza a la antigua.

Cuando ocurre un cambio en la topología, se producen mensajes de actualización de rutas. Por ejemplo, cuando un enlace se cae y es detectado por un router, éste recalcula las rutas y envía mensajes de actualización para propagar el cambio.

Formato de paquete RIP en redes IP

Se implementa sobre el puerto 520 de UDP.



Los campos indican lo siguiente:

- Command: indica si el paquete es una petición o una respuesta. Las respuestas pueden ser solicitadas o espontáneas (se hacen de forma regular)
- Version number de RIP
- Address family identifier (para IP vale 2)
- Address: dirección IP
- Metric: indica cuántos routers hay que atravesar hasta llegar al destino.

En un mismo paquete se pueden meter hasta 25 entradas. Si una tabla necesita más, se envían varios paquetes RIP.

Timers

RIP usa timers para regular su funcionamiento interno.

Generalmente el timer para propagar las rutas se sitúa en 30 s. Cada 30 s cada router da a conocer su tabla completa de rutas.

Otro timer de 90 s se asocia a cada entrada para declararla inválida en caso de no ser actualizada.

Un timer de 270 s sirve para borrar una entrada no actualizada.

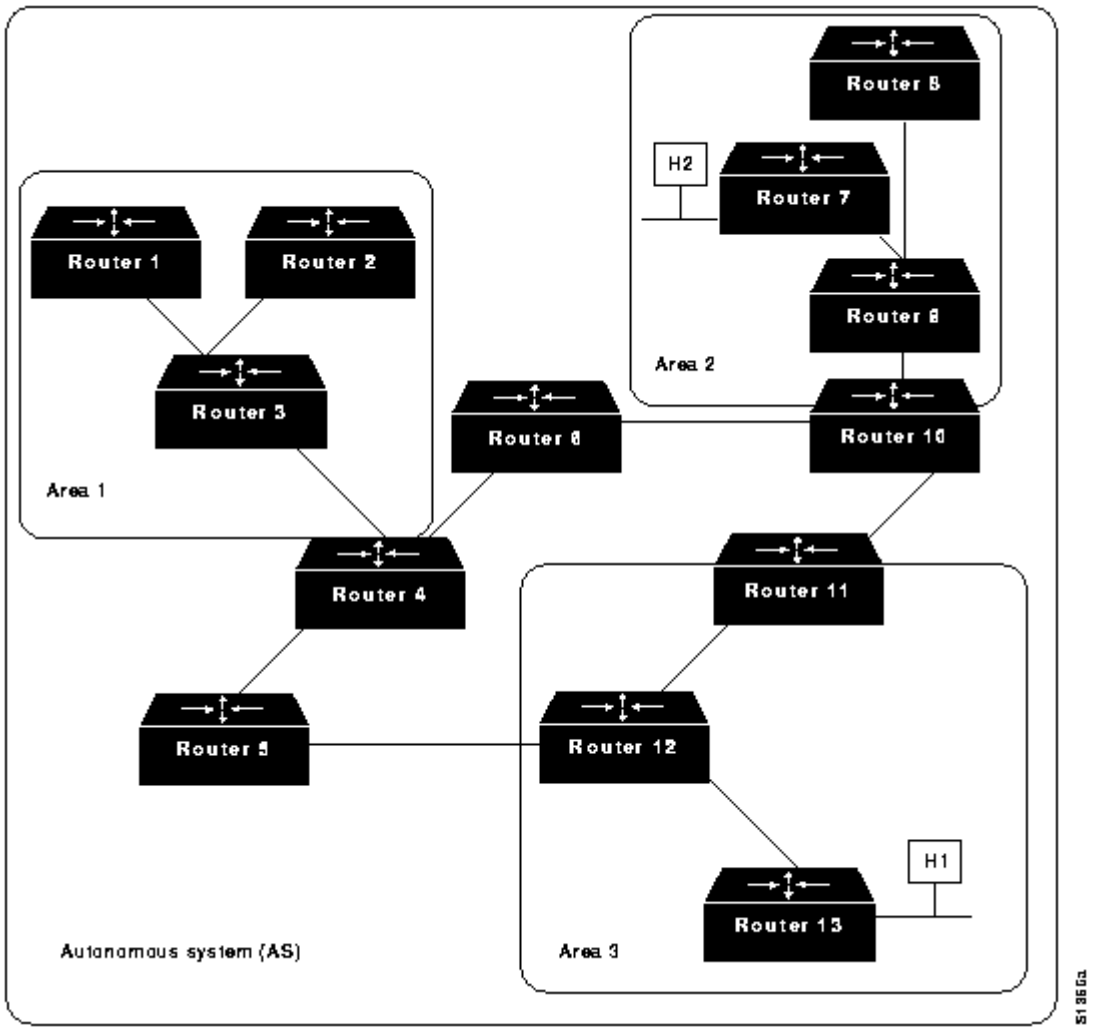
2.3.3. Open Shortest Path First (OSPF)

Es un protocolo:

- dinámico
- single-path
- jerárquico
- intradominio
- link-state

Este protocolo fue desarrollado para la familia de protocolos IP por un grupo de trabajo formado en 1988. La razón de su desarrollo fue que se vio que RIP no era adecuado en grandes redes heterogéneas. Su definición está en el dominio público (RFC 1247).

Usado por routers y algunos UNIX (demonio gated).



2.3.4. Configuración de rutas en UNIX

Los pasos necesarios necesarios para la configuración de rutas son bastante similares en todas las variantes de UNIX. Básicamente primero se configuran los interfaces, después se añaden las rutas estáticas, y opcionalmente se arrancan los demonios para gestión de rutas dinámicas.

En general:

1. ifconfig interface [family] address up options

2. route [-f] op [type] destination gateway hop-count
- op puede ser add o delete
- type puede ser net o host, sino se pone lo averigua por el tipo de dirección de destination

3. routed
-s para modo server
-q para modo host (quiet)
-t imprime los paquetes RIP enviados o recibidos
Implementa el protocolo RIP

4. gated
Implementa varios protocolos: RIP, OSPF, EGP

2.3.5. Configuración en arranque de Linux

Los ficheros de arranque implicados en la configuración de rutas son los siguientes:
/etc/rc.d/rc.inet1
/etc/rc.d/rc.inet2

1. Se configuran los interfaces buscando los ficheros
/etc/hostname
Este fichero contiene la dirección IP o el nombre del interfaz.

2. Se configuran interfaces con las máscaras de red contenidas en /etc/netmasks

3. Con los servicios de red ya arrancados se vuelven a configurar si los valores definidos en NIS son distintos:
ifconfig -au netmask+ broadcast+ ...

4.1. Si existe /etc/defaultrouter se añade una ruta estática con route add y se acaba el proceso

4.2. Si no existe, se comprueba el número y el tipo de los interfaces definidos

4.2.1. Si (hay > 2 interfaces inet) o (hay > 0 interfaces ptp) o (existe /etc/gateways) entonces quiere decir que se trata de un gateway y se arrancan:
in.routed -s
(modo servidor, “canta” las rutas)
in.rdisc -r
(para descubrir rutas)

4.2.2. Si no, no se arranca nada

2.3.6. Mensajes ICMP redirect

Lo habitual es que exista un fichero /etc/defaultrouter, por lo tanto la única ruta que existe en principio es la ruta por defecto. Aunque no estén arrancados los demonios que aprenden rutas, por medio de mensajes ICMP redirect, el kernel también es capaz de modificar su tabla de rutas.

Estos mensajes son enviados por routers que conocen mejores rutas para determinados destinos.

2.3.7. Prácticas

netstat
>netstat -rn

Routing Table:

```
  Destination        Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
127.0.0.1          127.0.0.1          UH      0     0  lo0
193.127.206.10      193.127.206.61      UGH     0    86
150.181.0.0         193.127.206.61      UG      0   545
150.81.0.0          193.127.206.61      UG      0     0
150.121.0.0         193.127.206.61      UG      0    27
150.101.0.0         193.127.206.61      UG      0     0
150.2.0.0          193.127.206.61      UG      0     0
150.64.0.0          193.127.206.61      UG      0     0
150.1.0.0          193.127.206.61      UG      0  1834
150.161.0.0         193.127.206.61      UG      0     0
150.141.0.0         193.127.206.61      UG      0     0
193.127.206.0       193.127.206.1       U       3   254  le0
224.0.0.0          193.127.206.1       U       3     0  le0
default            193.127.206.61      UG      0  2718
```

ifconfig

>ifconfig -a

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 193.127.206.1 netmask ffffffc0 broadcast 193.127.206.63
```

ping

Comprueba conectividad a nivel IP:

>ping www.midominio.es

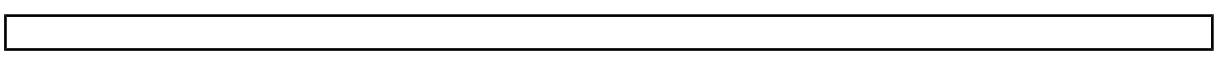
traceroute

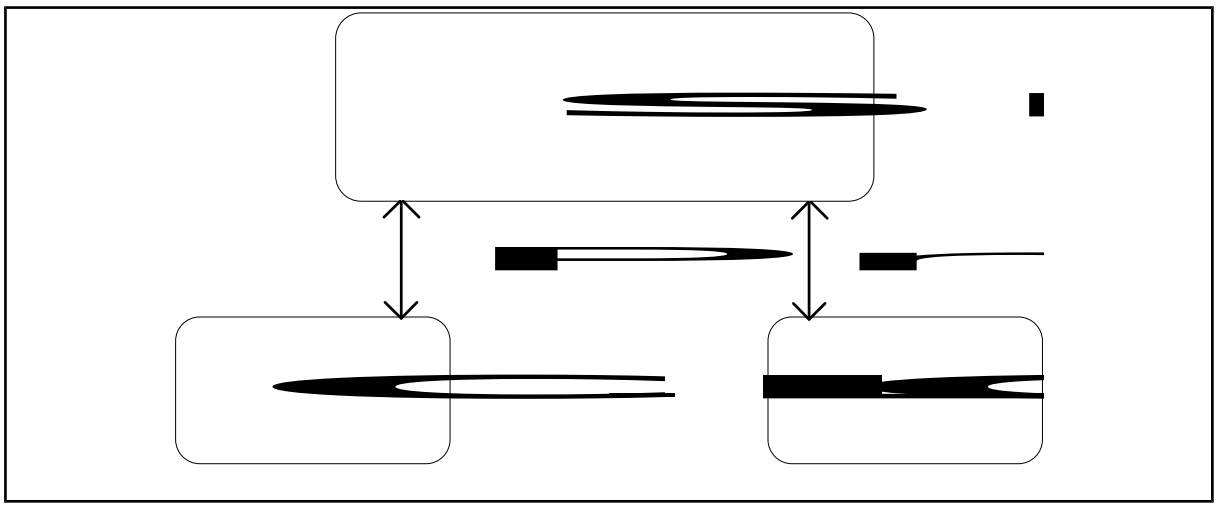
>traceroute rianxo.midominio.es

```
traceroute to rianxo.midominio.es (193.144.127.33), 30 hops max, 40 byte packets
 1  midominio (193.127.206.61)  -3.10412e+231 ms *  -3.10412e+231 ms
 2  193.127.206.62 (193.127.206.62)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
 3  ibernet (193.127.211.1)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
 4  madrid.eunet.es (193.127.1.1)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
 5  Amsterdam12.NL.EU.net (134.222.12.1)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
 6  Amsterdam6.NL.EU.net (134.222.85.6)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
 7  Amsterdam1.NL.EU.net (134.222.228.22)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
 8  Amsterdam1.dante.net (193.148.15.35)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
 9  NL-s1.dante.bt.net (194.72.26.5)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
10  NL-f0-0.eurocore.bt.net (194.72.24.1)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
11  ES-s0.dante.bt.net (194.72.24.173)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
12  EB-Madrid1.rediris.es (194.72.26.34)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
13  midominio-router.rediris.es (130.206.211.34)  -3.10412e+231 ms  -3.10412e+231 ms *
14  rianxo.midominio.es (193.144.127.33)  -3.10412e+231 ms  -3.10412e+231 ms  -3.10412e+231 ms
```

3. Servicios TCP/IP de Internet

La mayoría de los servicios de Internet están estructurados de acuerdo al modelo cliente-servidor





El **Cliente** es el responsable de interactuar con el usuario, aceptando por ejemplo entrada desde teclado, y mostrando el resultado de salida.

El **Servidor** es responsable de realizar las tareas solicitadas por el cliente: por ejemplo acceso a datos, realizar cálculos, etc.

El cliente y el servidor pueden ejecutarse en el mismo ordenador, pero más típicamente lo harán en dos ordenadores distintos, conectados por medio de una red de datos.

Para la red de datos Internet (con ello nos referimos al conjunto de protocolos TCP/IP) están definidos una serie de servicios estructurados de acuerdo al modelo cliente-servidor:

- NIS
- DNS
- Telnet
- FTP
- WWW
- Correo Electrónico

Cada uno de estos servicios utiliza en la parte servidor unos números de puertos definidos. Estos puertos pueden ser de TCP o de UDP:

- sendmail(25 de TCP)
- telnet(21 de TCP)
- http(80 de TCP)

3.1. Servicios Internet en sistemas UNIX

Los servidores están implementados con procesos que corren en modo background, llamados daemons (demonios). Estos procesos demonio “escuchan” en determinados puertos esperando aceptar peticiones de información por parte de los clientes. Existen dos tipos de demonios: unos son lanzados por un “superdemonio”, llamado *inetd* que escucha en diversos puertos, y cuando recibe una petición lanza el

demonio específico para ese puerto. Y otros son lanzados en el arranque de la máquina, y atienden peticiones sólo en su puerto característico. A estos demonios se les llama *standalone*.

3.1.1. inetd

El superdemonio inetd se configura en el fichero /etc/inetd.conf:

```
#
# Configuration file for inetd(1M).  See inetd.conf(4).
#
# To re-configure the running inetd process, edit this file, then
# send the inetd process a SIGHUP.
#
# Syntax for socket-based Internet services:
#  <service_name> <socket_type> <proto> <flags> <user> <server_pathname> <args>
#
# Syntax for TLI-based Internet services:
#
#  <service_name> tli <proto> <flags> <user> <server_pathname> <args>
#
# Ftp and telnet are standard Internet services.
#
ftp       stream tcp      nowait  root    /usr/sbin/ftpd                  ftpd
telnet    stream tcp      nowait  root    /usr/sbin/telnetd               telnetd
talk   dgram  udp         wait    root    /usr/sbin/talkd                 talkd
```

Cada línea tiene los siguientes campos: **nombre del servicio**, que en realidad este nombre se corresponde con un número de puerto que aparece en el fichero /etc/services. **Protocolo** de transporte usado, puede ser stream tcp o dgram udp. **Demonio** que se lanza, y **parámetros** del demonio.

Para habilitar, deshabilitar o cambiar un servicio determinado se debe comentar la línea correspondiente (con #) y lanzar una señal SIGHUP al demonio *inetd*.

3.1.2. xinetd

Es un “superdemonio” mejorado que permite realizar control de accesos a los servicios. Con el *inetd* tradicional el modo de funcionamiento de los servicios es todo o nada, no se puede controlar el acceso por determinados patrones.

xinetd puede ser configurado para:

- aceptar conexiones que vengan desde una determinada dirección IP
- aceptar conexiones sólo de determinados usuarios
- rechazar conexiones fuera de un determinado horario
- realizar un log de servicios cuando las conexiones son aceptadas o rechazadas

xinetd se puede obtener en la URL: **ftp://ftp.telebase.com/pub/security**

3.1.3. Standalone servers

Cada vez que un cliente solicita una petición a un servidor, el inetd tiene que lanzar el proceso para atenderlo. Esto es lento para servicios que se solicitan con frecuencia, por ejemplo para el WWW, en el cual cada objeto de una página debe ser transferido por un servidor exclusivo. Para

este tipo de servicios se emplean procesos dedicados sólo ese servicio. Ejemplos de servicios que funcionan como standalone son:

- DNS (named)
- WWW (httpd)
- Correo Electrónico (sendmail)

4. Servicio de Nombres DNS

El Domain Name Service (DNS) es un protocolo de nivel de Aplicación, que forma parte del conjunto de protocolos TCP/IP. El DNS puede verse como una base de datos distribuida y jerárquica, usada por las aplicaciones TCP/IP y que sirve fundamentalmente para dos propósitos:

- Traducir nombres simbólicos a direcciones IP
- Proveer información de rutas para el servicio de correo electrónico.

La implementación más usada de DNS se conoce como BIND (Berkeley INternet Domain Service), y se estructura según la arquitectura cliente-servidor. La versión actual (4.9.3) está mantenida por una organización sin ánimo de lucro llamada Internet Software Consortium (http://www.isc.org/isc).

En la parte del **cliente**, una aplicación que tenga que resolver un determinado nombre (por ejemplo la aplicación FTP) se enlaza con una librería específica que es la que traduce desde nombres hacia direcciones IP. Esta librería se conoce con el nombre de *resolver*.

En la parte del **servidor**, existen procesos que se encargan de resolver las direcciones para un determinado dominio. Desde el punto de vista de la autoridad que un servidor tiene sobre un dominio, existen varios tipos de servidores:

- Master. Son autorizados (authoritative) para el dominio que representan, es decir, pueden decir sin dudas si un determinadado nombre o dirección existen.
- Primario. Para un dominio sólo hay un master primario, que lee los datos referidos al DNS de su disco
- Secundario. Los servidores secundarios no contienen una base de datos propia sobre el dominio, sino que la copian del servidor primario del mismo (en el arranque) y la copia se actualiza cada cierto tiempo. Su función es diversificar las peticiones, relajar la carga sobre el servidor primario y servir de respaldo por si el Primario falla.
- Caching-Only. No son autoridad. Aunque todos los servidores hacen cache de las últimas consultas, los de éste tipo no tienen ninguna información inicial por sí mismos.

Para un dominio dado existirán al menos dos servidores autorizados de DNS, primario y secundario, para el caso de que falle el primario.

Desde el punto de vista de cómo resuelven las consultas que se le hacen, existen dos tipos de servidores:

- Recursivos: Al recibir una pregunta el servidor se encarga de devolver la respuesta definitiva. Para ello es normal que tenga que preguntar a otros servidores
- No recursivos: Al recibir una pregunta el servidor lee sus bases de datos. Si encuentra el dato pedido, lo devuelve. Puede ocurrir que sin conocer el dato, sepa que ordenador lo contiene. En particular puede conocer al servidor de nombres del dominio en que se encuentra la máquina. En ese caso el servidor no recursivo envía al interrogador la dirección del servidor de nombres del dominio. El ordenador interrogador preguntará a este nuevo servidor de nombres.

Como se ha dicho, el DNS está organizado de forma jerárquica. Cada dominio está gestionado por un servidor, que puede delegar la administración de sus subdominios. El dominio **es.** está gestionado por RedIRIS, en caso de que queramos administrar nuestro dominio (por ejemplo midominio.es.), primero deberemos comunicarlo a la autoridad (RedIRIS), y después instalar correctamente los servidores primario y secundario para este dominio. Este proceso sigue para los dominios de nivel inferior, es decir, el servidor de midominio.es. puede delegar la gestión de los dominios que parten de él a otros servidores de DNS.

4.1. Clientes DNS

Un ordenador cliente es aquel que pide a otro la traducción de un nombre a dirección IP o viceversa. La mayor parte de los ordenadores de una red actúan como clientes de uno o más servidores. Cuando un programa necesita una conversión invoca a una librería específica. Estas funciones envían un mensaje con la pregunta al servidor configurado. El servidor resuelve la pregunta y envía la respuesta en un mensaje al cliente.

En el caso de *midominio.es* cada uno de sus subdominios tiene un servidor de DNS al que preguntan los ordenadores de la red, y que a su vez dan los nombres de estos ordenadores a los que les pregunten. Así el servidor de DNS de *ha.midominio.es* es *ha.ha.midominio.es.* El firewall *entrada* está configurado como servidor de nombres del dominio *midominio.es*. Al recibir peticiones externas las trasladará al servidor debido.

4.1.1. PC

La configuración del DNS en los PCs depende del Sistema Operativo instalado. En general basta con indicar que existe un servidor y su dirección IP. En el caso de Windows 95 basta con acudir a las fichas de configuración del TCP/IP. En otros sistemas Windows depende del stack TCP/IP instalado. No suele presentar mayores dificultades.

4.1.2. UNIX

maq1 y el propio *entrada* actúan como clientes de *entrada*. Para actuar como clientes las estaciones UNIX deben configurar los siguientes ficheros:

/etc/hosts

Contiene pares nombre-dirección IP conocidos por el ordenador sin tener que recurrir al DNS.

```
#
# /etc/hosts de entrada.midominio.es.
#
193.144.104.9    entrada-interna
172.20.1.1       entrada-internet

#
# /etc/hosts de maq1.midominio.es
#
127.0.0.1                localhost
193.1.1.1                entrada
193.1.1.2                maq1
```

/etc/resolv.conf

Contiene las indicaciones sobre cómo resolver los nombres con *dns*. El campo *domain* contiene el nombre completo del dominio de la estación. Este dominio se emplea para completar los nombres de los ordenadores. El campo *nameserver* indica la dirección IP del servidor de nombres a utilizar. En este caso se ve que es la propia estación. Una estación Unix puede tener configurados en resolv.conf hasta tres *nameserver* distintos con los que va probando en caso de no recibir la respuesta de los anteriores.

```
#
# /etc/resolv.conf de entrada.midominio.es.
#
domain          midominio.es
nameserver      127.0.0.1
nameserver      193.1.1.1

#
# /etc/resolv.conf de maq1.midominio.es.
#
domain          midominio.es
nameserver      entrada
```

4.2. Servidores DNS

Los servidores DNS son ordenadores que guardan y sirven la información con los pares nombre-dirección en las redes TCP/IP. Para el caso de las máquinas UNIX los servidores DNS se implementan con un proceso (named) que tiene asociado un fichero de configuración (/etc/named.boot). Opcionalmente, y según el tipo de servidor existirán ficheros de base de datos sobre el dominio gestionado por el servidor (para el caso de que sea primario)

En este ejemplo se ha configurado al firewall *entrada* como servidor de nombres del dominio *midominio.es*. Conoce en particular las direcciones de los servidores de los distintos subdominios.

named

Este es el demonio servidor de nombres. Se lanza al arrancar la estación y permanece encendido y escuchando las peticiones que se le hagan. Su "pid" (Process Identifier) se escribe en el fichero */etc/named.pid*. Admite estas señales:

> kill -HUP `cat /etc/named.pid`
Fuerza la actualización de la base de datos.

> kill -INT `cat /etc/named.pid`
Vuelca los datos de la memoria caché almacenados en la memoria en un fichero: /var/tmp/named_dump.db

> kill -USR1 `cat /etc/named.pid`
Aumenta el nivel de depuración en uno. El resultado de esta depuración se escribe en el fichero /var/tmp/named.run

> kill -USR2 `cat /etc/named.pid`
Anula la depuración

/etc/named.boot

Este fichero es el de configuración de *named*. Indica para qué dominios y redes se pueden devolver respuestas y el tipo de servidor de que se trata (primario, secundario o sólo cache).

Servidor sólo cache:

```
; BIND boot file for cache-only server
directory /etc
cache    .                              named.root
primary  0.0.127.IN-ADDR.ARPA           named.local
```

Servidor secundario:

```
; BIND boot file for secondary server
directory /etc
cache    .                              named.root
secondary       midominio.es                    193.1.1.1       named.host
secondary       1.1.193.IN-ADDR.ARPA    193.1.1.1       named.rev
primary         0.0.127.IN-ADDR.ARPA    named.local
```

Servidor primario:

```
; BIND boot file for primary server (entrada)
directory /etc
cache    .                              named.root
primary  midominio.es                           named.host
primary  1.1.193.IN-ADDR.ARPA           named.rev
primary  0.0.127.IN-ADDR.ARPA           named.local
```

directory indica el directorio en el que se encuentran por defecto los ficheros que luego se señalan.

cache quiere decir que el contenido del fichero que sigue debe ser guardado en la memoria del sistema al arrancar, y afecta al dominio del segundo campo. En este caso el fichero db.cache contiene las direcciones de los servidores de nombres de el dominio raíz (.). Al arrancar, *named* no conoce más direcciones que las de su propia red. Para ir formando la caché se le dan estas direcciones, de manera que pueda ir conociendo a otros servidores con las distintas preguntas.

primary indica que el servidor es primario para el dominio indicado a continuación, y que los datos sobre este dominio están en el fichero que aparece en el tercer campo.

secondary indica que el servidor es secundario para el dominio indicado, y que los datos los debe leer de alguno de los servidores que aparecen a continuación (generalmente el primario, aunque pueden especificarse otros secundarios para el caso de que no esté disponible). Por último se especifica un fichero donde guarda los datos para posteriores arranques (se chequea con algún servidor para saber si sigue siendo válido)

La línea

```
primary         127.144.193.IN-ADDR.ARPA     named.rev
```

sirve para resolver las peticiones inversas: dada una dirección IP, averiguar el nombre de la máquina. Para ello se concibe todo el conjunto de las redes IP como un subdominio del dominio *arpa*, en concreto el subdominio *in-addr.arpa*. Este subdominio se divide a su vez en otros subdominios, uno por cada red IP existente. La línea se entiende así: este ordenador es servidor de nombres de la red IP 193.144.127. La información correspondiente se encuentra en un fichero (para el servidor primario) o en otro servidor.

La línea

```
primary         0.0.127.IN-ADDR.ARPA         named.local
```

indica que este ordenador es el servidor de nombres para la red 127.0.0.0 (o sea, él mismo).

Los ficheros de base de datos para servidores primarios de DNS

La base de datos DNS se compone de los ficheros referenciados en named.boot. Estos ficheros tienen un formato fijo.

```
#
# /etc/named.host
#
; midominio.es, mapeado directo
@       IN      SOA     entrada.midominio.es.  root.entrada.midominio.es.     (
                        1996042801      ;Serial
                        21600           ;Refresh
                        1200            ;Retry
                        3600000         ;Expire
                        432000)                 ;Minimum

        IN      NS              entrada.midominio.es.
        IN      MX      10              maq1.midominio.es.
localhostIN     A               127.0.0.1
entrada         IN      A               193.1.1.1
maq1    IN      A               193.1.1.2
www     IN      CNAME           entrada.midominio.es
dns     IN      CNAME           entrada.midominio.es
mail    IN      CNAME           maq1.midominio.es
mailhost        IN      CNAME           maq1.midominio.es

pre     IN      NS              pre.pre.midominio.es.
...
pre     IN      MX      20      pre.pre.midominio.es.
...
pre.pre IN      A               193.1.1.7
```

Los campos de una línea de la base de datos tienen la siguiente estructura:

```
[nombre] [ttl] [clase] tipo dato
```

nombre es el de la máquina o dominio al que se refiere esta línea. Si está vacío, se supone que es el mismo *nombre* que la línea anterior. Si el nombre es @, es el mismo que el del dominio.

ttl es el "time to live" o tiempo de vida de este dato en la caché del de otro servidor. Generalmente no se indica y se toma del SOA correspondiente (ver más abajo)

clase se refiere al tipo de la red, que es casi siempre IN (Internet). Por defecto toma este valor.

tipo indica el tipo de dato que se da sobre el *nombre*

Por último se encuentra el o los *datos*, cuyo formato depende del *tipo*.

Registro SOA: El registro SOA (Start Of Authority) guarda información general sobre los registros que le siguen. Esta información es la siguiente:

entrada.midominio.es
Es el nombre del ordenador servidor primario para el dominio indicado en *nombre*.

root.entrada.midominio.es
Dirección de correo del administrador de este dominio. En la forma tradicional es root@entrada.midominio.es. En este registro se cambia la "@" por "."

A continuación aparecen una serie de números entre paréntesis. Estos indican los siguientes datos:

1996042801 ;*Serial*
Es el dato más importante. Número de serie de la configuración. Los servidores secundarios sólo cargarán una base de datos si su número de serie es superior al de la que ya tienen cargada. Como se puede ver, hemos asociado el número a la fecha de última edición (27-2-96, versión 01). Siempre que se haga una actualización se debe cambiar este número.

21600 ;*Refresh*
Tiempo en segundos tras el que los servidores secundarios deben volver a consultar al primario.

1200 ;*Retry*
Si un servidor secundario no logra cargar la base del primario, tiempo entre reintentos, en segundos.

3600000 *;Expire*
Tiempo tras el que expiran los datos de la base, si no se logran recargar.

432000 ;*Minimum*
Este campo se toma como Time To Live si no se especifica otra cosa en las entradas. Tiempo durante el cual los registros de la base de datos son válidos y pueden estar almacenados en una caché.

Registro NS : El *dato* siguiente es el nombre de un servidor de nombres con autoridad sobre la zona (authoritative). Este servidor puede ser primario o secundario.

Registro CNAME: "Canonical Name". El *dato* es el nombre canónico del *nombre* de la línea. Esto permite dar varios nombres a una misma máquina, que viene dada por un solo nombre canónico.

Registro A: El *dato* es la dirección IP que corresponde al *nombre*.

Registro PTR: Registro inverso. *nombre* es una dirección IP y *dato* es el nombre de la máquina.

Registro MX: Mail Exchanger. Indica el nombre del ordenador al que se envía el correo dirigido a *nombre*. El *dato* tiene dos partes:

nombre IN MX Prioridad Nombre_máquina

Prioridad es un número entero
Nombre_máquina el nombre de un ordenador que recoge el correo destinado a *nombre*.

Pueden existir varios registros MX para un mismo *nombre*. El correo se intenta entregar en primer lugar al destino con un menor número en *Prioridad*. A continuación se intenta en el siguiente en prioridad, y así consecutivamente.

Registro HINFO: Contiene información sobre el hardware y el sistema operativo de la máquina:

nombre IN HINFO Hardware OS

Registro WKS: Well Known Services soportados por un protocolo (TCP o UDP) en una determinada dirección

nombre IN WKS protocol list of services

4.3. Práctica: Comprobación del DNS con nslookup

5. Servicio de Transferencia de Ficheros FTP

Existen dos métodos de acceso a un servidor FTP: anónimo y de usuario. Un FTP de usuario es para aquellos usuarios que tienen una cuenta en el host servidor. Cuando se accede, el usuario proporciona su nombre y su contraseña. De esa forma, el servidor da libre acceso a los ficheros de la misma forma que si el usuario hubiera hecho un *login*, aunque sólo puede ejecutar determinados comandos (entre ellos listar directorios y cambiar de directorio). Sin embargo este método de acceso no se recomienda, porque las contraseñas se envían al servidor en formato plano, y además los usuarios con contraseñas fáciles de adivinar podrían crear problemas de seguridad. Por ello, el método recomendado es el acceso anónimo al servidor FTP. De esa forma pueden acceder al servidor tanto los usuarios que tienen cuenta como los que no la tienen, restringiendo el acceso a las zonas del disco que decida el administrador del servicio.

Durante la instalación, se crea un usuario llamado ftp sin password de acceso y sin login shell para permitir el FTP anónimo a la máquina. Al hacer *FTP anónimo* (con nombre de usuario **ftp** o **anonymous**) el demonio del FTP hace una llamada a chroot() con el directorio que se haya especificado como home para el usuario ftp. En ~ftp/pub/ se instalarán los ficheros que se quieran hacer públicos.

La forma de autentificar a los usuarios (no anónimos) se basa en estas reglas:

1. El usuario debe estar en /etc/passwd (para el caso del acceso anónimo, el usuario es ftp, y la clave no necesaria).

2. Si el usuario está en el fichero /etc/ftpusers, se niega el acceso.

3. Si el shell especificado en /etc/passwd para el usuario no está listado en /etc/shells, se niega el acceso.

Respecto al área de ficheros que se hace pública, la precaución que hay que tener es la de no crear enlaces desde este directorio hacia otras partes del disco, porque de esa forma abriríamos una vía hacia información que no queremos hacer pública. Un método para evitar este problema consiste en crear un sistema de ficheros específico para situar la información pública; de ese modo no se pueden crear enlaces físicos entre este sistema de ficheros y el resto (ni intencionadamente ni por descuido).

5.1. Configuración de FTP en UNIX

La configuración del servicio de FTP requiere una serie de pasos que se detallan a continuación:

1. Instalar el módulo ftp.
2. Fijar una entrada en el fichero de configuración del superdemonio de internet inetd. En /etc/inetd.conf:

ftp stream tcp nowait root /usr/sbin/ftpd ftpd

3. Configurar el área de ficheros para acceso público.

A continuación se dan como referencia los directorios y ficheros necesarios para el funcionamiento del demonio de FTP.

~ftp

Directorio de login del usuario ftp. Es propiedad de root y nadie tiene permiso de escritura. Además no debe estar en un sistema de ficheros montado con la opción nosuid.

~ftp/bin

Este directorio contiene los ejecutables que necesita el demonio de FTP, en concreto ls (éste tiene modo 111). Es propiedad de root y nadie tiene permiso de escritura. Es un enlace simbólico a ~ftp/usr/bin.

~ftp/etc

Este directorio contiene un fichero passwd, que lógicamente es distinto del de sistema. Este fichero es necesario para que el demonio determine los permisos de lectura de los ficheros, y los ejecutables del directorio bin obtenga el usuario y grupo (fichero group) propietarios. Los ficheros passwd y group sólo contienen una información mínima, y son usados para mapear identificadores de grupo y usuario a nombres de grupo y usuario (usado por ls). Es propiedad de root y nadie tiene permiso de escritura.

~ftp/pub

Este directorio contiene los ficheros que se quieren hacer públicos (se utiliza este nombre por convenio). Es propiedad de ftp y tiene modo 555 ó 777 (este último sólo si se quiere permitir la escritura).

5.2. WU FTP para UNIX

Conocido como WU Archive FTP, tiene una serie de características que lo hacen más interesante que el FTP estándar. Se puede encontrar en la URL

ftp://wuarchive.wustl.edu

La distribución está en fuente, y hay que compilarla para cada sistema UNIX concreto. En la distribución se pueden encontrar instrucciones para cada sistema operativo en particular. Entre las características mejoradas de este FTP están:

- Se pueden establecer características de acceso basadas en la identidad del usuario que accede, o en la dirección desde dónde se accede. Se crean clases de usuarios, y se puede restringir el número de usuarios de una clase que acceden a un archivo determinado.
- Basándose en esas clases, se puede controlar el acceso a determinadas funciones, por ejemplo quién puede borrar ficheros.
- El servidor puede realizar funciones de compress, uncompress o tar sobre ficheros que está transmitiendo.
- Se pueden registrar cargas, descargas, o en general cualquier comando enviado por los clientes FTP.
- Se permite hacer un shutdown del servidor de forma ordenada, adviertiendo a los clientes que estén accediendo en ese momento

Los ficheros de configuración son los siguientes:

5.2.1. ftpaccess

Define las clases de usuarios basados en la cuenta y en el host desde el que acceden los usuarios. Según la clase a la que pertenezca el usuario, el servidor se comportará de forma distinta. Para ftpd hay tres tipos de usuarios:

- real: como un servidor ftp normal, no se hace chroot()
- anonymous: los que acceden como ftp o anonymous, sí se hace chroot()
- guest: tienen cuentas protegidas con passwords, con acceso restringido, sí se hace chroot()

Para definir una clase se usa la siguiente sintaxis: **class classname type [,type...] address [address...]**
Donde *type* es uno o más de los tipos vistos anteriormente, y *address* es una dirección IP o una basada en dominios, con * permitidos. Por ejemplo: **class local real *.midominio.es**

A los usuarios que no estén en ninguna clase se les *deniega el acceso*. Además también se puede denegar el acceso de forma explícita a determinados usuarios, con un mensaje asociado: **deny *.midominio.es /etc/msgs/msg.dead**

Para restringir el acceso se utiliza el comando limit: **limit class number times mesagge**
Con este comando se puede limitar el número de usuarios de una clase determinada que pueden estar accediendo simultáneamente al servidor, e indicar los días a los que se aplican. Por ejemplo para limitar a 200 en sábados y domingos, y a 100 el resto de la semana:
limit anonymous_class 200 SaSu

limit anonymous_class 100 Any /etc/ftpmsgs/msg.toomany

Además del número de accesos simultáneos se pueden restringir las operaciones que se pueden realizar, como por ejemplo las operaciones de compresión o los comandos que el usuario puede ejecutar.

5.2.2. ftpconversions

Este fichero configura el comportamiento del servidor. En él se le indica las sintaxis para las compresiones y descompresiones basadas en los sufijos de los ficheros.

El servidor de ftp de WU, entiende formatos de ficheros por su sufijo. Así, dado el fichero **prueba.Z**, el cliente puede solitarlo haciendo **get prueba**, de forma que el servidor primero descomprimirá el fichero y luego lo enviará. De igual forma, se puede solicitar que un fichero que no está comprimido se comprima previamente y después se nos envíe.

5.2.3. ftphosts

Es para casos especiales en los que interesa deshabilitar de forma directa el acceso desde una determinada **máquina**, (de forma análoga a lo que permite el fichero ftpusers para usuarios), con entradas del tipo:
deny anonymous *.midominio.es 190.2.3.*

5.3. Configuración de FTP en Windows NT

El Internet Information Server de Microsoft, además de actuar como servidor de WWW, puede actuar también como servidor de ftp. Permite el acceso al servicio a todos los usuarios que tengan cuenta en el servidor NT. También permite el acceso de modo anónimo.

La configuración de este servidor se hace a través del Internet Service Manager, que presenta al administrador un conjunto de cuadros de diálogo con diferentes opciones. Los cuadros de diálogo permiten configurar:

- El directorio origen de los usuarios.
- El tipo de listado presentado a los usuarios (algunos navegadores necesitan que el listado sea en formato UNIX).

Los permisos de lectura y escritura de los directorios.

Se pueden crear ficheros que describan los contenidos de los directorios. Este fichero se presenta automáticamente a los navegadores.

Se pueden crear directorios especiales para cada usuario, que pueden actuar como directorio origen (en lugar de su cuenta).

6.- Servicio WEB

6.1. Hypertext Transfer Protocol (HTTP)

Los clientes y servidores Web se comunican usando el protocolo HTTP. Cuando un cliente hace una petición a un servidor, aquél abre una

conexión, el documento es transferido usando HTTP, y la conexión se cierra.

El modelo WWW (World-Wide Web) hace uso de la naturaleza distribuida de los hipertextos y, mediante un esquema cliente-servidor, proporciona un servicio de transferencia de hipertextos basado en el Protocolo HTTP.

HTTP (Hipertext Transfer Protocol), es un protocolo de nivel de aplicación ligero, que proporciona la agilidad y velocidad necesarias para sistemas de informacion Hipermedia distribuidos.

HTTP es un protocolo orientado a objetos, de caracter genérico, que puede ser usado para muchas tareas, tales como, servidores de nombres y sistemas de manejo de objetos distribuidos, a través de la utilización de las extensiones de sus comandos.

Una caracteristica del Protocolo HTTP es el *typing* y la negociación de la representacion de los datos, permitiendo que los documentos sean presentados independientemente de los datos que sean transferidos.

6.1.1. Versiones de HTTP

HTTP / 0.9

Es el protocolo original HTTP. No permitía la negociación de tipos de datos. Se describía en unas pocas páginas. Está obsoleto.

HTTP / 1.0

HTTP/1.0 soporta la negociación de tipos de datos entre clientes y servidores, añadiendo información MIME al protocolo.

La única documentacion para las primeras versiones del protocolo HTTP / 1.0 consistía en una discusion redactada en forma HTML , escrita por Tim Berners-Lee y luego actualizado por Ari Luotonen. Esta documentacion está disponible solo por razones historicas , cuando esta documentacion se hizo demasiado extensa fue reemplazada por el Internet Drafts, y no refleja practica actual entre aplicaciones de WWW .

El útimo draft es HTTP / 1.0 Internet draft 04.

HTML usa el tipo MIME “text” y el subtipo “html”:

text/html

Se definen además otros muchos tipos soportados por clientes y servidores, como por ejemplo para imágenes:

image/gif

Cada vez que un cliente solicita una página, éste pasa una lista de los tipos MIME que soporta. Con esta información el servidor envía sólo aquella información que el cliente es capaz de representar. El servidor envía primero el tipo MIME del fichero que va a mandar, después una línea en blanco, y por último el fichero de datos.

HTTP-NG

Una propuesta para la Proxima Generacion de HTTP, propuesta por Simon Spero. Es un protocolo binario con muchos nuevos aspectos para accesos más rapidos que usan TCP.

6.1.2. Comandos HTTP

El protocolo tiene 4 fases:

1. Establecimiento de conexión
2. Petición (cliente --> servidor)
3. Respuesta (servidor --> cliente)
4. Final de conexión

Formato de Petición

```
Request          =     SimpleRequest | FullRequest
SimpleRequest    =     GET <uri> CrLf
FullRequest      =     Method URI ProtocolVersion CrLf
                              [*<HTRQ Header>]
                              [<CrLf> <data>]
<Method>         =     <InitialAlpha>
ProtocolVersion  =     HTTP/1.0
uri              =     <as defined in URL spec>
<HTRQ Header>    =     <Fieldname> : <Value> <CrLf>
<data>           =      MIME-conforming-message
```

Formato de Respuesta

```
<status line>   ::=    <http version>  <status code>  <reason line> <CrLf>
<http version>  ::=    3*<digit>
<status code>   ::=    3*<digit>
<digit>         ::=    0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<reason line> ::=   * <printable>
```

6.2. LENGUAJE HTML 2.0

El lenguaje HTML nace en 1991 inventado por Tim-Berners-Lee del CERN. Es un sistema de hipertexto concebido como medio de transmisión de información entre físicos del CERN. En 1993 Dan Connelly escribe el primer draft del lenguaje: HTML 1.0. Y en 1994 se publica el draft de HTML 2.0, cuyas diferencias más importantes son:

- adición de los formularios que pueden ser rellenados y enviados al servidor, incluyendo elementos como menús, campos de texto, botones, etc.
- creación de mapas sensibles “pinchables” por el usuario

En la actualidad ya existe la versión HTML 3.0. Aunque es soportada por pocos navegadores.

Paralelamente a la versión de HTML 2.0 se ha definió el Common Gateway Interface (CGI), que define la interfaz entre los programas ejecutables y el WWW.

La descripción que sigue es la de la versión HTML 2.0

6.2.1. Fundamentos del lenguaje HTML 2.0

Se recoge en este apartado una breve descripción del lenguaje HTML. Para obtener una documentación completa consulta el documento del curso de "Creación de Páginas WEB".

Un “programa” escrito en HTML está compuesto por texto ASCII y comandos de este lenguaje. Estos comandos pueden estar en mayúsculas o minúsculas indistintamente. Los espacios, líneas en blanco, tabuladores, etc. no son tenidos en cuenta para formatear documentos, en vez de eso se utilizan comandos especiales.

Los comandos se insertan en el texto como etiquetas que tienen la siguiente estructura:
<etiqueta>texto...</etiqueta>

Algunas comandos tienen parámetros con valor:
<etiqueta parametro=valor>texto...</etiqueta>

6.2.2. Estructura de un documento HTML 2.0

Básicamente un documento escrito en HTML tiene dos partes: una cabecera y un cuerpo, que van delimitados por las etiquetas **HEAD** y **BODY**.

```
<HTML>
<HEAD>
...
</HEAD>
<BODY>
...
</BODY>
</HTML>
```

Cabecera

En la cabecera se incluyen campos que contienen metainformación, es decir, que no afectan a la estructura del documento, y que en algunos casos son ignoradas. Algunos de estos campos son los siguientes.

- Título que aparece fuera del documento, por ejemplo Netscape lo muestra en la parte superior de la ventana:
<title>Titulo del documento</title>

- Path del documento actual, que se usa si las referencias contenidas dentro de él son relativas:
<base href=“URL”>

Cuerpo

Es aquí donde se pone el texto a mostrar, y donde se pueden incluir todas las etiquetas (salvo las definidas para la cabecera). Las etiquetas de formato general son las siguientes:

Comandos de Cabecera

Existen hasta 6 niveles.
<h1>...</h1>
...
<h6>...</h6>

Comandos de Formato

<p>	nuevo párrafo
** **	nueva línea
<hr>	regla horizontal
<pre>...</pre>	texto preformateado
<listing>...</listing>	listado

Comandos de Estilos Lógicos

...	énfasis
...	refuerzo
<code>...</code>	código fuente

Comandos de Estilos Físicos

...	negrita
...	itálica
<u>...</u>	subrayado
<tt>...</tt>	máquina de escribir

Imágenes

Para incluir una imagen en un documento se utiliza la etiqueta *img* (que no tiene terminador):
<img src="URL" [alt=] [align=] [ismap=]>
donde ...

- **URL** determina la imagen a mostrar
- **alt** muestra un texto alternativo para el caso de que el browswer no pueda mostrar imágenes
- **align** indica la forma de alineamiento, puede ser t*op, middle, bottom, left, right*.

Enlaces

Para entender el concepto de enlaces en HTML, hay que definir previamente lo que es una URL (Uniform Resource Location). Este concepto puede verse como una extensión al sistema de ficheros local, que incluye a todos los documentos que están en la red.
Una URL tiene 3 partes:

- El protocolo por el que se accede al documento
- La máquina donde está el documento
- El path del documento en esa máquina (generalmente relativo)

Por ejemplo:

http://www.gva.es/index.html
file://usr/include/stdio.h

Para crear enlaces se tiene el comando **<A>** que puede tomar varios parámetros. La forma básica de hacerlo es con el siguiente parámetro:

<A HREF="URL">...</A> Convierte el texto en un enlace a esa URL
Todo lo que esté encerrado por la etiqueta se convierte en un enlace a esa URL, por lo que si encierra una imagen será también un enlace.

Si se quieren crear enlaces a lugares dentro de un documento, primero se crean puntos de referencia en el documento, y después enlaces a esos puntos:
<A NAME="nombre">...</A> Identifica al texto como punto de referencia
<A HREF="URL#nombre">...</A> Convierte al texto en un enlace a nombre

6.3. Servidores HTTP

Un servidor de HTTP (HyperText Transfer Protocol) es una aplicación que se ejecuta sobre una red TCP/IP. Su función es enviar documentos (páginas HTML, otro tipo de ficheros,etc.) a las aplicaciones clientes (navegadores) que los solicitan.

Los servidores de HTTP se pueden aplicar a entornos Internet y a entornos Intranet.

El servidor del protocolo HTTP lo implementa un proceso lanzado en modo background llamado httpd. Se puede arrancar en dos modos: como standalone o desde el internet daemon (inetd). Es preferible el modo standalone, en el que al arrancar la máquina se lanza el servidor y espera conexiones en un puerto prefijado, haciendo una llamada al sistema fork para cada petición. Este modo es mucho más rápido que si el que lanza el servidor es inetd.

Los servidores pueden suministrar a los clientes tres tipos distintos de páginas HTML, además de información Gopher o ficheros "transparentes":

- **Páginas HTML estáticas**. Páginas que existen antes de que el cliente realice la petición. El servidor se limita a enviarlas.
- **Páginas HTML dinámicas**. Se crean en respuesta a la petición del cliente. La petición del cliente incorpora un conjunto de datos que el servidor utiliza para ejecutar un script o una aplicación o realizar consultas a bases de datos. Una parte del proceso consiste en crear una página HTML que se devuelve al cliente.
- **Listados de directorios**. Algunos servidores se pueden configurar para que sirvan en formato HTML listados de los contenidos de directorios. Esto es útil en el caso de servidores que atienden también peticiones de ftp.

6.3.1. Servidores HTTP disponibles

Existen programas servidores de HTTP de dominio público y comerciales. En el campo del dominio público, los servidores más usados son los desarrollados por el NCSA (National Center for Supercomputer Applications) y el del CERN europeo. Existe otro servidor que últimamente es muy popular conocido como Apache basado en la versión 1.3 del de NCSA (A PAtCHEd server) y que implementa mejoras con respecto al servidor original en cuanto a transferencias y versatilidad.

Entre los servidores comerciales, los más utilizados actualmente son los de Netscape Corporation (http://www.netscape.com), especialmente para ordenadores con sistema operativo UNIX. En ordenadores con sistema operativo Windows NT se ha extendido la utilización del servidor de Microsoft.

Servidores de dominio público

La diferencia fundamental entre los servidores de dominio público está en que el del CERN puede actuar como proxy. Un servidor proxy es una pasarela a nivel de aplicación, que se comporta como servidor y como cliente a la vez. Esto es útil para una organización que tiene un esquema de direcciones IP privado para sus máquinas, pues les permite acceder a determinados servicios de Internet (HTTP, FTP, Wais, Correo electrónico) a través de este servidor proxy, manteniendo la privacidad de su red interna (las máquinas no pueden ser accedidas porque no tienen direcciones IP válidas). Otra ventaja del servidor del CERN es se puede implementar una caché de disco para los últimos accesos realizados por los clientes que lo usan como proxy.

La distribución del servidor del CERN es de dominio público, y puede obtenerse en la URL http://www.w3.org/pub/WWW/Daemon

La última versión disponible del servidor NCSA es la 1.5 y se puede obtener en http://hoohoo.ncsa.uiuc.edu/docs/

El servidor Apache mejora el NCSA. En concreto elimina bugs encontrados en las versiones 1.3 y 1.4. Dice ser más rápido y eficiente que las versiones de NCSA y más estricto respecto a las especificaciones de HTTP. Se puede obtener en http://www.apache.org

Existen versiones precompiladas para una variedad de plataformas y sistemas operativos. En este caso, lo único que se necesita es descargar el fichero adecuado, descomprimirlo (uncompress) y expandirlo (tar).

Si no se dispone de una versión precompilada para nuestra plataforma, es necesario descargar los ficheros fuenet y compilarlos, ejecutando previamente unos scripts que personalizan los makefiles.

En cualquiera de los casos, el servidor del protocolo HTTP lo implementa un proceso lanzado en modo background llamado httpd. En el campo del dominio público, los servidores más usados son los desarrollados por el NCSA (National Center for Supercomputer Applications) y el del CERN europeo. Existe otro servidor que últimamente es el más popular conocido como Apache basado en la versión 1.3 del de

NCSA (A PAtCHEd server) y que implementa mejoras con respecto al servidor original en cuanto a transferencias y versatilidad.

La diferencia fundamental entre estos servidores de dominio público está en que el del CERN puede actuar como proxy. Un servidor proxy es una pasarela a nivel de aplicación, que se comporta como servidor y como cliente a la vez. Esto es útil para una organización que tiene un esquema de direcciones IP privado para sus máquinas, pues les permite acceder a determinados servicios de Internet (HTTP, FTP, Wais, Correo electrónico) a través de este servidor proxy, manteniendo la privacidad de su red interna (las máquinas no pueden ser accedidas porque no tienen direcciones IP válidas). Otra ventaja del servidor del CERN es que debido al uso del proxy, se puede implementar una cache de disco para los últimos accesos realizados por los clientes que lo usan como proxy.

6.3.1.1. CERN Server

La alternativa de utilización del servidor del CERN se basa en los mecanismos de proxy y de cache en disco para las últimas consultas efectuadas (de páginas HTML, de ficheros descargados por FTP, de WAIS y de Gopher) por los clientes del proxy. El modo proxy sería necesario, si usáramos un método de direccionamiento con direcciones IP no válidas según el esquema de la RFC 1597.

La caché es una facilidad que aumenta significativamente el rendimiento en ancho de banda del servidor al evitar salidas a la red (sobre todo para páginas HTML muy solicitadas), y se puede configurar sólo cuando el servidor actúa como proxy.

Este servidor también proporciona un mecanismo de autenticación basado en direcciones IP de origen o en usuarios. Para el caso de autenticación por usuarios se necesitan unos ficheros con usuarios y claves asociadas que se mantienen con la utilidad htadm incluida en la distribución del CERN, aunque en la versión actual del servidor este mecanismo sólo es válido cuando no se utiliza el modo proxy.

Instalación

La distribución de este servidor es de dominio público, y puede obtenerse directamente en formato ejecutable para distintas plataformas en la URL http://www.w3.org.

Lo único que se necesita es descargar el fichero adecuado, descomprimirlo (uncompress) y expandirlo (tar).

Configuración

A diferencia del servidor NCSA, el servidor CERN usa un único fichero de configuración, que por defecto es /etc/httpd.conf.

En la distribución se incluye una completa documentación y ficheros de configuración de ejemplo para distintos casos:

- con autentificación de usuarios
- con proxy

- con proxy y con cache

El siquiente es un ejemplo de un fichero estándar:

```
#       Sample configuration file for cern_httpd for running it
#       as a normal HTTP server.
# See:
#

<http://info.cern.ch/hypertext/WWW/Daemon/User/Config/Overview.html>
# for more information.
# Written by:
#       Ari Luotonen  April 1994  <luotonen@dxcern.cern.ch>
#       Set this to point to the directory where you unpacked this
#       distribution, or wherever you want httpd to have its "home"
ServerRoot /opt/cern_http
#
#       The default port for HTTP is 80; if you are not root you have
#       to use a port above 1024; good defaults are 8000, 8001, 8080
Port   80
#
#       General setup; on some systems, like HP, nobody is defined so
#       that setuid() fails; in those cases use a different user id.
UserId        nobody
GroupId       nogroup
#
#       Logging; if you want logging uncomment these lines and specify
#       locations for your access and error logs
# AccessLog       /where/ever/httpd-log
# ErrorLog  /where/ever/httpd-errors
LogFormat  Common
LogTime             LocalTime
#
#       User-supported directories under ~/public_html
UserDir       public_html
#
#       Scripts; URLs starting with /cgi-bin/ will be understood as
#       script calls in the directory /your/script/directory
Exec  /cgi-bin/*    /opt/cern_http/cgi/*
#
#       URL translation rules; If your documents are under /local/Web
#       then this single rule does the job:
Pass  /*      /local/Web/*
```

Arranque y parada

El servidor se puede arrancar en dos modos: como standalone o desde el internet daemon (inetd). Es preferible el modo standalone, en el que al arrancar la máquina se lanza el servidor y espera conexiones en un puerto prefijado, haciendo una llamada al sistema fork para cada petición. Este modo es mucho más rápido que si el que lanza el servidor es inetd.

Si se quiere especificar otro fichero de configuración que el /etc/httpd.conf, se debe usar la opción -r al arrancar el demonio de HTTP:

httpd -r /otro/directorio/httpd.conf

Para arrancar el servidor en un puerto distinto del 80, por ejemplo en 8080:

httpd -p 8080

Para que los cambios realizados en el fichero de configuración tengan efecto es necesario enviar una señal HUP al proceso servidor, por ejemplo con kill -HUP process_id, o bien invocarlo con la siguiente opción:

httpd -restart

Otra opción interesante del servidor para las fases de instalación y pruebas es la -v y -vv, que significan verbose y very-verbose respectivamente. Con estas opciones el servidor muestra por la salida estándar las acciones que va llevando a cabo.

Discusión del modo proxy y la cache de disco

Las características de proxy y cache son independientes, pero la cache está derivada del proxy.

El uso del modo proxy se justifica principalmente para permitir el acceso a determinados servicios de Internet a clientes que no tienen direcciones IP válidas.

Sin embargo, en el caso del servidor HTTP del CERN, existe una razón independiente para su uso, y es la cache de disco que se puede habilitar opcionalmente cuando funciona en este modo.

Utilizando esta cache de forma directa, el tiempo de acceso de un usuario para páginas HTML compuestas de muchos campos de tamaño pequeño, no mejora significativamente, porque cada campo debe ser validado entre la cache y el servidor real de forma independiente. Para páginas con campos grandes (por ejemplo que contengan imágenes GIF grandes), este tiempo sí mejora cuando esos campos se hayan dejado previamente en la cache. Lo que se consigue en cualquier caso es una reducción del ancho de banda utilizado para las consultas.

Una posible configuración sería utilizar la cache sólo para las URLs más demandadas que sepamos que son estables, poniendo un tiempo de expiración de la cache razonable (de algunas horas), y configurar el servidor para que siempre use la cache de esas URLs. De esa forma se mejoraría tanto el tiempo de respuesta para los usuarios, como el ancho de banda utilizado.

6.3.1.2. NCSA HTTPd server

Instalación

La última versión disponible es la 1.5

Existen versiones precompiladas para una variedad de plataformas y sistemas operativos, en la actualidad están soportados los siguientes:

IRIX 4.0.5	- SGI Indigo
IRIX 5.3	- SGI Indy
SunOS 4.1.3 / Solaris 1.x	- SPARCserver 690MP
SunOS 5.4 / Solaris 2.4 x86	- Pentium 90
SunOS 5.4 / Solaris 2.4 SPARC	- SPARCstation 20
SunOS 5.3 / Solaris 2.3 SPARC	- SPARCstation 20
AIX 3.2.5	- IBM RS/6000 Model 550
HP-UX 9.05	- HP 9000 model 715
OSF/1 3.0	- Dec Alpha
Ultrix 4.3	- Dec Mips 3100
Linux 1.2.13, libc 5.0.9 ELF	- Pentium 120

Para otras plataformas, se necesita compilar los fuentes, ejecutando previamente unos scripts que personalizan los makefiles.

La instalación genera una serie de directorios:

cgi-bin
: Contiene scripts de ejemplo y binarios precompilados. Es el directorio donde se pondrán los scripts desarrollados.

conf
: Contiene los ficheros de configuración del servidor.

icons
: Iconos usados para la indexación de directorios.

logs
: Se guardan los ficheros de log y de error.

support
: Este directorio contiene los programas que se usan para control de accesos global y por directorio.

Configuración

Los ficheros de configuración son 4 (a diferencia del servidor del CERN que sólo usa 1):

- httpd.conf
- srm.conf
- access.conf
- mime.types (generalmente no hay que modificarlo)

En general hay que hecer pocos cambios en los ficheros, el servidor está casi listo para funcionar.

Hay una serie de reglas que se aplican a todos los fichreos de configuración, y que son las siguientes:

No se distingue entre mayúsculas y minúsculas excepto para paths y URLs
- Todos los comentarios empiezan por #
- Sólo puede haber una directiva por línea
- Espacios adicionales se ignoran

httpd.conf

Controla cómo se comporta el servidor, sin detalles específicos sobre los ficheros que sirve. Los parámetros más interesantes son:

```
# Usuario que corre el proceso httpd
User        http
Group           www

# A quién quejarse en caso de problemas
ServerAdmin     root0gva.es

#Path donde se pone el binario del servidor; sirve de camino relativo para los otros ficheros
ServerRoot /opt/NCSA_1.5

#Nombre del servidor (deberá tener una entrada CNAME en el DNS)
ServerName      www.gva.es

#Tipo de servidor (standalone o inetd)
ServerType standalone
```

srm.conf

Controla dónde se encuentran los ficheros y los scripts. Para una instalación sin restricción de acceso:

```
#Dónde están los documentos
DocumentRoot    /export/htdocs

# DISABLED o public_html, para servir a usuarios en sus $HOME
UserDir         DISABLED

#Para poner iconos o scripts en otros directorios
Alias           /otro/directorio
ScriptAlias     /algun/otro
```

access.conf

Fundamentalmente sirve para restringir el acceso a directorios de documentos.

Se puede hacer en base a usuarios o a grupos, que son distintos que los de la máquina.

Los usuarios se administran con la herramienta htpasswd:

```
htpasswd [-c] .htpasswd username
```

Con la opción -c crea el fichero.
Sirve para dar de alta y para cambiar los passwords de usuarios, de forma interactiva.

Los grupos se definen en un fichero de texto, con la siguiente estructura:

groupname1: user1 user2 ...
groupname2: user2 usern ...

Por ejemplo, una posible configuración para proteger el directorio /export/htdocs/privado, sería.

```
<Directory /export/htdocs/privado>
        Options Indexes FollowSymlinks
        AllowOverride     None
        Auth UserFile             /opt/NCSA_1.5/support/.htpasswd
        Auth GroupFile    /dev/null
        AuthName          Atención, acceso restringido
        AuthType          Basic
        <Limit GET>
                require user username
                require group groupname
        </Limit>
</Directory>
```

Arranque y parada

Para arrancarlo: httpd &
Admite 3 parámetros
-d directorio si ServerRoot se ha cambiado de su sitio por defecto (habitual)
-f file especifica un httpd.conf alternativo
-v muestra la versión

Para hacer un restart:

```
kill -HUP `cat logs/httpd.pid`
```

Para pararlo:

```
kill `cat logs/httpd.pid`
```

6.3.1.3. APACHE Server

Existen versiones precompiladas para múltiples plataformas.

El servidor Apache reemplaza al NCSA v1.3. En concreto elimina bugs encontrados en 1.3 y 1.4. Dice ser más rápido y eficiente que las versiones de NCSA y más estricto respecto a las especificaciones de HTTP.

Entre las ventajas objetivas que incorpora están:

- DBM databases para autentificación, se pueden proteger páginas con un gran número de usuarios autorizados elevados sin mermar significativamente el rendimiento. Permite crear respuestas a medida para errores y problemas del servidor.
- Múltiples directivas DirectoryIndex, permite por ejemplo enviar index.html o ejecutar index.cgi cuando se pide una URL.
- Número de directivas Alias y Redirect ilimitadas.

- Negociación de contenidos, permite servir a clientes que tienen diversas versiones de HTML documentos diferentes.
- Multi-homed servers, permite al servidor distinguir entre peticiones hechas a distintas direcciones IP mapeadas en la misma máquina.

Como está basado en el servidor de NCSA, tanto los directorios como los ficheros de configuración son idénticos a los descritos para NCSA.

6.3.1.4. Communications Server v1.1 de Netscape

Netscape Corporation ofrece dos servidores distintos: *Communications Server* y *Commerce Server*. El segundo es igual que el primero, excepto porque incorpora mecanismos para encriptar la información transferida.

Para el cifrado de la información, *Commerce Server* utiliza el protocolo SSL (Secure Sockets Layer). Este protocolo está implementado en todos los navegadores de Netscape. El protocolo que se debe especificar en las URLs accedidas mediante SSL es *https*.

Además, estos servidores pueden ser configurados para discriminar los accesos por la dirección desde la que se ha hecho la petición y por el documento que se ha pedido.

El servidor http de Netscape se configura en base a formularios html. Para utilizarlos es preciso disponer de un navegador, por ejemplo Netscape Navigator. Este mecanismo permite además la configuración remota de los servidores.

El servidor de administración es un demonio httpd independiente que atiende peticiones por el puerto tcp 11111 (en la configuración actual) y es preciso arrancarlo de modo independiente.

Configuración e instalación

El servidor http de Netscape se configura en base a forms html. Para utilizarlas es preciso disponer de un navegador, por ejemplo Netscape Navigator.

El servidor de administración es un demonio httpd independiente que atiende peticiones http por el puerto tcp 11111 (en la configuración actual) y es preciso arrancarlo de modo independiente (/opt/ns-home/start-admin).

Una vez establecida la conexión con el servidor de administración mediante el URL http://fallera.gva.es:11111, el servidor solicitará una pareja usuario/password para el administrador (en la configuración actual admin/nsadmin). Realizada la validación, aparecerán las forms de configuración del servidor http en las que se pueden variar los directorios de trabajo, número de procesos mínimo y máximo, etc. generándose automáticamente los ficheros de configuración admin.conf, obj.conf y magnus.conf.

En la instalación actual, los ficheros de configuración se localizan en /opt/ns-home/httpd-80/config

La configuración de los servidores de Netscape está registrada en varios ficheros:

```
# Fichero magnus.conf
Port 80
LoadObjects obj.conf
RootObject default
ErrorLog /opt/ns-home/httpd-80/logs/errors
PidLog /opt/ns-home/httpd-80/logs/pid
User nobody
ServerName sol.midominio.es
MinProcs 2
MaxProcs 4
DNS off
Init fn=load-types mime-types=mime.types
Init fn=init-clf global=/opt/ns-home/httpd-80/logs/access
```

El fichero magnus.conf se encarga de guardar la configuración de control de los servidores, en aspectos que no sean de manejo de documentos o directorios (de ello se encarga el fichero obj.conf).

Todas las líneas de este fichero tienen el formato:

Directiva Valor

Cada directiva especifica un aspecto del funcionamiento del servidor, y el formato del campo valor depende de la directiva de que se trate.

La lista de las posibles directivas y su significado:

- **ServerName**. Nombre del host.
- **Port**. Número del puerto TCP en el que el servidor escucha peticiones.
- **User**. Nombre de la cuenta UNIX propia del servidor.
- **MaxProcs**. Número máximo de procesos activos.
- **MinProcs**. Número mínimo de procesos activos.
- **ProcessLife**. Número de peticiones que que puede servir un proceso hijo antes de morir.
- **ErrorLog**. Directorio en el que el servidor guarda los registros de los errores.
- **PidLog**. Nombre del fichero en el que se guarda el identificador del proceso servidor principal.
- **LoadObjects**. Especifica el fichero de configuración de objetos.
- **RootObject**. Define el objeto por defecto del servidor.
- **Chroot**. Permite restringir los ficheros accedidos a los de un directorio, por razones de seguridad.
- **Init**. Directiva especial. Sirve para inicializar los subsistemas del servidor (por ejemplo, logs de los accesos).
- **DNS**. Servidor de DNS.
- **Security**. Sólo para el Commerce Server. Especifica el tipo de seguridad.

```
# This file obj.conf was automatically generated by the server.
# Edit at your own risk.

<Object name="default">
NameTrans from="/mc-icons" fn="pfx2dir" dir="/opt/ns-home/mc-icons"
NameTrans from="/cgi-bin" fn="pfx2dir" dir="/var/opt/ns-home/cgi-bin" name="cgi"
NameTrans root="/var/opt/ns-home/docs" fn="document-root"
PathCheck fn="unix-uri-clean"
```

```
PathCheck fn="find-pathinfo"
PathCheck index-names="index.html,home.html" fn="find-index"
ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/plain"
Service fn="imagemap" method="(GET|HEAD)" type="magnus-internal/imagemap"
Service fn="index-common" method="(GET|HEAD)" type="magnus-internal/directory"
Service fn="send-file" method="(GET|HEAD)" type="*~magnus-internal/*"
AddLog fn="common-log"
</Object>

<Object name="cgi">
ObjectType fn="force-type" type="magnus-internal/cgi"
Service fn="send-cgi"
</Object>
```

El fichero obj.conf indica al servidor cómo deben manejarse los documentos, ejecutables, etc. Es obligatorio que el fichero contenga las descripciones de dos objetos: default y cgi. Las descripciones de otros objetos pueden ser incluidas por el administrador.

```
#--Netscape Communications Corporation MIME Information
# Do not delete the above line. It is used to identify the file type.

type=application/octet-stream  exts=bin,exe
type=application/oda           exts=oda
type=application/pdf           exts=pdf
type=application/postscript    exts=ai,eps,ps
type=application/rtf           exts=rtf
type=application/x-mif         exts=mif
type=application/x-csh         exts=csh
type=application/x-dvi         exts=dvi
type=application/x-hdf         exts=hdf
type=application/x-latex       exts=latex
type=application/x-netcdf      exts=nc,cdf
type=application/x-sh          exts=sh
type=application/x-tcl         exts=tcl
type=application/x-tex         exts=tex
type=application/x-texinfo     exts=texinfo,texi
type=application/x-troff       exts=t,tr,roff
type=application/x-troff-man   exts=man
type=application/x-troff-me    exts=me
type=application/x-troff-ms    exts=ms
type=application/x-wais-source exts=src
type=application/zip           exts=zip
type=application/x-gtar        exts=gtar
type=application/x-shar        exts=shar
type=application/x-tar         exts=tar
type=application/mac-binhex40  exts=hqx

type=audio/basic               exts=au,snd
type=audio/x-aiff              exts=aif,aiff,aifc
type=audio/x-wav               exts=wav

type=image/gif                 exts=gif
type=image/ief                 exts=ief
type=image/jpeg                exts=jpeg,jpg,jpe
type=image/tiff                exts=tiff,tif
type=image/x-cmu-raster        exts=ras
type=image/x-portable-anymap   exts=pnm
type=image/x-portable-bitmap   exts=pbm
```

```
type=image/x-portable-graymap  exts=pgm
type=image/x-portable-pixmap   exts=ppm
type=image/x-rgb               exts=rgb
type=image/x-xbitmap           exts=xbm
type=image/x-xpixmap           exts=xpm
type=image/x-xwindowdump       exts=xwd

type=text/html                 exts=htm,html
type=text/plain                exts=txt
type=text/richtext             exts=rtx
type=text/tab-separated-values exts=tsv
type=text/x-setext             exts=etx

type=video/mpeg                exts=mpeg,mpg,mpe
type=video/quicktime           exts=qt,mov
type=video/x-msvideo           exts=avi
type=video/x-sgi-movie         exts=movie

enc=x-gzip exts=gz
enc=x-compress exts=z

type=magnus-internal/imagemap  exts=map
type=magnus-internal/parsed-html exts=shtml
type=magnus-internal/cgi exts=cgi
```

El fichero mime.types le indica al servidor cómo convertir ficheros discriminados por su extensión a ficheros MIME.

En general, las líneas de este fichero tienen tres campos:
- **type/subtype**. Identifica el tipo de objeto MIME.
- **exts**. Identifica la extensión de los ficheros del tipo o subtipo.
- **icon**. Indica el icono que el navegador presenta para el tipo de fichero.

Arranque y parada del servidor

El servidor http Communications Server de Netscape puede ser arrancado de tres formas diferentes:
- Automático desde inittab.

Se añade la siguiente línea al fichero /etc/inittab:

```
http:2:respawn:/opt/ns-home/httpd-80/start -i
```

- Automático desde ficheros /etc/rc*.d.

Es la manera recomendada. Para ello se crea un fichero de arranque en el directorio rc correspondiente. Por ejemplo, creando el fichero /etc/rc.d/rc.httpd:

```
#! /bin/sh
#
echo "Arrancando servidor httpd... \c"
/opt/ns-home/httpd-80/start
#
```

- Manualmente.

Desde el prompt de comando como usuario root mediante la orden:

```
# /opt/ns-home/httpd-80/start
```

Existen una serie de opciones que se pueden incluir en la línea de comando:

-p XX Arranca el servidor en el puerto especificado (XX) en lugar del indicado en magnus.conf.

-i Ejecuta el servidor en modo respawn de inittab.

Hay que tener en cuenta que el servidor debe estar parado antes de arrancarlo manualmente, de otro modo el comando fallará.

Si el servidor http está arrancado y se desea reinicializar (para que funcione con una nueva configuración, por ejemplo) se puede hacer manualmente ejecutando el script /opt/ns-home/httpd-80/restart. Este script envía una señal HUP al pid del proceso httpd padre.

Para detener el servidor http, se puede ejecutar el script /opt/ns-home/httpd-80/stop como usuario root. Si se arrancó desde inittab, hay que comentar la línea correspondiente al arranque en el fichero /etc/inittab antes de parar el servidor.

6.3.1.5. Servidor de Microsoft

Microsoft ha llamado a su servidor de WWW *Internet Information Server*. Está disponible para el sistema operativo Windows NT, sobre diferentes arquitecturas hardware (PC de Intel, Power PC, Alpha y MIPS).

La configuración de este servidor se hace a través de cuadros de diálogo del sistema operativo que ofrecen formularios y campos para rellenar. Permite la administración remota del servidor a través de un programa que se ejecuta sobre Windows NT.

Este servidor incluye, además del servidor de WWW, un servidor de FTP y uno de Gopher. Permite también el acceso a bases de datos

Si para dar servicio de WWW se necesita (o se encuentra conveniente) disponer de un servidor de DNS, es necesario disponer de un ordenador con sistema operativo UNIX, que son los únicos para los que existen estos servidores.

6.3.2. Caché

Caché es una facilidad que aumenta significativamente el rendimiento en ancho de banda del servidor al evitar salidas a la red (sobre todo para páginas HTML muy solicitadas).

Utilizando caché de forma directa, el tiempo de acceso de un usuario para páginas HTML compuestas de muchos campos de tamaño pequeño, no mejora significativamente, porque cada campo debe ser validado entre la caché y el servidor real de forma independiente. Para páginas con campos grandes (por ejemplo que contengan imágenes GIF grandes), este tiempo sí mejora cuando esos campos se hayan dejado previamente en la caché. Lo que sí se consigue en cualquier caso es una reducción del ancho de banda utilizado para las consultas.

Una posible configuración es utilizar la cache sólo para las URLs más demandadas que sepamos que son estables, poniendo un tiempo de expiración de la caché razonable (de algunas horas), y configurar el servidor para que siempre use la caché de esas URLs. De esa

forma se mejoraría tanto el tiempo de respuesta para los usuarios, como disminuiría el ancho de banda utilizado.

6.3.3. Análisis de Logs

Todos los accesos al servidor web, así como la información accedida, quedan registrados en ficheros. Los ficheros crecerán en tamaño en función de los accesos al servidor. Es conveniente (para su posterior análisis) no dejarles alcanzar un tamaño demasiado grande y generar varios ficheros de un tamaño menor. Lo razonable es proveer algún mecanismo automático que realice el cambio de fichero de log cada cierto tiempo, de manera que su tamaño nunca sea excesivo. Esto se puede hacer, por ejemplo, con un *shell script* al que se invoca desde el *cron* (en un sistema UNIX). Si se utiliza el servidor de Microsoft, en la propia configuración del servidor se especifica el plazo que transcurre entre actualizaciones de los ficheros. Con el siguiente script, para Unix, podemos cerrar un fichero de log e iniciar otro nuevo:

```
#!/bin/sh
# Rotates server log files, without affecting users who may be
# connected to the server.
# This can be run as a cron script
DATE=`date +%d-%h`
ROOTDIR=`dirname $0`
# Add here any additional logfiles you want rotated.
LOGS='access errors secure'
(cd $ROOTDIR/logs;
 for i in $LOGS; do
   if [ -f $i ]; then
     mv $i $i.$DATE
   fi
 done)
$ROOTDIR/restart
```

En general, cada servidor escribe los ficheros de log en un formato distinto. Aunque la información registrada suele ser parecida, sí que puede cambiar el orden de la misma o su formato.

Existe una gran cantidad de herramientas para realizar análisis automático de los ficheros de log de los servidores. Puesto que el formato de los ficheros de log cambia de un servidor a otro, la herramienta de análisis que se utilice debe saber interpretar el formato de los ficheros escritos por el servidor. Existen algunas herramientas capaces de entender más de un formato, si se invocan correctamente.

Hay varios analizadores de logs de dominio público. Analizan los logs de los servidores públicos para UNIX (CERN, NCSA, Apache). Como en el caso de los propios servidores, se pueden encontrar versiones precompiladas para diferentes plataformas. Si no está disponible una versión precompilada para la plataforma escogida, se pueden descargar los fuentes y compilarlos o ejecutarlos, si se trata de scripts.

Los servidores de Netscape utilizan un formato en los ficheros de log que es igual al del servidor de NCSA. Cualquier herramienta de análisis de logs del servidor NCSA puede ser utilizada con los logs de Netscape.

Netscape proporciona una serie de herramientas en el directorio /opt/ns-home/extras/log_anly para analizar los ficheros de log. La manera más cómoda de realizar el análisis es utilizar un navegador web (por ejemplo, Netscape Navigator) y utilizar el siguiente form:
/opt/ns-home/extras/log_anly/a_form.html

Rellenando los diferentes campos de este form obtenemos un fichero de resultados (en ASCII o en formato html) con los datos deseados. El form invoca a su vez al CGI a_form.cgi, y este a su vez a la herramienta analyze.

Netscape también proporciona una herramienta de análisis en modo "texto" (sin interfaz gráfico), que acepta las órdenes a través de la línea de comandos. Para que funcione correctamente la invocación al CGI se ha configurado el servidor de forma que extensión de fichero CGI sea ejecutable (en vez de agrupar todos los CGIs en un único directorio) y se ha creado un directorio /export/htdocs/analisis con los tres ficheros necesarios y un bookmark para acceder al formulario.

También se dispone de una utilidad que permite realizar el análisis en 'modo línea':

/opt/ns-home/extras/log_anly/analyze

Con la opción -h muestra una pequeña ayuda con las diferentes opciones de la utilidad:

El servidor de Microsoft permite almacenar los logs en ficheros o en una base de datos. No incorpora ninguna herramienta de análisis de estos logs.

6.3.4. Otras características de los Servidores HTTP

- **Bases de Datos para autentificación**. Se pueden proteger páginas especificando un gran número de usuarios autorizados sin mermar significativamente el rendimiento.

- **Negociación de contenidos**. Herramientas adicionales que permiten crear y servir a clientes diversas versiones HTML documentos diferentes. Útil para marketing directo y para realizar comercio electrónico y "banco en casa".

- **Multi-homed servers**. Permite al servidor distinguir entre peticiones hechas a distintas direcciones IP mapeadas en la misma máquina. Se pueden servir páginas diferentes por cada una de las direcciones.

6.3.5. Optimización para Servidores en Unix

Para cualquiera de los servidores vistos, se puede optimizar el funcionamiento de los servidores modificando los parámetros por defecto del stack TCP/IP.

Otra optimización interesante es eliminar procesos httpd lanzados por clientes que se quedan "colgados", típicamente clientes Windows. Se puede hacer un shell script que sea ejecutado periódicamente desde crontab. El funcionamiento se basa en buscar los pid de los

procesos httpd de un momento dado, esperar 190 segundos y eliminar los procesos que todavía subsistan de la lista obtenida, que serán aquellos lanzados por clientes que se han quedado "colgados". El listado de este script se da a continuación (tomado de Arthur Secret, webmaster@w3.org):

```
#!/bin/csh
# este fichero es /usr/local/bin/lanza_limpiador
#
set colgados = `ps -ef | grep ´httpd ´ | grep nobody | awk ´{print $2}ˆ
echo ´#!/bin/csh´ > /tmp/limpiador
echo "kill $colgados" >> /tmp/limpiador
echo ´ ´ >> /tmp/limpiador
chmod +x /tmp/limpiador
sleep 190
/tmp/limpiador
rm /tmp/limpiador
```

A continuación se añade en /etc/cron.d/cron.allow

nobody

y como root:

#crontab -e nobody

e insertar la línea:

0, 10, 20, 30, 40, 50 * * * * /usr/local/bin/lanza_limpiador > /dev/null 2>&1

6.4. CGI (Common Gateway Interface)

El CGI es un mecanismo que permite intercambiar información entre un servidor Web y un programa que se ejecuta en la máquina del servidor y le devuelve un resultado.
Este programa puede ser un script de shell (sh, Perl) o un ejecutable (escrito en C, C++, ...).

El programa proporciona acceso a determinada información que no está disponible en formato HTML, por ejemplo:

- otros servicios de información, como Wais
- bases de datos
- imágenes sensibles

El mecanismo de paso de información del servidor Web al programa se puede hacer de dos formas:

Por medio de argumentos en la línea de comados (método **GET**)

- Un programa shell leería $1, $2, ...
- Un programa C usaría argv y argc

Por medio de variables de entorno (método **POST**, habitual)

- Un programa shell tendría disponibles las variables
- Un programa C haría una llamada a getenv()

Formularios

Para realizar formularios con páginas WWW se necesita lo siguiente:

<FORM METHOD=“POST” ACTION=“cgi-bin/gateway_a_ejecutar”>

... definición del FORMULARIO ...
</FORM>

Esta página es presentada por el programa cliente, pero éste no analiza los datos, esto es responsabilidad del programa ejecutado en el servidor. Para devolver información, el programa escribe en la salida estándar (en HTML) y el servidor Web recoge esta salida, y enviada al cliente, donde se presenta el resultado de procesar el formulario.

6.5. Estadísticas de uso del Servidor HTTP con WUSAGE

La herramienta Internet utilizada para obtener estadísticas de uso del servidor WEB de MIDOMINIO esWusage 4.1 de Boutell.Com,Inc.

Esta herramienta identifica datos como los sitios más visitados y número de accesos a nuestro servidor, información de gran valor para las compañias que están en Internet. Los informes generados pueden ser diarios, semanales, mensuales y anuales según se le pida a la herramienta. Los informes generados son páginas html que se visualizan con el navegador de Netscape.

Wusage 4.1 tiene muchas opciones de configuración. Las más usuales son:

- Directorio donde se guardan los informes.
- Directorio donde están los logs del servidor de WEB o del proxy.
- Dominio de más alto nivel. En este caso .es.
- Formato de la fecha y dia de comienzo de la semana.
- Frecuencia de generación de estadísticas.
- Nombre de la página indice.

La generación de informes de estadísticas se suele realizar semanalmente, instalando un cron que lanza de forma automática la petición de los informes cada lunes a la 01.00h.

La generación de informes puede ser lanzada en el firewall de forma manual ejecutando:

/wusage4.1/wusage -c confgfile

Para visualizar los informes de las estadísticas de uso del proxy acceder a la dirección www://195.76.12.130/usage/usage.proxy/index.html.

Para visualizar los informes de las estadísticas de acceso al servidor de MIDOMINIO acceder a la dirección www://195.76.12.130/usage/usage.servidor/index.html.

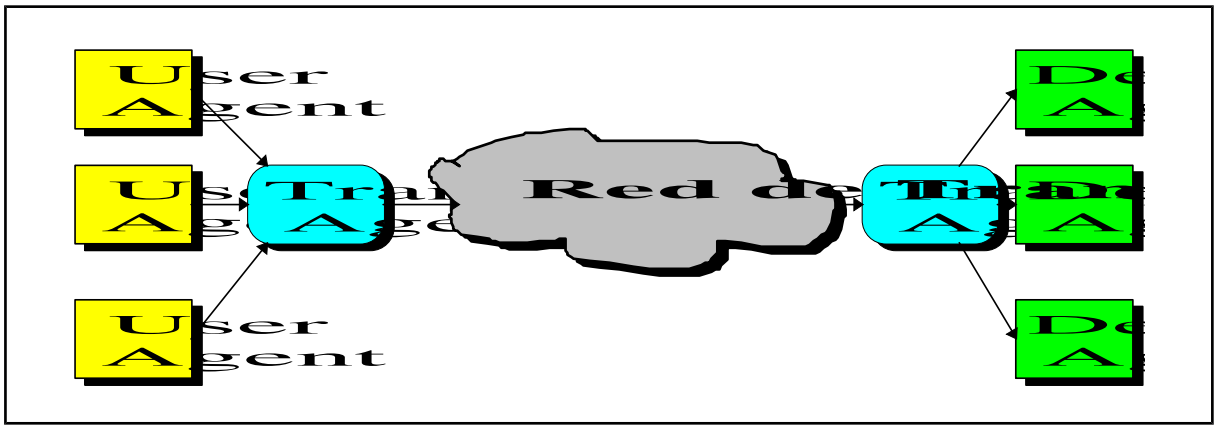
7. Servicio de Correo Electrónico

Configurar los sistemas de correo electrónico es un tema complicado. La razón de esta complejidad se debe a que el correo puede atravesar distintos tipos de redes, que usan sus propios protocolos, a diferencia del resto de servicios Internet.

7.1. Componentes de una sistema de correo electrónico

Hablando desde un punto de vista funcional, un sistema de correo tiene 3 partes:

"User Agent" Permite a los usuarios componer, leer y enviar correo
"Transport Agent" Envía mensajes entre máquinas
"Delivery Agent" Recoje los mensajes en buzones de los usuarios



User Agents

Para el correo electrónico SMTP los programas que se usan como lectores y emisores de correo son: ATT mail, BSD mail (llamado mailx o Mail según el vendedor), pine, emacs (GNU) y elm para UNIX. Y Eudora, Pegasus, Netscape o Explorer para Windows.
Cada uno tiene sus propios ficheros de configuración, a nivel de sistema y a nivel de usuario.

Transport Agents

Tienen dos funciones:

1. Aceptan correo de un UA, interpretan la dirección de destino, y entregan el mensaje a la dirección adecuada.
2. También acepta correo de entrada de otros TA

En el sistema UNIX hay muchos TA: zmailer, MMDF, smail, upas. Sin embargo el más usado es sendmail

Delivery Agents

Aceptan el correo de un TA y lo entregan a su destino, que puede ser:

- una persona
- una lista de distribución
- un fichero
- un programa

7.2. Tipos de correo electrónico

Las redes más habituales sobre las que funciona el correo electrónico son TCP/IP y UUCP.

UUCP (Unix to UNIX Copy) es el conjunto de programas desarrollados a principios de los años 70 para comunicar sistemas UNIX sobre líneas RS232, generalmente con modems y usando RTC. Lo que los programas UUCP hacen es transmitir ficheros de un sistema a otro, y mecanismos adicionales para ejecutar programas con los datos trasmitidos en el sistema remoto. El correo electrónico sobre UUCP es simplemente un

mecanismo de este tipo, en el que solicita que la máquina destino ejecute "mail" sobre un fichero de datos. UUCP solo opera sobre máquinas que hablan unas con otras via UUCP. Enviar correo sobre esta red requiere que el usuario conozca la topología de los enlaces UUCP, y de ahí el formato de las direcciones de correo de esta red.

Las implementaciones de correo electrónico más extendidas son:

- X.400 (CCITT)
- UNIX-Internet mail systems

En este documento se recoge la implementación UNIX-Internet, basado en el protocolo **SMTP** (RFC821), y a partir de ahora nos referiremos sólo a este tipo de correo electrónico.

7.3. MIME

Es el estándar oficial para el correo multimedia Internet, encapsulado en mensajes estándar. Mime se refiere al formato del cuerpo de los mensajes, no a las cabeceras, para incluir información no textual. Las facilidades incluyen:

- envío de múltiples objetos en un solo mensaje
- uso de conjuntos de caracteres distintos que el ASCII
- uso de mensajes multi-font
- inclusión de elementos no textuales, como imágenes, audio y otros

Para una descripción más detallada puede verse:

ftp.uu.net:mail/metamail/MIME-overview.txt.Z.

RFC 1341: N Borenstein & N Freed, ``Mime (Multipurpose Internet Mail Extensions) mechanisms for specifying and describing the format of Internet message bodies'' (June 1992).

7.4. Servidor de Correo Electrónico sendmail

Creado por Allman dispone de un conjunto de versiones:

V5
Es la original, escrita por Eric Allmanen (1983)
La última versión es la V5.67, se puede obtener en ftp.uu.net
Mantenida hasta 1987

IDA
Una mejora de V5, por Lennart Lövstrand de la universidad de LinKöping en suecia (1987)
Disponible en vixen.cso.uiuc.edu

V8
Mejora de Eric Allman (1993)
Disponible en ftp.cs.berkeley.edu
Versión actual 8.7.5

Toma las extensiones de IDA, su prohabla ESMTP (Extended SMTP). Incluye fichero de bases de datos Berkeley, seguridad mejorada. Usa el procesador de macros m4 para generar los ficheros de configuración.

7.5. Direcciones

Básicamente existen dos tipos de direcciones:

1. Basadas en la ruta (estilo UUCP)
maquina_1!maquina_2! ... !maquina_n!usuario

2. Independientes de la ruta (estilo Internet)
usuario@maquina.algun.dominio

7.6. Mensajes

Tienen dos partes: cabecera (header) y cuerpo (body), separados por una línea en blanco.

Cabecera del mensaje

Los campos de una cabecera de un mensaje real son (el orden puede cambiar)

From mmartinez@midominio.es Tue Mar 26 08:44:32 1996

Añadido durante la entrega del mensaje.

Received: from midominio.es (rianxo) by carballo.midominio.es (4.1/1.2); Tue, 26 Mar 96 08:44:31 GMT
Received: from mail1.midominio.es (carballo.midominio.es) by midominio.es (5.x/1.2); Tue, 26 Mar 1996
08:57:18 +0100
Received: from pc9.subdom.midominio.es by mail1.midominio.es with SMTP
(1.38.193.5/16.2) id AA18037; Tue, 26 Mar 1996 08:51:48 +0100

Estas líneas indican el camino seguido por el mensaje hasta llegar a su destino (se leen de abajo a arriba). Se van añadiendo por los Transport Agents que el mensaje va atravesando.

Message-Id: <9603260757.AA04731@midominio.es>

Identificador único de mensaje. Sirve por ejemplo para generar estadísticas de correo.

Date: Tue, 26 Mar 1996 08:51:48 +0100
To: pperez
From: mmartinez@midominio.es (Manuel Martinez)
Subject: Viaje a Brasil

Campos estándar generados por los User Agent (el subject no es obligatorio).

X-Mailer: Windows Eudora Version 1.4.4

Identifica el User Agent del que envió el mensaje

Status: RO

Es la última línea del mensaje añadida por el User Agent, y en este caso indica que ha sido leído (Read) y es viejo (Old)

Cuerpo del mensaje
Después de las líneas de cabecera viene una línea en blanco, y a continuación el cuerpo del mensaje, que puede contener otras líneas en blanco, pero en este caso no significan nada para la estructura del mensaje.

7.7. Relación sendmail-DNS

El programa *sendmail* usa el Domain Name System (DNS) para entregar correo.

Para instalar correctamente un sistema de correo electrónico, es necesario que las configuraciones del sendmail y del DNS sean correctas y estén coordinadas.

Si un mensaje se tiene que enviar a un usuario que no es local, entonces la parte del dominio de la dirección de correo tiene que examinarse para saber a qué máquina enviar el mensaje.

Los pasos que sigue *sendmail* para enviar un mensaje a *user@dominio.es*

1. pregunta al DNS resolver local, por entradas MX de dominio.es
2. Si hay entradas MX para ese dominio. En ese caso se examina el campo de preferencia, y el mensaje se envía a la máquina cuya entrada tenga un valor más bajo (más preferencia).
3. Si no hay entradas MX se buscarán las entradas CNAME o A, para intentar mandar el mensaje a esas direcciones.

La configuración de DNS para centralizar todo el correo de entrada para una organización que está dividida en subdominios es:

```
midominio.es.              IN      MX      6       mail1.midominio.es.
carballo.midominio.es.     IN      MX      6       mail1.midominio.es.
subdom.midominio.es.       IN      MX      6       mail1.midominio.es.
pc9.subdom.midominio.es.   IN      MX      6       mail1.midominio.es.
```

7.8. Reglas de reescritura de sendmail

La parte más importante de la configuración del sendmail, son los conjuntos de reglas de reescritura definidas en el fichero sendmail.cf. Estas reglas determinan:

- reescritura de direcciones *from*
- reescritura de direcciones *to*
- qué mailer usar

Las reglas se agrupan en rulesets, que están numeradas y separadas por *Sn*, donde n es el número del ruleset. La estructura de las reglas es (campos separados por tabuladores):
Rlhs rhs comentarios
donde:

lhs es un test sobre la dirección From/To
rhs es la regla que se aplica si lhs se cumple

7.9. Prácticas

- /usr/lib/sendmail -v user©algun.sitio.es

- telnet algun.sitio.es smtp

7.10. Seguridad en Correo Electronico: PGP

PGP es una aplicación de propósito general independiente del S.O. Fue introducido en 1991 y su crecimiento explosivo ha sido debido a:

- Soft. implementador de PGP es de dominio público.
- PGP corre en una gran variedad de plataformas.
- Basado en algoritmos extremadamente seguros.
- RSA: para encriptación de clave pública.
- IDEA: encriptación de mensaje.
- MD5: hash coding..

PGP ofrece tres servicios:

- Confidencialidad. Permite a un usuario, mediante encriptación, garantizar que sólamente el destinatario podrá leer el mensaje.
- Autentificación. Permite a un usuario firmar un documento antes de enviarlo, lo cual permite verificar que el mensaje ha sido firmado por una determinada persona.
- Integridad o certeza de que el documento no ha sido modificado, puesto que ha sido firmado. Si se alterara el mensaje, la firma (que depende no sólo de la identidad del remitente sino también del contenido del mensaje) no sería válida.

7.10.1. Encriptación Convencional (simétrica)

Supongamos que Pedro quiere enviar un mensaje seguro a María , Pedro coloca el mensaje en una caja y la cierra con una llave. La caja es enviada a María, nadie podría ver el mensaje pues la caja está cerrada. Cuando María coge la caja la abre con una copia de la la misma llave con la que la cerró Pedro.

7.10.2. Encriptación con Clave Pública (asimétrica)

Imaginemos una caja con un mecanismo de cierre que use dos tipos de llaves. Una llave que gire hacia la izquierda y la otra llave que gira hacia la derecha . Es decir para abrir la caja es necesario girar a derecha e izquierda y por lo tanto el mecanismo estará en el centro. Si se gira hacia la derecha o izquierda no podrá abrirse, si se cierra con la derecha se podrá abrir con la izquierda y viceversa.
Supongamos que Pedro quiere enviar un mensaje seguro a María, supongamos que Pedro tiene la llave D (derecha) y María la I (izquierda). Pedro coloca el mensaje en la caja y lo cierra con su llave D, la caja es enviada y nadie podrá abrirla excepto María que tiene la llave I.

Esta encriptación es asimétrica porque necesita dos claves distintas a diferencia de la encriptación convencional.

Supongamos que en una comunidad sus miembros se compran una caja con dos llaves D e I y que cada uno de ellos se queda con la D y que las llaves I son compartidas por todos (públicas) y colocadas en un cajetín

cada llave I con su nombre. De esta forma se tiene una confidencialidad total, ya que si Pedro quiere enviar un mensaje seguro a María lo mete en la caja con la llave I de María (llave pública izquierda de María), de esta forma solo María podría abrir la caja con su llave D (llave derecha privada de María).

Los pasos para usar PGP son los siguientes:

- Cada usuario genera una par de claves usadas para la encriptación y desencriptación de mensajes.
- Cada usuario pone a disposición pública una de las claves (clave pública), la otra se la guarda (clave privada).
- Si Pedro desea enviar un mensaje privado a María, debe de encriptarlo con la clave publica de María, para que sólo ella pueda abrirlo.
- Cuando María recibe el mensaje lo desencripta usando su clave privada. Nigún otro receptor puede desencriptar el mensaje.

7.10.3. Ventajas de las encriptación con Clave pública respecto a la Convencional.

Con la encriptación convencional se requiere que ambas partes compartan una clave secreta. Cada par emisor/receptor debería de compartir una clave secreta. Por lo tanto una de las mayores desventajas es la distribución segura de las claves entre emisor/receptor, este es el problema solventado por la encriptación con clave pública, con la cual no es necesario distribuir la clave privada. El único problema adicional es la seguridad de que una clave pública sea de verdad de quien dice ser el propietario. Más adelante veremos como se resuelve ese problema.

7.10.4. Firmas Digitales. (Hash Code)

Hemos visto cómo resolver el problema de la confidencialidad, veamos ahora como la criptografia de clave publica resuelve los problemas de autentificación e integridad.

La "Firma digital" se genera utilizando la clave privada del remitente, y un extracto del mensaje obtenido mediante hashing, de manera que depende del contenido de todo el mensaje. Este extracto es encriptado con la clave privada y el resultado es la Firma digital, que se incluye al final del mensaje.

Ejemplo: supongamos que Pedro desea enviar un mensaje seguro a María. Pedro genera una "Firma digital" de su mensaje la adjunta al final del mismo y se lo envía a María. María al ver la "firma digital" tiene la oportunidad de verificar que el mensaje ha sido enviado por Pedro. Primero, María actuando sobre el contenido del mensaje, usa la misma "hash function" que Pedro para obtener el "hash code" de ese mensaje. Segundo desencripta la firma digital usando la clave pública de Pedro, obteniendo así el "hash code". Si ambos coinciden tiene la prueba de que el mensaje no ha sido modificado. Además, dado que ha podido usar la clave pública de Pedro, tiene la garantía de que sólo él podrá haber haber encriptado la firma con su clave privada (todo este proceso es realizado automáticamente sin intervención por parte del usuario)

No se puede robar la "firma digital" de un mensaje en tránsito, pues aunque se pueda generar un "hash code" no se podría encriptar ya que no se conoce la clave privada del emisor.

No se puede eliminar o modificar el contenido de un mensaje con "firma digital", porque el nuevo mensaje tendrá un "hash code" diferente que el viejo y la firma (que incluye el "hash code" encriptado del mensaje original) no será válida.

Se puede ver el contenido del mensaje, dado que la "firma digital" proporciona autentificación y garantía de integridad, pero no confidencialidad. Para obtener confidencialidad también habría que seguir los pasos descritos más arriba en el apartado de "Criptografia asimétrica" además de los descritos en la generación de la firma digital.

7.10.5. Enviando y recibiendo mensajes PGP

El **envío** de mensajes consiste basicamente de 4 pasos:
- **Firma digital**. Este primer paso es opcional. Partiendo de un texto normal lo primero que hace PGP es la creación de una "firma digital", la cual garantiza tanto la integridad del mensaje como la autenticidad de su origen, como se ha explicado.
- **Compresión**. Este paso es automáticamente ejecutado por PGP a no ser que el usuario no desee hacerlo. Se obtiene una reducción notable del tamaño del mensaje, sobre todo si es texto. PGP usa ZIP para la comprensión. Por defecto, sólo las partes encriptadas son comprimidas.
- **Encriptacion del mensaje**. Este paso de encriptación del mensaje también es opcional. PGP utiliza el algoritmo IDEA para encriptar, combinado con RSA. Se genera una clave de sesión, aplicando el algoritmo RSA a la clave pública del receptor. Mediante el algoritmo IDEA y esta clave de sesión, se encripta el mensaje.
- **Codificación**. Tanto la firma como la compresión como la encriptación no genera un fichero texto sino binario para lo cual PGP convierte los datos binarios en caracteres ASCII (representables).

Para la **recepción** de mensajes PGP se invierten todos los pasos del proceso de envio anterior. Los pasos para recibir son:
- **Decodificación**. Paso de ASCII a binario.
- **Desencriptacion del mensaje**. Si el mensaje es encriptado, PGP recupera la clave de sesión, la cual fue encriptada usando RSA con la clave publica del receptor. Por lo tanto el receptor usará su clave privada para obtener la clave de sesión. Con la clave de sesión PGP desencripta el mensaje usando el algoritmo de desencriptacion IDEA.
- **Descompresión** del mensaje
- **Firma digital**. Si el mensaje fue firmado, PGP verifica la firma, la cual fue encriptada con la clave privada del emisor del mensaje, por lo que PGP usará la clave pública de este usuario. Se extrae el "hash code" del mensaje y PGP lo compara con el que él a calculado, si los dos encajan la firma es correcta.

7.10.6. Ejemplo de uso de PGP en la práctica

PGP mantiene para cada usuario dos ficheros, pubring.pgp (con todas las claves publicas que este usuario conoce) y secring.pgp (con su clave privada). El procedimiento para la generación de las claves se ejecuta nada más que una vez por usuario. Los siguientes ejemplos asumen un entorno UNIX. Las versiones disponibles para Windows o Macintosh incluyen interfaces de usuario para realizar las mismas funciones.

Generación de claves.

pgp -kg

Se nos pedirá una identificación (normalmente es nombre mas dirección de email), y un password. Genera un par de claves, asociadas a esa identificación y protegidas con el password.

Recepción de un mensaje

pgp fichero

Automáticamente se ejecutan los siguientes pasos:
- Si contiene claves públicas serán incluidas en pubring.pgp.
- Si contiene firmas digitales serán comprobadas, indicando si son válidas o no, y de quién son (siempre que conozca sus claves públicas correspondientes).
- Si contiene algún texto encriptado lo desencriptará (siempre que conozca sus claves públicas correspondientes).

Emisión de un mensaje.

Ejecuta alguna de las tres opciones siguientes (se necesita el password utilizado durante la generación de las claves):
- Encriptado: "pgp -e fichero"
- Firmado: "pgp -s fichero"
- Ambas cosas: "pgp -es fichero"

Si se requiere encriptación, se nos preguntará el destinatario, y se requerirá tener ya su clave pública.

7.10.7. Gestión de claves

Dado que las claves públicas están (por definición) disponibles para todo el mundo, se necesita alguna forma de asegurar su autenticidad. Es por eso que las claves pueden ser firmadas, como los mensajes. Cuando PGP detecta una clave nueva, nos muestra todas las firmas que contiene, y nos pregunta si queremos firmarla con nuestra propia firma. Debemos firmarla siempre (a menos que no exista ninguna duda de s autenticidad). Para facilitar la comprobación de claves a través de otros canales (por ejemplo, por teléfono o en persona), es posible obtener la "huella digital" (fingerprint) de una clave. Esto consiste en 16 números determinados por la clave secreta, de manera que la probabilidad de que otra clave distinta tenga los mismos números es muy baja. Hay veces en las que, aunque nosotros mismos no podamos garantizar la autenticidad de una firma, las firmas que la avalan tienen prestigio suficiente como para que no dudemos. Cada vez que PGP detecta una firma nueva, nos pregunta

también hasta que punto nos fiaríamos de la misma para que actuara de "sponsor" de otras firmas. Hay varios niveles de credibilidad: desconfianza, desconocimiento, confianza marginal y confianza absoluta. Es posible configurar PGP para definir que es lo que entendemos por una firma fiable: por ejemplo, puedo decidir que una firma será fiable si tiene al menos un "sponsor" de confianza absoluta o al menos 3 de confianza marginal.

Supongamos ahora que nos llega una clave nueva de un tal señor X, y a su vez va avalada por la firma de los señores Y y Z. Si decidimos incluir esta nueva clave pueden pasar varias cosas:
1. Supongamos que tenemos ya la clave pública de Y, como "de toda confianza" y la de Z, como "confianza marginal". Dado que hay un aval de confianza, la clave de X será aceptada sin mas.
2. Supongamos que la clave de Y es tambien de "confianza marginal". Si tenemos configurado PGP para aceptar dos confirmaciones marginales, la clave será aceptada. Si exigimos tres o mas, se nos pedirá confirmación: Key is not completely verified. Generally trusted verification from Y 'identificacion de Y...'. Generally trusted verification from Z 'identificacion de Y...'
3. Supongamos que la clave de Y y Z, o bien no son conocidas o su nivel de confianza es "desconocido'". También se nos pedirá confirmación, por ejemplo: Key is not completely verified. Questionable verification from 'unknown signator, can't be checked' Questionable verification from Z 'identificacion de Z...' En este caso, PGP no tiene la clave pública de Y, y aunque tiene la de Z, el nivel de confianza asignado a la misma es "desconocido".

7.10.8. Distribución de claves

Se puede dar el caso de que recibamos un documento avalado por PGP pero no tengamos la clave pública necesaria para comprobarlo. Hay varios formas de conseguir una clave, y la más sencilla es recurrir a un servidor de claves. Un servidor de claves esta basado en correo electrónico, y entiende comandos simples en la línea "Subject:":
* Subject: add
* Mensaje: claves publicas
Acción: el keyserver las incluirá en su base de datos

* Subject: get
Acción: nos devolverá un mensaje con la clave correspondiente a la identificación.

* Subject: mget
Acción: nos devolverá un mensaje con todas las claves cuya identificación concuerde con esa expresión regular (*=todas las del servidor)

Esto es un ejemplo, la sintaxis puede variar en otros servidores; en cualquier caso un mensaje con 'help' en el subject informara de los comandos existentes. Hay que tener en cuenta que cualquiera puede mandar claves a un keyserver, así que éste no garantiza de por sí su autenticidad (para eso esta el mecanismo de verificación que las propias claves incluyen). El keyserver actúa como base de datos, organizador y

distribuidor, y al mismo tiempo intercambia claves con otros keyservers, de manera que la información de todos ellos sea coherente.

Supongamos que una organización se dedica a firmar claves de usuarios, garantizando que éstas son auténticas. Eso es lo que se conoce como una "autoridad de certificación" (CA), sus funciones van mas allá que las del simple servidor de claves, que no deja de ser un servicio pasivo. Las CA se organizan jerárquicamente, y aunque en principio pueden asumir cualquier sistema basado en la criptografia de clave pública, su mayor utilidad es para aquellos que al contrario que PGP no incluyen un mecanismo interno de certificación de claves. El establecimiento de CA normalmente no parte de iniciativas individuales, sino que se realiza de forma coordinada con otras CA ya existentes, integrándose en la jerarquía. Finalmente, otro medio utilizado por algunas personas es el grupo de News 'alt.pgp.keydist'.

8. Servicio de NEWS

El servicio de NEWS está proporcionado por el paquete INN (InterNet News). Este servicio sirve para la propagación a través de Internet de **artículos** generados por diversos autores. Los artículos están agrupados en **grupos de noticias** que tienen una estructura jerárquica. Existen grupos no moderados a los que cualquiera puede enviar un articulo, y grupos moderados en los que existe un responsable que supervisa los artículos antes de difundirlos.

La propagación de artículos por Internet se hace a través de múltiples servidores de noticias.

El servicio de news está soportado por un **servidor** que se encarga de gestionar conexiones/envíos/recepciones con otros servidores de news. Existen unos archivos de configuración que gestionan los permisos para conectarse a otros servidores, así como para enviar o recibir artículos de/a estos servidores.

El servidor de news se encarga también de gestionar la conexión de **clientes** (usuarios). Los permisos de acceso de clientes están controlados por archivos de configuración. En general un cliente puede conectarse con el servidor, y por otra parte leer/escribir artículos.

En este documento se comentan los procedimientos de compilación, instalación y modo de funcionamiento del servicio.

8.1. Instalación Servidor INN

La instalación consiste en realizar los siguientes pasos:

1. Se ejecuta la utilidad de creación de los directorios necesarios **$inn/makedirs.sh**.
2. Se ejecuta **$inn/make update** que a su vez ejecuta *make Install* en cada uno de los subdirectorios de $inn excepto en $inn/site .
3. Se ejecuta **$inn/site/make all** que instala scripts y archivos de configuración.

Directorio de programas ejecutables:/usr/local/etc (Demonios)
/usr/local/news/bin (Scripts)
Directorio de archivos de configuración: /usr/local/news
Directorio de archivos de LOG: /var/log/news
Directorio para archivar artículos: /var/spool/news

4. Se ejecuta cd $inn, y posteriormente BUILD. Se ejecuta para asegurarnos de que todos los componentes están instalados.

Importante:

Hay que tener en cuenta que **$inn/make update** solo afecta a los ejecutables binarios y documentos de manual, no instala los archivos de configuración (control) ni los scripts. En cambio **$inn/site/make all** sustituye los archivos de configuración y scripts por los originales y esto puede ser peligroso, solo debe hacerse en la primera instalación.

8.2. CONFIGURACION INN

En este apartado se hace una descripción de la forma de trabajar del servidor de news. Se recomienda utilizar la información dada por el **man** para los distintos programas y archivos que aquí se describen.

Se puede ver el servidor de news como un conjunto de programas ejecutables, archivos de configuración y control, archivos de LOG, y archivos de artículos que se encuentran en los directorios instalados.

Los programas ejecutables utilizan los archivos de control principalmente para saber que permisos tienen los servidores con los que interactúan y los permisos que tienen los posibles clientes. Los avatares y resultados de su ejecución se reportan al *syslogd*, y a través de este a diferentes ficheros de LOG en /var/log/news. Los artículos recibidos, de otro servidor o de clientes, se almacenan a partir del directorio /var/spool/news.

8.2.1. Programas Ejecutables

Importante:

La ejecución de ciertos programas debe hacerse desde el **usuario news**.

En realidad la única ejecución que debe hacerse desde otro usuario es el script que lanza al servidor; este es "rc.news" (instalado desde el fichero original /usr/local/etc/rc.news). Esto es así porque estas utilidades pueden cambiar la propiedad de algunos archivos necesarios (LOG) en beneficio del usuario que las ejecutó. Así una posterior ejecución normal no podría acceder a ellos.

Los programas que se ejecutan en el servidor de news son:

- /usr/local/etc/innd: es el demonio principal. *innd* es lanzado por **/etc/rc.d/rc.news** a través de **inndstart**. Debe estar permanentemente en ejecución. Es el encargado de:

1-. Gestionar intentos de conexión de otros servidores. Se aceptarán o no dependiendo del contenido del fichero **hosts.nntp .**

2-. Aceptar o no artículos de otro servidor. Se puede prohibir la recepción de diferentes grupos de noticias en el archivo **hosts.nntp**.

3-. Controlar el acceso de clientes. Si una maquina no esta dada de alta como servidor en hosts.nntp, será tratada como un cliente. Este cliente será atendido de acuerdo con los permisos dados en el fichero **nnrp.access**. Cuando un cliente se conecta al servidor, innd lanza un demonio para atenderlo. Este demonio es **nnrpd**.

- /usr/local/etc/inndstart: se encarga de preparar la ejecución de innd. *inndstart* es lanzado por **/etc/rc.d/rc.news**.

- /usr/local/news/bin/innwatch: es un script lanzado a la vez que innd, sirve para controlar el estado del disco ante la posibilidad de que los artículos lo inunden. Actualmente lo comprueba cada 10 minutos, esto es configurable en el fichero **/usr/local/news/innshellvars**. *innwatch* es lanzado por **/etc/rc.d/rc.news.** Debe estar permanentemente en ejecución.

/usr/local/etc/nnrpd: es un demonio lanzado por innd que permite a los clientes acceder al servicio. Existe un limite en el número de conexiones NNTP simultáneas. Este límite se indica con el flag "-i" en la llamada a innd y su valor está especificado en /etc/rc.d/rc.news ("-i0" indica no tenerlo en cuenta). Se recuerda que los permisos de acceso para los clientes se encuentran en el archivo **nnrp.access.** *nnrpd* es lanzado por innd cada vez que accede un cliente.

- /usr/local/news/bin/nntpsend: es el programa encargado de enviar artículos generados hacia otros servidores. La información relativa a qué servidores pueden recibir grupos de noticias, qué grupos, y en qué archivo se encuentra la lista de artículos a enviar, está en fichero **/usr/local/news/newsfeeds.** Los artículos se pueden enviar a distintos tipos de destino; ver el manual de newsfeeds. La dirección de los servidores a los que debe conectarse se encuentra en el archivo de control **nntpsend.ctl.** *nntpsend* se ejecuta periódicamente utilizando el *crontab* del *root* haciendo 'su news' previamente. En la configuración actual, se ejecuta cada hora. Con el contenido actual de *newsfeeds*, **nntpsend** trabaja como sigue:

1.- Al arrancar nntpsend lee el contenido de *newsfeeds* , este le indica que en el directorio */var/spool/news/out.going* existe el fichero *servnews.* En este archivo se encuentra una lista de artículos (con path a partir del directorio /var/spool/news) que deben ser enviados a la dirección especificada para servnews en *nntpsend.ctl*. Por otra parte nntpsend deja trazas de ejecución en **/var/log/news/nntpsend.log.**

Nota: Se ha añadido un filtro previo a nntpsend para que antes de enviar artículos al exterior modifique en la cabecera de los artículos la linea *NNTP-Posting-Host* . La función de este filtro (/usr/local/news/bin/filtro.awk) es cambiar este campo para que tome siempre el valor *rianxo.midominio.es* y no sea la dirección IP de cada cliente o el nombre completo FQDN (p.e. pc37.midominio.es) caso de que el fichero */etc/hosts* tuviera toda la información local. En principio esto no sería necesario si el servidor utilizará la información contenida en el archivo *inn.conf*; archivo que contiene diversos parámetros (nobre de la compañía, del servidor etc.) utilizados para rellenar la cabecera de los artículos. Esta posibilidad no está implementada en la versión 1.4 de Inn.

- /usr/local/news/bin/news.daily: este script es una utilidad que genera informes a partir de los ficheros de LOG y de innwatch, enviándolos por mail al NEWSMASTER,

especificado en **innshellvars.** *news.daily* se ejecuta diariamente utilizando el *crontab* del *root* haciendo 'su news' previamente.

- /usr/local/news/bin/scanlogs: devuelve un sumario de los archivos de LOG. Es utilizado por news.daily pero puede ser ejecutado expresamente. Si no se le añade la opción *norotate* vaciará los archivos LOG. Debe ejecutarse desde el ususario news.

- /usr/local/news/bin/ctlinnd: programa para enviar mensajes de control a *innd* . Útil para **crear y eliminar grupos de noticias**, hacer *reload* de los ficheros de configuración modificados etc.

8.2.2. Archivos de Control y Configuración

En el apartado anterior ya se ha comentado la función de los archivos de control y su utilización por parte de las aplicaciones. En esta sección se describen los contenidos de cada uno de los ficheros.

En la mayoría de los archivos de configuración los diferentes campos con significado propio están separados por el signo de dos puntos ":" .

- /usr/local/news/innshellvars: contiene principalmente path´s y algunas variables, utilizados por los programas script de news. Indica, por ejemplo, quién es el *newsmaster* al que serán enviados los mail de control.

- /usr/local/news/hosts.nntp: indica a **innd** qué otros servidores pueden enviarnos artículos, si necesitan password y qué artículos aceptaremos. En nuestro caso su contenido es:
servnews.eunet.es: indica que las peticiones de conexión desde *servnews.eunet.es* para *suministrarnos* artículos serán aceptadas sin necesidad de password. Además no hay restricciones respecto a los grupos que *servnews* puede enviarnos.

/usr/local/news/nnrp.access: señala a **innd** qué clientes pueden conectarse al servicio de news, con qué derechos (lectura/escritura), nombre de usuario, password para este, y grupos de noticias a los que puede acceder. En nuestro caso su contenido es:
:: -no- : -no- :!
Condiciones iniciales, nadie se conecta, no se pueden leer ni presentar artículos.
150.1.*.*:R P:::*
1.- Se pueden conectar los clientes con dirección IP del tipo 150.1.X.X .
2.- Tienen derecho a leer (Read) y presentar (Post) artículos.
3.- No es necesario identificación de usuario (campo vacío "::") .
4.- No se necesita password, obviamente, (campo vacío "::") .
5.- Lo anterior es valido para cualquier grupo de noticias (*) .
.midominio.es:R P:::
Idéntico al anterior excepto por la notación de las direcciones(FQDN). Esta notación es válida si el archivo *hosts* del servidor (rianxo) contiene todos los posibles clientes.

- /usr/local/news/newsfeeds: indica a **nntpsend** a qué otros servidores debemos suministrar artículos, qué artículos no se deben suministrar al exterior, qué tipo de

destino es, qué información escribir en la lista de artículos a enviar y cual es el nombre del archivo que contiene esa lista. En nuestro caso:

servnews::Tf,Wmn:servnews

Indica que:

1.- Debemos suministrar artículos a *servnews*.
2.- No hay restricciones en los grupos a suministrar (campo vacío "::")
3.- Al destino se le alimenta a través de un ficheros (subcampo "Tf").
4.- En la lista de artículos a enviar a servnews se escribirá el identificador del articulo a enviar y el path hasta el articulo, este path es relativo al directorio de spool (/var/spool/news en este caso) (subcampo "Wmn")
5.- El nombre del archivo donde está la lista de artículos a suministrar es "servnews " (/var/spool/news/out.going/servnews)

- /usr/local/news/nntpsend.ctl: indica a **nntpsend** cuales son los servidores a los que se deben suministrar articulo y su dirección IP o FQDN. Puede haber más campos. En nuestro caso:

servnews:servnews.eunet.es

Advierte a nntpsend que los artículos para el servidor *servnews* deben ser enviados a la dirección *servnews.eunet.es* .

- /usr/local/news/active: señala cuales son los grupos de noticias activos en cada momento. Se entiende por *activos* los grupos de noticias que nuestro servidor recibe. Si un grupo no se encuentra en el fichero *active,* cualquier artículo presentado por un cliente a este grupo será ignorado.

!Importante! : siempre deben estar activos los grupos "control" y "junk"

-/usr/local/news/control.ctl: es una lista de los usuarios que pueden enviar ordenes de control al servidor de news. Ordenes como creación y borrado de grupos de noticias vía correo electrónico. No ha sido modificado. Actualmente sólo tienen permisos de creacion y eliminación de grupos (*doit=newgroup, doit=rmgroup*) los mensajes provinientes de la dirección **tale@*.uu.net**. Los mensajes de creación y eliminación serán notificados vía correo al administrador de news aunque no sean ejecutados. En general, es el administrador el responsable de crear o eliminar grupos de noticias (ver *ctlinnd*). El formato seguido en este archivo es el siguiente (campos separados por ":"):

newgroup:tale@*.uu.net:comp.*|misc.*|news.*|rec.*|sci.*|soc.*|
talk.*:doit=newgroup

1.- Orden *newgroup*
2.- Procedente de *tale@*.uu.net*
3.- Respecto a los grupos *comp.* misc.** etc.
4.- ejecutarla. También se enviará un mail al administrador.

Los siguientes archivos de LOG reciben trazas por indicación de control.ctl : control.log, miscctl.log, newgroup.log, rmgroup.log, unwanted.log.

- /usr/local/news/expire.ctl: señala los plazos de expiración para los artículos según el grupo. Los artículos pueden llevar incluido un plazo de expiración. En nuestro caso contiene:

/remember/:14

-. Establece que el identificador de un articulo debe ser conservado 14 días después de

que este ha expirado.

***:A:1:14:21**

1.- " * " indica que esta linea es válida para todos los grupos, sea cual sea su nombre.
2.- " A" indica, además, para todos los grupos, Moderated y Unmoderated.
3.- " 1 " plazo mínimo de retención de un articulo. En días.
4.- "14" plazo asignado para artículos que no tengan fecha de expiración.
5.- "21" plazo máximo de retención de un articulo, aunque su cabecera indique mas.

/usr/local/news/inn.conf: establece varios parámetros locales que son utilizados por disitintos programas del servicio de news. La mayoría son valores para la cabecera de los artículos. En nuestro caso hemos establecido los siguientes valores para la cabecera:
domain: midominio.es
fromhost: rianxo.midominio.es
pathhost: rianxo.midominio.es
organization: MIDOMINIO
server: rianxo.midominio.es
En principio este archivo debería permitir esconder a los clientes detrás del servidor incluyendo en la cabecera de los artículos NNTP-Posting-Host: rianxo.midominio.es . Pero en la versión actual, 1.4, **esto no está implementado**. Además varios de los campos mencionados por *man inn.conf* no se encuentran comentados en el archivo inn.conf (mime-version, mime-contenttyp, mime-encoding) .

8.2.3. Directorio de Artículos

Los artículos y cierta información de control se almacenan en una jerarquía de subdirectorios a partir del directorio **/var/spool/news**.

Los artículos se almacenan en subdirectorios creados para cada grupo. Como los grupos tienen estructura jerárquica los subidrectorios siguen esta jerarquía; por ejemplo los artículos para el grupo "abg.amiga" se almacenan en el directorio /var/spool/news/abg/amiga, mientras que los artículos del grupo "abg.atari" se guardan en el directorio /var/spool/news/abg/atari.

Por otro lado hay ciertos subdirectorios de control.

- /var/spool/news/out.going: en este directorio se mantiene un archivo por cada servidor al que *rianxo* puede suministrar artículos. Estos archivos contienen la lista de artículos que **nntpsend** debe enviar a otros servidores. El nombre de estos archivos y la clase de información que se le suministra está especificada en el archivo de configuración **newsfeeds** . En nuestro caso este archivo es "servnews". Por tanto *"/var/spool/news/out.going/servnews"* contiene la lista y localización de los artículos que *nntpsend* debe enviar a *servnews.* Una vez enviados, el archivo se deja vacio.

- /var/spool/news/in.coming: directorio donde se indican los últimos artículos recibidos de otros servidores.

8.2.4. Archivos de LOG

Las aplicaciones de news *innd, nntpsend, innwatch* etc. aportan información a través de syslog en los archivos del directorio **/var/log/news** (ver syslog.conf) .

La utilidad **news.daily** toma diariamente información de estos archivos para generar sus informes llamando previamente a **/usr/local/news/bin/scanlogs**.

Si se quiere consultar la información de estos archivos sin modificarlos debe añadirse la opción *norotate* en la ejecución de *news.daily* o de *scanlogs* (deben ejecutarse **siempre bajo el usuario *news***).

La utilidad news.daily *resetea* los archivos almacenando la información (comprimida con ZIP) antigua en el directorio **/var/log/news/OLD** . Aquí las informaciones diarias se van numerando de manera que los números mayores indican mayor antigüedad.

Los archivos de log que se mantienen son los siguientes:

- *news.notice:* donde se recibe información de error y de estado desde *innd*. A su vez, innd puede recibir información de otros programas como nntpsend. Traza de llegada de un artículo (ver *man innd*):
mon dd hh:mm:ss.mmm + feed <Message-ID> site...
Los tres primeros campos indican la fecha de llegada del artículo. El cuarto campo indica si el articulo fue aceptado (+). Si fue aceptado pero almacenado en el grupo "junk" (j). Si se recibió un mensaje de cancelación antes de que este fuera aceptado (c).Si fue rechazado (-). El campo feed indica el servidor que nos suministra el artículo. Si el artículo fue aceptado "site" indica una lista del resto de servidores a los que es enviado el artículo. Si el artículo fue rechazado se indica la razón.
El demonio innd deja trazas de su propio estado con mensajes del tipo:
Nov 2 23:45:39 rianxo innd: ME running
Trazas de utilización de archivos de alimentación a otros servidores (servnews) :
Nov 3 00:00:02 rianxo innd: servnews opened servnews:16:file
Presentación de artículos por parte de clientes:
Oct 30 10:35:40 rianxo nnrpd[23534]: 150.1.2.137 post ok <30952698.27F9@mail.midominio.es>
Envío de artículos a otros servidores:
Oct 30 11:00:06 rianxo innxmit[23891]: servnews.eunet.es stats offered 1 accepted 1 refused 0 rejected 0

- *nntpsend.log:* utilizado específicamente por *nntpsend.* Deja trazas de los envíos de artículos a otros servidores. Ejemplo de conexión con servnews (servnews.eunet.es):
nntpsend: [20983:21007] begin servnews.eunet.es Fri Nov 3 11:46:25 MET 1995
nntpsend: [20983:21007] innxmit -a servnews.eunet.es ...
nntpsend: [20983:21007] end servnews.eunet.es Fri Nov 3 11:46:27 MET 1995

- *news*: donde se informa de artículos recibidos desde otros servidores y/o clientes. Ejemplo:
Nov 3 11:39:13.202 + Postmaster <47crgg$k7c@rianxo.midominio.es> servnews
1.- Fecha: "Nov 3 11:39:13.202 "
2.- Estado: "+" Aceptado (ver *man innd* o apartado news.notice).

3.- Origen: Postmaster (información extraida de la cabecera del artículo)
4.- Identificador del artículo.
5.- Servidor donde enviarlo, "servnews".

- news.err: contiene los errores importantes proporcionados por innd.

- news.crit: para errores críticos de innd.

- errlog: errores de demonios o scripts lanzados por innd.

- expire.log: información sobre el trabajo realizado por el programa expire. Este script es lanzado por news.daily y es utilizado para purgar los artículos que hayan expirado.

- expire.list: lista de articulos eliminados por haber expirado.

- unwanted.log: mantiene la cuenta de artículos que no fueron aceptados por no existir ese grupo de noticias.

Para más información ver "man newslog" y "man -s 5 newslog"

8.3. CONFIGURACIÓN DE CLIENTE NETSCAPE

La configuración de Netscape para utilizarlo como cliente de NEWS es bastante simple:

Netscape para Unix (versión 1.1):

Ver ventana *options* , *preferences, news and mail* y actuar en consecuencia.

Netscape para Windows:

Ver ventana *options* , *preferences, news and mail* y actuar en consecuencia.
Ver fichero *fat* (generalmente c:\netscape\news\fat). En él se definen los servidores de news a los que se accede.

Netscape suele traer un servidor de news por defecto (generalmente llamado *news*) que, como es natural, no nos sirve y es preciso eliminar del archivo *fat*.

8.4. REFERENCIAS A INN Y NEWS EN INTERNET

Ref 1 : ftp://ftp.uu.net/networking/news/nntp/inn
Ref 2 : Guía de Norman J. Pieniazek en:
http://www.cis.ohio-state.edu/hypertext/faq/usenet/inn-faq/part4/faq.html
Otras localizaciones:
ftp.univ-lyon1.fr/pub/unix/news/inn,
munnari.oz.au/pub/news/inn,
src.doc.ic.ac.uk/computing/usenet/software/transport
Parches para INN en :
ftp://ftp.pop.psu.edu/pub/src/news/inn/patches
Contribuciones a INN en:
ftp://ftp.pop.psu.edu/pub/src/news/inn/contrib

9.- SEGURIDAD.

La prestación a través de Internet de cualquier servicio entraña diversos riesgos que es necesario contrarrestar para mantener la seguridad del servicio y de las redes privadas de la organización que lo presta.

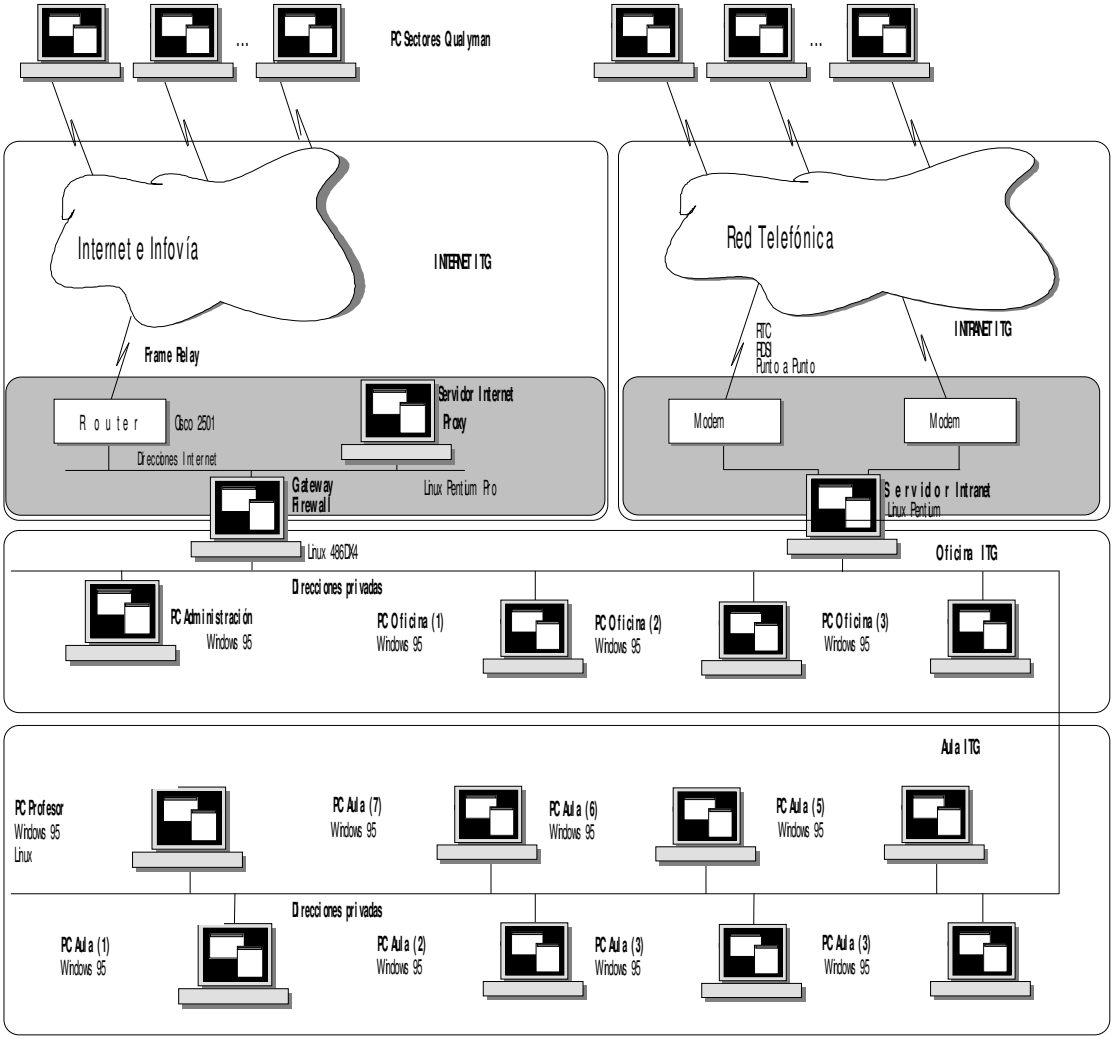
Si ya disponemos de una red corporativa y el ordenador en el que van a residir los servicios está conectado de alguna manera a ella es necesario salvaguardar la red corporativa de la posible entrada de intrusos.

En segundo lugar, el propio servidor debe estar protegido contra las intrusiones, preservando la información contenida en él y los procesos que atienden los servicios que ofrece.

Con el fin de asegurar tanto el servidor como la red corporativa se ha definido una arquitectura consistente en el establecimiento de una red de “interposición” entre Internet y la red privada de la organización.

En este caso, el acceso de un intruso procedente de Internet está dificultado por la existencia de un equipo que realiza un filtrado de los paquetes que los atraviesan en cualquier sentido.

A este equipo de interposición se conecta por un lado el servidor que va a ofrecer servicios a la comunidad Internet, y por otro los clientes de la red privada interna (ver figura 1).



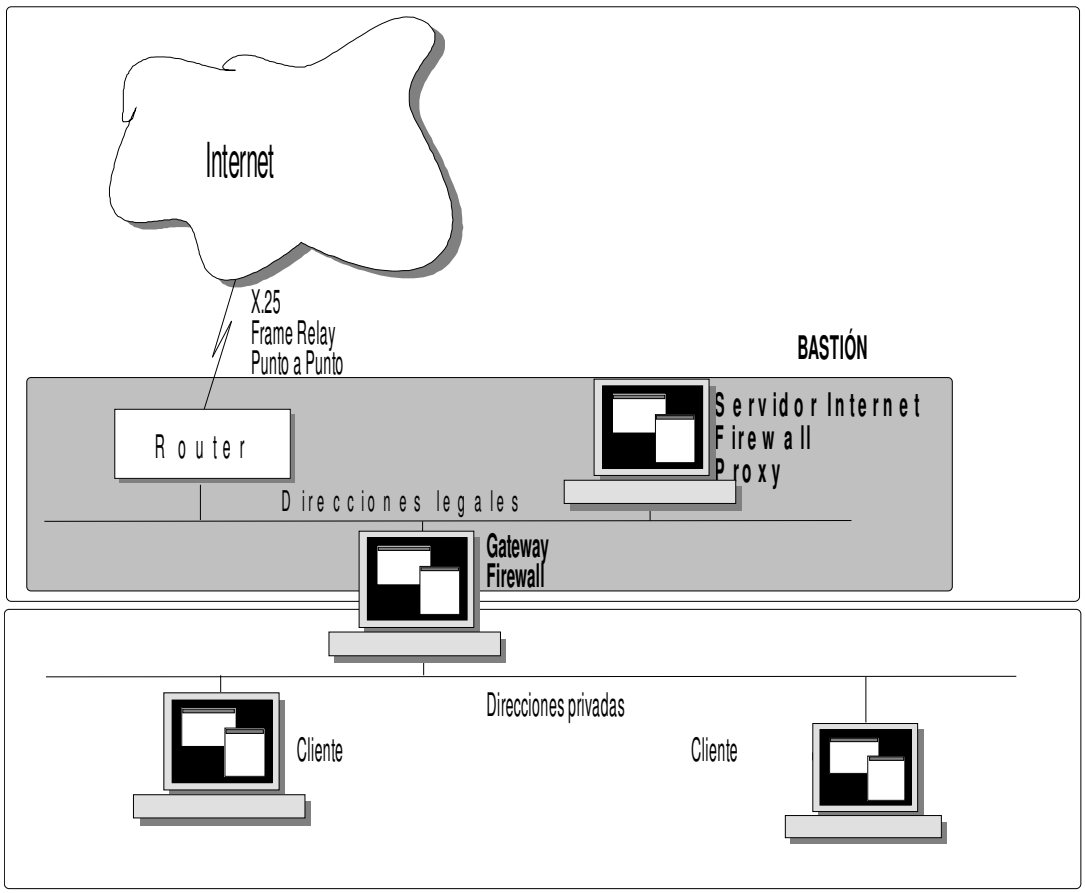


Figura 1. Sistema de seguridad basado en un router y dos firewalls.

El *router*que se conecta a Internet cumple dos funciones principales:

Es la vía de comunicación entre el proveedor de acceso a Internet y el servidor de la organización. Para realizar esta función debe estar provisto de todo lo necesario para conectarse tanto a la red de interposición como al enlace con el proveedor (X.25, Frame Relay, línea punto a punto).

- Efectúa un filtrado de los paquetes que van y vienen a/desde Internet (cuyas normas dicta la administración de la red privada) basándose en las direcciones IP y en los protocolos, evitando accesos indeseados de muy diversos tipos.

Para aumentar la seguridad del conjunto, el *router*sólo conoce la existencia del servidor Internet. De esta forma, en caso de intrusión, el intruso no podrá conocer la dirección del *router*que se encuentra entre esta red y la red privada. El *router*que se conecta a Internet debe conocer al menos la dirección del servidor Internet para poder desempañar su función.

Otro elemento dentro de la red es el servidor Internet. Esta ordenador cumple varias misiones importantes:

Es el encargado de ofrecer a Internet los servicios que provee la organización (World Wide Web, FTP, Telnet, etc.). En ella residen los procesos y los datos necesarios para estos proporcionar estos servicios.

- Se ocupa de atender las solicitudes de servicios provenientes de la red privada.
- Tiene como misión actuar de servidor de correo electrónico Internet, enlazando con el correo interno de la organización.
- Actúa también como servidor de grupos de noticias, enlazando con servidores externos a la organización.
- Debe disponer de un software de seguridad que actúa en conjunción con el software que conoce las direcciones privadas de la red corporativa. Este software de seguridad filtra las demandas de Internet y de la red privada, autorizando o desautorizando determinado tipo de peticiones según reglas fijas, establecidas por el administrador del sistema. Este elemento software es el que se conoce como *Firewall*(Cortafuegos).

En Internet muchos de los servicios se proporcionan a usuarios no identificados. Pero esto no implica que los servidores se vuelvan inseguros. La correcta configuración de un software *Firewall*permite prestar estos servicios a la vez que filtra las demandas externas de forma que sólo se atiendan de una forma segura.

Además, los *Firewalls*son útiles como herramientas estadísticas y de registro de los accesos que han tenido lugar sobre nuestros servicios prestados a Internet.
El servidor, debido a que efectúa un intercambio de datos permanente tanto con Internet como con la red privada, debe tener conocimiento de la existencia de ambas, y concretamente de las direcciones de los dos *routers*que se encuentran enlazados con el bastión.
La sensibilidad de este sistema es clara, de ahí la necesidad del *router*de conexión a Internet de que se habló anteriormente. Una buena configuración de este *router*puede impedir el acceso de mensajes enviados a explorar el servidor Internet en busca de agujeros de seguridad.

Direcciones IP

Para que los servidores de Internet sea accesible desde el exterior es necesario que se le asigne una dirección IP "legal", esto es, una dirección exclusiva concedida por una autoridad.
Por la escasez de direcciones IP que sufre ahora Internet, no es posible que todos los ordenadores de todos los usuarios de una institución dispongan de direcciones legales para acceder a los diferentes servicios. La única solución es asignar a esos ordenadores direcciones no legales, y utilizar un servidor de *proxy*. El servidor de *proxy* actúa como intermediario entre los ordenadores de los usuarios e Internet, ocultando las direcciones no legales.
Por otra parte, la utilización de direcciones no legales internamente aumenta la seguridad, al convertir en "invisibles" desde el exterior los ordenadores internos.

Para poder implantar los servicios de correo y noticias, y para que los usuarios de una institución puedan acceder a servidores de Web en Internet, y a otros servicios, es necesario que el ordenador servidor de Intranet también disponga de una dirección IP legal.

El segundo de los *routers*, que da acceso a la red local de la organización, tiene a su cargo las siguiente misiones:
Poner en comunicación la red donde residen los servidores Internet con la red corporativa.

- Realizar un filtrado de paquetes provenientes de cualquier dirección de Internet distinta de la del servidor Internet.
- Opcionalmente, realizar un filtrado de paquetes que van desde el interior de la red corporativa a cualquier dirección distinta de la del servidor Internet.

Una solución equivalente a la anterior consiste en la sustitución del router de interconexión de las dos redes de área local por un equipo pasarela (*gateway*) entre ambas, dotándole también de funciones de cortafuegos (*Firewall*).
Esta solución permite hacer un filtrado y control de los paquetes en los niveles bajos, como la solución anterior, y a la vez permite el filtrado a nivel de aplicación (por puerto o servicio).
En esta configuración el equipo que realiza las funciones de pasarela puede ser asimismo el que realice las funciones de control y monitorización de la red. La arquitectura sería la descrita en la figura 2.
En ambos casos, conviene distinguir entre la red de acceso público (con direcciones registradas o *legales*), la red de acceso privado (con direcciones registradas o *privadas*) y la pasarela entre las dos redes.

Es posible una tercera configuración basada en la anterior, colocando el servidor Internet en el lugar del equipo pasarela. Sin embargo, esta configuración es más insegura, puesto que el router de conexión a Internet debe conocer la dirección del servidor: en caso de intrusión se conocería entonces la dirección del equipo que da acceso a las redes privadas.
El **router de conexión al exterior** desempeña una doble misión. Por una parte, sirve para conectar las redes de la sede con el exterior, y en particular, la red a la que está conectado el ordenador Servidor de Información. Por otra, sirve para incrementar la seguridad de las redes internas y la integridad de los servicios de información realizando un filtrado de los paquetes IP procedentes del exterior. Los paquetes se filtran según su dirección de destino, impidiendo el acceso a otros ordenadores que no sean el servidor de información.
La **Plataforma de Seguridad** cumple tres funciones. La principal es contribuir a la seguridad de las redes internas y a la integridad de los servicios de información filtrando los paquetes IP

que procedentes del exterior llegan a través del router de conexión al exterior. En el capítulo anterior ya se ha explicado en detalle el papel que una plataforma de seguridad de este tipo desempeña en la conexión de una organización con Internet.
La segunda función de la Plataforma de Seguridad es controlar el acceso a Internet desde las redes internas. El acceso al exterior se puede autorizar o denegar dependiendo del usuario que lo solicite y del servicio pedido.
En tercer lugar, la Plataforma de Seguridad actúa como *gateway* (pasarela) entre las redes internas, la red que la conecta con el router de acceso al exterior y la red a la que está conectado el Servidor de Información.
El **Servidor de Información** es el ordenador encargado de soportar las aplicaciones que posibilitan la prestación de los diferentes servicios de información. En este ordenador se instalarán las aplicaciones servidoras de los diferentes servicios, las páginas HTML con información corporativa o de marketing y los programas que posibilitarán el acceso de los usuarios a las bases de datos.

9.1. Seguridad en Sistemas UNIX

seguro: algo que está libre de riesgo o daño, o también algo que está garantizado contra pérdidas
seguridad: cualidad de ausencia de peligro, miedo o ansiedad; también significa protección, medidas que se toman para proteger contra el espionaje o el sabotaje.

Todas estas definiciones se aplican a los sistemas de información. En concreto lo que aquí nos interesa es la pérdida de datos y evitar los accesos no autorizados.

En cuanto a la pérdida de datos se estima que en un 80% de los casos se debe a desastres (errores humanos, fuegos o agua) y sólo en un 20% a razones relacionadas con la seguridad. Además sólo un pequeño porcentaje de éste está relacionado con accesos malintencionados, de los que la mayoría son de empleados de la propia compañia o de antiguos empleados. El resto se debe a intentos de sabotaje desde el exterior, que son los que nos interesan.

Internet fue concebida en sus inicios como una asociación de redes de investigación. Ninguna preocupación por la privacidad de la información.

Unix era un S.O. creado para investigadores por investigadores. Ninguna preocupación por la seguridad. Sólo dos niveles: usuario y root.

Los mecanismos de Internet y de Unix (protocolos, comandos, etc...) no están concebidos para dar seguridad con sencillez. En general es posible conseguir un alto grado de seguridad en los datos pero existe un compromiso entre

- Seguridad
- Conveniencia

Un sistema muy fácil de emplear suele ser muy frágil e inseguro. Un sistema con un alto grado de seguridad frente a accidentes o ataques puede ser muy engorroso para los usuarios autorizados.

Amenazas a los datos:

a) Datos almacenados

- Destrucción
- Alteración
- Espionaje.

b) Datos comunicados

- Impersonación de una de las partes, o de las dos ("man in the middle"), introduciendo datos falsos.
- Escucha de la comunicación.
- Negación de servicio

Pueden producirse:

- ***De manera accidental***
- Por ataques maliciosos.
- La mayor parte de los ataques (80%) son realizados por personal de la propia organización. Existe una cierta paranoia respecto a los ataques de sangrientos hackers quiceañeros. En realidad La mayor parte de las pérdidas de datos no son intencionadas. ("¡Dios mío, he borrado todos los ficheros del último año!")

9.1.1. Seguridad Física

La seguridad es un concepto amplio . No merece la pena construir una muralla inmensa en una ciudad a la que se puede entrar por un túnel sin protección, o que no cuenta con medidas aceptables contra los incendios. No hay que olvidar que la mayor amenaza contra los datos son los errores propios.

Riesgos físicos accidentales:

- ***Caída de energía eléctrica: Además de la información, también puede dañar al hardware. Se previene mediante la instalación de UPSs***
- Fuego: Sistemas antiincendios.
- Inundación y goteras: Evitar los sótanos o colocar el hardware sobre mesas. Contra las goteras lo mejor es tener a mano plásticos.
- Terremoto, atentado, guerras... Estos riesgos amenazan a todo el edificio.
- Errores del operador: borrado de ficheros, derrames de café...
- Interrupción de las comunicaciones
- Errores del software/hardware

Backups: Son la mejor protección contra accidentes y catástrofes.

- ***Backups de los sistemas de ficheros. IMPRESCINDIBLE.***
- Hardware
- Conexiones y comunicaciones, incluso proveedores.

- Emplazamiento: para prevenir una interrupción en servicios muy sensibles se habilitan lugares alternativos:
  - ***"Cold Sites": Emplazamientos preparados para instalar los ordenadores. Todo lo demás está presente y preparado.***
  - "Hot sites": Emplazamiento completamente preparado, incluyendo ordenadores y software, para ser encendido y suplir al servidor primario.

Riesgos físicos no accidentales:

- Intrusión. Se previene con defensas físicas: guardias, cerrojos, cámaras...
- Lectura de material de deshecho. Las cintas, discos, etc que sean tirados o cedidos deben ser sobreescritos con ceros.
- Lectura de información a través de radiaciones electromagnéticas . "Pinchazos". Se previene mediante el apantallamiento de los sistemas.

9.1.2. Identificación

Para identificarse un usuario debe demostrar que él es quien dice ser. Para esto debo suministrar al sistema...

- Algo que sé. p.ej: Passwords
- Algo que tengo. p.ej.: una tarjeta
- Algo que soy (o como se comporta). p.ej.: biométrica

Passwords

Es el principal mecanismo de seguridad en Unix. Tiene dos importantes condiciones:

- Ser fácil de recordar
- Ser difícil de adivinar

Estas dos condiciones son contradictorias, este es el problema principal de las passwords. Existen distintos métodos para forzar al usuario a emplear passwords difíciles de averiguar.

Una posibilidad alternativa son las passwords variables.Al usuario se le facilita una lista de passwords, debiendo utilizar una por vez.

Tokens

Es un mecanismo de autentificación basado en *tener* algo.

- Tarjeta de identificación: requiere un lector especial
- Máquina de santo y seña. Cuando se intenta una conexión remota el ordenador envía al usuario un código. El usuario introduce el código en una máquina pequeña, dotada de un teclado y un display. La máquina realiza una transformación única de ese código y devuelve un resultado, que el usuario a su vez envía al ordenador.

Biométrica

Se basa en reconocer a un usuario según alguna característica física

- Dinámica de la firma
- Forma de la mano
- Patrón de la voz
- Fondo de retina, etc...

Problema: Existe una cierta probabilidad de error en el reconocimiento.

Hemos visto algunas vaguedades sobre el tema de la seguridad y la autentificación. Ahora vamos a ir a lo que realmente nos interesa. ¿Puede entrar alguien en nuestro sistema? ¿Cómo lo va a hacer? Vamos a hacer un repaso de los puntos débiles más empleados por los atacantes a un sistema.

Objetivos de un ataque

- Conseguir una cuenta de usuario
- Conseguir derechos de administrador
- Buscar ordenadores que confíen en el actual
- Revisar el correo en busca de passwords.
- Buscar usuarios con cuentas en otros ordenadores.
- Obtener passwords a través de snooping de la red, con caballos de Troya o aplicando un diccionario a /etc/passwd. Emplear las passwords en otros ordenadores.
- Crearse una cuenta o una entrada trasera.
- Borrar las huellas de su paso.

Claves

La manera más frontal de asaltar una cuenta es intentar adivinar la password. También es la más utilizada y la que ha conseguido historicamente más y mejores éxitos. El primer paso es intentar palabras más o menos obvias que un usuario poco cuidadoso ha escogido:

- Retorno de carro
- Sus iniciales
- Su login
- Su login invertido...
- Números obvios: fecha de nacimiento, una cifra famosa (pi, *e*), etc...

El siguiente paso es intentar atacar la clave con una batería de palabras claves hasta que una de ellas resulta ser la buena.

Estos ataques parecen primitivos pero pueden tener un éxito notable. El “Internet Worm” (Nov. 1988) lanzaba este tipo de ataque. En primer lugar probaba palabras comunes o evidentes, luego un diccionario personalizado de 432 palabras y por último un diccionario muy amplio. Contra este tipo de ataques frontales se puede:

- Aleccionar a los usuarios sobre el tipo de claves que debe utilizar.
- Repasar /etc/passwd. Comprobar que no hay usuarios extraños o sin clave.
- /etc/default/password: Fichero de configuración de passwd. Determina restricciones a las passwords. Se puede obligar a cambiar las

passwords cada cierto tiempo. Problema: los usuarios alternan claves sencillas.

- /etc/default/login: con él se configura:

a) Llevar un registro por syslog de todos los logins como root.

b) Impedir que el root acceda desde otro puesto que no sea la consola. Si el root quiere acceder por telnet tiene que hacer su.

Mensaje amenazador en el "banner". /etc/motd

- Repasar /var/adm/loginlog (aqyuí se registra cada vez que un usuario comete cinco fallos consecutivos en el login)
- Repasar /etc/passwd con **crack** para encontrar passwords obvias.

Ataques de diccionario

/etc/passwd: Fichero muy sensible. Passwords encriptadas. Debe ser legible por todos. Ataques con diccionarios ya encriptados.

Defensas: Emplear un S.O. que utilice /etc/shadow (p.ej. Solaris). Mantener /etc/passwd no accesible por extraños. Ej.: fichero falso para la zona de **ftp** anónimo.

Otros ataques a las claves

- ***Espiar a alguien mientras teclea su clave. Es otro método sencillo y muy utilizado.***
- Caballo de Troya: falso login
- Pinchar un Telnet (p.ej. **snoop**)

Otros ficheros a vigilar

- ***Se debe proteger el disco de la escritura directa de un programador habilidoso. Controlar permisos escritura en /dev/dsk***
- /etc/hosts.equiv y ~/.rhosts: Mantenerlos como ficheros de longitud cero y comprobarlo

X11

Utiliza tres formas de autentificación:

basada en host
SUN-DES-1
MIT-MAGIC-COOKIE (por defecto al arrancar openwin)

Con la opción por defecto acceso al servidor X se controla con xhost, basada en usuarios xhost + es peligroso, cualquier usuario puede acceder al servidor: abrir clientes, espiar teclado

NFS

Sistemas de ficheros exportados (shared en terminología Solaris 2.x)

Comando: share -F nfs -o rw /algun/directorio, entonces otro sistema puede acceder a ese directorio si tiene permisos de root

Sólo los directorios de usuario deben ser rw, y se pueden limitar a determinados grupos de usuarios

Comandos "r"

rlogin, rsh, rcp vienen de Berkeley UNIX. Acceso basado en ficheros /etc/host.equiv y ~/.rhosts. Peligroso, usuarios con ~/.rhosts:

+ desde cualquier máquina
+ + desde cualquier máquina, cualquier usuario

Scripts

Tener cuidado, sobre todo con el bit de *setuid*. Nunca ponerlo si no es estrictamente neesario. Vigilar especialmente los *cgis*.
Demo: Alteración de un fichero script con setuid.

Alteraciones al rutado

- Los paquetes de ICMP pueden alterar el rutado de nuestra red. "Denial of service".
- La opción de source routing puede llevar a falsificar direcciones.
- Los paquetes RIP también se pueden falsificar facilmente.
- *spoofing* Usurpar una dirección de red. Puede ser vigilado por los firewalls.

DNS

- Rica información sobre la estructura de la organización.
- Se puede confundir el cross-check de un rlogin si se controla parte de la traducción, para que sea fiable. (Hacernos pasar por otra máquina)

Demo: Correo falso con DNS falso.

SMTP

- sendmail presenta muchos fallos de seguridad. Ninguna implementación anterior a la 8.6.9 puede ser considerada segura. Ej: DEBUG (Berferd, ATT.)
- El SMTP puede ser sujeto a un bombardeo de información para provocar un colapso del servicio. Ej: Servidor ETA
- MIME. Peligro de ejecuciones indeseadas. Discutido en la especificación del MIME.

Time Protocol

Un ataque al protocolo del tiempo, una irregularidad en la hora, puede servir como aviso de un ataque a mayor escala. Esto se debe a que la etiqueta de "i-node changed" no puede ser modificada.

finger

finger y rwho proveen de información acerca de los usuarios de un ordenador. Esta información es de mucha utilidad para los asaltantes.

- Cuentas de usuarios
- Usuarios menos conectados

Soluciones:

- Prohibirlo
- Sustituirlo por una base de datos saneada.

RPC

Secure *RPC*: versión encriptada con DES pero no ha sido muy implantado.
Sólo *NFS* tiene puerto fijo. El resto de los puertos se obtiene con *portmapper*. No se debe permitir el acceso a este programa desde el exterior.

NIS. Se puede atacar:

a) Introduciendo una dirección falsa como backup de servidor de NIS

b) Recibiendo el /etc/passwd distribuído por NIS

NFS: Se basa en la entrega de handles que no expiran. Las máquinas pueden retener una autorización.

FTP anónimo

Restringir los accesos de *anonymous* a su directorio.

No permitir la escritura en los directorios, y *ftp* no debe ser dueño de los directorios tampoco.

Si es necesario permitir que se suban ficheros, hacerlo en un directorio que no pueda ser leido. así evitamos que nuestro sistema se convierta en un repositorio de información ilegal.

Instalación de shareware

Peligros:

- Agujeros, errores. Cuanto más largo y complicado es un programa más errores puede contener. Ej: sendmail.
- Trampas intencionadas. “Caballo de Troya”.

Asegurarse de que el software ha sido probado antes. Si no se está seguro, probarlo en máquinas aisladas.

Leer las news: aparecen noticias sobre bugs encontrados en programas.

Procedimientos de Administración

```
Return-Path: <piratas@maquina.saqueada.empresa.es>
Received: from maquina.saqueada by correo.empresa.es (5.x/SMI-SVR4)
      id AA04916; Wed, 29 May 1996 16:58:31 +0200
Date: Wed, 29 May 1996 16:59:37 +0200
From: piratas@maquina.saqueada.empresa.es (Juan Manuel Sanchez)
Message-Id: <9605291459.AA07302@falso.>
To: jsimplez
Subject: Arreglos
X-Sun-Charset: US-ASCII

Hola Juan:

      Necesitamos realizar un arreglo en la configuracion del ordenador que
administras, para hacerlo compatible con el nuevo entorno de ventanas
corporativo. Por favor cambia hoy mismo la password de root a:
P1ratas
      El jefe del area nos esta metiendo prisa, asi que hazlo por favor
antes de irte esta tarde.
Saludos.
```

Se debe tener una política muy estricta con las passwords, y en especial con la de root. Crear usuarios con derechos restringidos de administración (backups, logs, etc...) para que no se tengan que hacer regularmente logins como root.

No permitir la divulgación de una password de usuario.(Ej: Universidad, hijo de secretaria hackeó los Vaxes)

Seguridad de login

Elegir buenos 'passwds' (configurable en Solaris en el fichero /etc/default/passwd)
No permitir cuentas sin passwd (Solaris no lo permite por defecto)
No permitir login directamente como root (/etc/default/login)
Password cracking (Solaris mantiene un /etc/shadow que sólo puede leer root)
Cuentas guest

inetd

'Superdemonio' de internet, lanza otros demonios para servicios específicos.
Son peligrosos, sobre todo los que corren como root
Se pueden desabilitar

sendmail

Programa muy complejo y 'grande'
Probabilidad de agujeros de seguridad es directamente proporcional a la complejidad
Es mejor aislar la máquina donde se ejecuta el sendmail

FTP y HTTP

Acceso anónimo, medidas específicas
Suele ser software de dominio público
Abierto a agujeros (aún por descubrir)

9.2. Seguridad en Internet. Firewall

Lo visto en los apartados anteriores es sólo un resumen de posibles debilidades de seguridad de un sistema o una red entera. El poder controlar tantos factores es una tarea imposible, por mucho que un administrador intente mantener su red segura siempre quedan los usuarios.

Es necesario por tanto un mecanismo de seguridad que permita olvidarse de toda esta complejidad y mantener la red interna en el mismo estado que antes de conectarla a Internet. Las posibilades son varias, según se vio en el apartado de tecnologías de seguridad, pero las ventajas de una solución basada en el Firewall-1 se pueden enumerar en las siguientes.

Es un producto comercial (mantenimiento y servicio on-line)

Es el único producto que implementa NAT

Frente a una solución basada en direcciones privadas y proxies:
transparencia para los clientes
no soluciona acceso público a nuestra información

Frente a una solución basada en routers y listas de acceso:
facilidad de administración, mantenimiento, auditoría
routers no filtran por encima de TCP/UDP, problemas con RPCs (puertos variables) por tanto con NIS, NFS

routers no hacen logging, conocer qué intrusos tenemos para defenderse

9.2. Firewall

En este apartado se hace una presentación de las distintas topologías que puede adoptar una red para su conexión a Internet. El diseño de una topología concreta está muy relacionado con la seguridad de la red privada que se conecta y que hay que proteger. Por seguridad se entiende tanto la protección e integridad de la información, como los accesos de usuarios no autorizados. Esta seguridad se consigue mediante lo que se conoce genéricamente como firewall, y que admite variaciones en cuanto a diseño y componentes concretos. A la hora de elegir un modelo habrá que tener en cuenta aspectos tales como la facilidad de instalación, facilidad de mantenimiento, costes y tipo de servicios ofrecidos a los usuarios de la red. Por ello, en primer lugar se hacen una serie de definiciones básicas y a continuación se analizan las ventajas e inconvenientes de cada una de las alternativas posibles, tanto en lo que se refiere a su facilidad de uso como a su vulnerabilidad.

En la literatura técnica sobre Internet, se encuentran numerosas referencias al término *firewall*, y a menudo hay una confusión sobre su significado. Esto es así porque firewall sirve para describir cualquier mecanismo de seguridad que se use para conexión entre redes, pero siempre existen diferencias entre cada implementación concreta, y a veces con propósitos distintos. Por ello se definen primero los componentes básicos (screening router y bastion host) y a continuación la forma de combinar estos componentes en tres arquitecturas clásicas (dual homed gateway, screened host gateway y screened subnet).

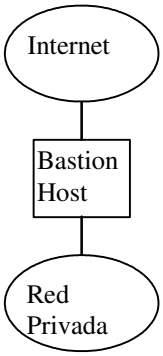
Componentes:

Un **screening router** es un componente básico de la mayoría de los firewalls. Un screening router puede ser un router comercial, o uno basado en un host con capacidad de filtrado a nivel de direcciones IP o a nivel de puertos (TCP/UDP). Algunos firewalls sólo constan de este componente, situándose entre la red privada que se pretende proteger y la Internet.

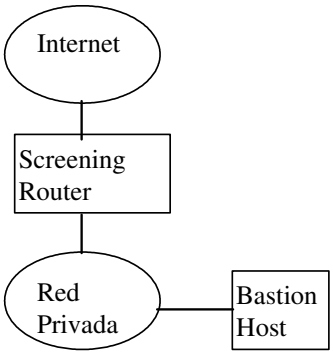
Un **bastion host** es el punto crítico en la seguridad de una red. El administrador del sistema debe prestar su atención a este punto, tanto en lo que se refiere a monitorización de los accesos como a la instalación de software modificado que incremente su seguridad.

Arquitecturas:

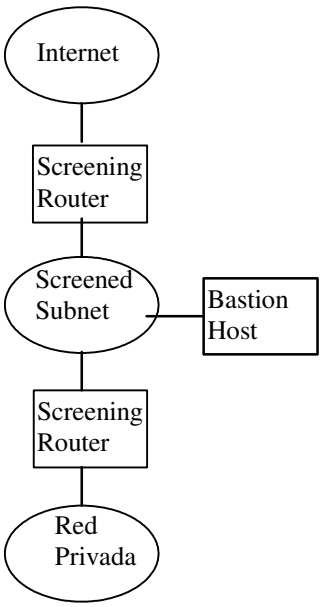
Dual Homed Gateway. Algunos firewalls se implementan sin un screening router, poniendo un sistema en la red privada y en la Internet a la vez y deshabilitando el paso de paquetes TCP/IP a través de él. Los hosts de la red privada y de la Internet pueden comunicarse con el gateway, pero el tráfico directo no está permitido. Por definición, este tipo de gateway es un bastion host.



Screened Host Gateway. Es la configuración más utilizada de gateway, en la que se usa un screening router y un bastion host. El bastion host suele estar en la red privada, y el router se configura para que los hosts externos sólo puedan alcanzar al bastión (y probablemente sólo determinados puertos).



Screened Subnet. En esta arquitectura se crea una red aislada entre Internet y la red privada, utilizando para ello dos routers que implementan distintos niveles de filtrado. Tanto los hosts de la red interna como los de Internet pueden acceder a la red aislada, pero el tráfico a través de esta red no está permitido. Esta red intermedia se conoce como screened subnet y también como DMZ (DeMilitarized Zone), y suele tener un bastion host situado en ella, donde se instala un software específico de seguridad (que se suele llamar de firewall, aunque en realidad el firewall incluye también a los routers).



La implementación dual homed gateway es la más fácil de realizar. Como no permite el paso de paquetes entre Internet y la red privada, el tráfico entre estas dos redes queda bloqueado. La facilidad de uso viene determinada por la forma en que se permita el acceso a los usuarios. Existen dos formas de permitir este acceso: mediante gateways a nivel de aplicación (proxies) y facilitando “logins” o sesiones en el bastión. La primera alternativa provee a los usuarios un número limitado de servicios (aquellos que tengan instalado su gateway en el bastion), mientras que la segunda da acceso pleno Internet. En general se prefiere el uso de proxies, puesto que al habilitar logins se abre la puerta a usuarios externos que usen passwords de usuarios fáciles de adivinar. El punto débil de esta configuración está en que el bastion está directamente expuesto a intentos de acceso desde el exterior, siendo los puntos más “probados” los sistemas de ficheros montados por NFS, ficheros *.rhosts* de usuarios, programas de backup y scripts de administración del sistema. Además, si el bastión falla, un usuario externo podría reconfigurar las rutas de forma que podría alcanzar cualquier host situado en la red interna.

Usar un screened host gateway, es una forma más segura de conectividad, manteniendo la facilidad de instalación. Como el bastion se encuentra en la red interna, los usuarios de esta red tienen conexión directa con el bastión, y por lo tanto tienen acceso hacia la red externa. La zona de riesgo se restringe al bastión, pero para llegar a él desde el exterior debe pasarse previamente por el screening router, que habrá sido configurado para el acceso a determinados puertos del bastión. En esta configuración puede usarse un mecanismo de gateways a nivel de aplicación como el descrito para la configuración basada en dual homed gateway. Este mecanismo es particularmente interesante para el caso de que en la red interna se use un esquema de direccionamiento privado como el descrito en la RFC 1597. De esta forma la red interna es invisible desde el exterior, con lo que queda invulnerable a un acceso desde Internet, pero a cambio los usuarios internos sólo podrán acceder a aquellos servicios que tengan instalado su gateway en el bastión (como en el caso anterior). En caso de fallo del bastión, para reconfigurar sus rutas

hay que tomar previamente el screening router, por lo que esta alternativa es más segura que la basada en dual homed gateway.

Una configuración basada en screened subnet se suele usar con un bastion host como el único punto de acceso dentro de la subred protegida. La zona de riesgo consiste en este bastión, y los screeened routers que conectan las redes externa e interna con la screened subnet. Esta aproximación fuerza a que todos los servicios sean provistos a través de gateways a nivel de aplicación. Si una instalación de este tipo es atacada, el intruso debe reconfigurar el rutado de tres redes, sin que sea detectado. Esto sigue siendo posible, pero se puede dificultar desabilitando el acceso a los routers, o configurándolos para permitir el acceso sólo desde determinados hosts de la red privada. En este tipo de configuración se instala un software específico de seguridad en el bastión, que filtre el acceso a los servicios de la DMZ y que registre los intentos no autorizados. Para el caso de un bastión basado en una máquina Netra de Sun, la opción más aconsejable sería el Firewall-1 que se puede adquirir opcionalmente.

9.2.1. Tecnologías

Un sistema de seguridad debe proteger los datos y recursos de red de una organización del acceso no autorizado realizado desde fuera de la organización. Esta protección debe entenderse en un sentido amplio, y no solamente restringirse a accesos a través de Internet, pues puede darse el caso de una red protegida mediante un firewall para Internet muy costoso y que sin embargo ofrezca una vía de acceso a través de una conexión dial-up. Por esta razón, la organización debe establecer primero una política de acceso que cubra la totalidad de accesos a toda su red, y después elegir la forma de implementar dicha política a través de mecanismos específicos. Los distintos mecanismos de protección son el objeto de este apartado, y van desde paquetes de software de dominio público, hasta soluciones (hardware y software) comerciales.

El acceso restringido es sólo un aspecto de la seguridad. El rendimiento es también otro aspecto a considerar, pues un firewall que reduce el tiempo de respuesta puede llevar a los usuarios de la organización a buscar otros caminos para superar el cuello de botella, que abran la posibilidad de un ataque desde el exterior. Otro punto importante es la transparencia desde el punto de vista del usuario, de forma que éstos puedan operar como si no existiera un firewall y no tengan que aprender nuevas formas de hacer su trabajo.

Las tecnologías disponibles para construir un sistema de seguridad son tres en la actualidad: filtros de paquetes, servicios proxy y traductores de direcciones. Estas tecnologías son compatibles entre sí, y un sistema determinado suele aplicar en la práctica una o más de estas tecnologías.

9.2.2. Filtros de paquetes

Hacen, como su propio nombre indica, un filtrado de paquetes basado en direcciones IP o en puertos TCP contenidos en las cabeceras de los paquetes que pasan a través de ellos. Las direcciones IP y los puertos

TCP son direcciones a nivel de red y de transporte respectivamente, pero algunos traductores de direcciones pueden operar también a nivel de aplicación, cubriendo servicios como RPC, NIS y NFS. Los filtros de paquetes se implementan como parte del núcleo del sistema operativo del sistema bastión, o en los routers de acceso a Internet (por ejemplo los routers CISCO permiten configurar listas de acceso basadas en un esquema de direcciones IP más puertos).

La diferencia entre estas dos soluciones estriba en que la primera realiza un registro de los accesos realizados, pudiéndose establecer alarmas para determinados eventos (por ejemplo intentos de acceso a recursos no permitidos o ataque desde una máquina externa que pretende ser un usuario autentificado), que no permite la solución basada en routers. Además los filtros implementados en el núcleo permiten un control de acceso sobre servicios más completo que los implementados en los routers.

La ventaja de los filtros de paquetes es que pueden añadirse a cualquier red corporativa a un coste bajo, ofreciendo un buen rendimiento si se definen sólo unos pocos, y siendo absolutamente transparentes para las aplicaciones y los usuarios finales. El inconveniente es que por ellos mismos ofrecen una pobre seguridad. El problema está en que si se gana el acceso a la máquina donde está el filtro el intruso podría cambiar los filtros para poder acceder a cualquier punto de la red interna. Otro problema es que los filtros de paquetes no pueden monitorizar información de estado del enlace, por tanto no pueden registrar protocolos no orientados a la conexión (tipo datagrama, basados en UDP) como NFS (Network File System) y RPCs (Remote Procedure Calls).

9.2.3. Traductores de direcciones

Es la última tecnología que ha aparecido en el mercado, y la razón de su desarrollo es el limitado espacio de direcciones IP. Los traductores de direcciones (conocidos como NAT boxes en la literatura, Network Address Translators) sirven como intermediarios entre la red interna y la externa, y permiten el uso de un esquema de direcciones privadas en la red interna mientras que hacia la externa se presentan un conjunto de direcciones IP válidas. Como estas NAT boxes sirven de intermediarias entre las redes interna y externa, se pueden adaptar para usar también servicios proxy.

Las soluciones comerciales disponibles son aún escasas, y vienen en una combinación de hardware y software específicos. Las ventajas e inconvenientes de los traductores de direcciones son análogos a los descritos para los servicios proxy, pero teniendo en cuenta que suponen un coste adicional, y que los servicios proxy también permiten tener un esquema de direccionamiento IP privado en la red interna (puede consultarse la RFC 1597). La diferencia entra ambas tecnologías es que en el caso de los proxies el acceso servicios Internet estará limitado a aquellos servicios que tengan instalado su proxy en el firewall, pero que generalmente será suficiente para el tipo de servicio que se quiere ofrecer a los usuarios.

9.2.4. Firewall-1

Este firewall está construido en el núcleo del sistema operativo, y se basa en la tecnología de filtros de paquetes y de traducción de direcciones. El filtro de paquetes permite hacer un registro de los paquetes que pasan por él, y también establecer alarmas (ejecución de scripts, envío de e-mails al administrador) para eventos sospechosos. Además soporta servicios basados en UDP (como DNS, Domain Name Service, o Archie), basados en RPC (NIS, Network Información Service, o NFS, Network File System), basados en FTP y en HTTP. El software se puede instalar en el gateway de acceso a la red, en servidores o en hosts individuales, y puede ser administrado desde un puesto central a través de comandos, o preferentemente mediante un interfaz OpenLook muy claro.

Para la solución propuesta de Screened Subnet, sólo sería necesario instalar el Solstice Firewall-1 en el bastión que protege la red DMZ, con lo que la administración se limitaría a ese punto. La seguridad ahora dependería del router y además del bastión, y se tendría una zona donde poner la información que se quisiera hacer pública.

Componentes del Firewall-1

Inspection Modules: se instala en el núcleo del sistema operativo, entre IP y la red

Firewall (para un gateway o un host)
Router (para un router)
Autentificación

Management Module: controla a los inspection modules, su front end es el interfaz gráfico

Adicionalmente existen versiones encriptadas o no encriptadas

Lo que se puede comprar incluye una combinación de algunos de estos componentes, en concreto se tienen dos paquetes:

Medium Security Center:

1 Firewall Inspection Module
1 Management Module
Se deben instalar en la misma máquina

Router Module:

1 Router Inspection Module

Estos dos paquetes tienen la opción de encriptación

Instalación del Firewall-1

Para instalar el software Firewall-1 puede usarse tanto la herramienta *pkgadd* desde la línea de comandos (tal y como aparece a partir de la página 14 del manual) o con la utilidad gráfica *swmtool*. En cualquiera de los dos casos hay que contestar una serie de preguntas, y los valores que se le han dado son los siguientes:

Installation Directory: /opt/SUNWfw
Installation Options: (3) Solstice Firewall-1 Medium Security Center
Access Permissions: No se añadió grupo, sólo root puede ejecutarlo
Authentication: No se habilitó
Inspection Module hosts: No se añadió ninguno
Host external interface: le0 (para chequear nº de licencias)

Al finalizar la instalación se recomienda modificar los ficheros iniciales de login (.profile para sh y ksh, .cshrc y .login para csh) para incluyan las siguientes variables de entorno:

FWDIR es /opt/SUNWfw
PATH es $FWDIR/bin
MANPATH es $FWDIR/man

Una vez instalado el software hay que solicitar a Sun las licencias adecuadas para los distintos componentes. En la caja de cada módulo viene una tarjeta con el nombre del módulo y con su número de serie, que junto con el "hostid" de la máquina hay que remitir a Sun (por correo electrónico o por fax). El centro de distribución de licencias de Sun responderá con las instrucciones para añadir las licencias, que lógicamente sólo funcionará para máquina de la que se ha enviado el hostid. Este proceso ya está completo, pero a continuación se da el método de instalación de las licencias por si hubiera que introducirlas de nuevo.

Para FW-Medium Security Center:
fw putlic 0x807d126a 7fffadf7-18bd5f39-196f08c9 stdmedium

Para FW-Router Module
fw putlic 0x807d126a 7fffdac1-02d13f41-bcdea8c1 router1

Para FW-VPN Encryption
fw putlic 0x807d126a 7fff13fe-34615bff-0a69c001 encryption

Manejo del Firewall-1

En la instalación se ha dispuesto un firewall sobre el ordenador *ninot* que actúa como gateway entre una red externa, una red interna, una zona desmilitarizada y una conexión hacia las demás consejerías. *ninot* corre el software "Soltice Firewall-1", versión 2.0.

Ubicación: El software se encuentra instalado en el directorio
/opt/SUNWfw

En particular los ficheros ejecutables se encuentran en:
/opt/SUNWfw/bin

Para arrancar el firewall el usuario *root* debe ejecutar:
/opt/SUNWfw/bin/fwstart

Este comando se ejecuta durante el arranque desde el script
/etc/rc3.d/S95firewall1

El firewall se detiene con el comando:
/opt/SUNWfw/bin/fwstop

El firewall dispone de un interfaz gráfico muy completo y bastante intuitivo. Para lanzar este interfaz debemos tener corriendo el interfaz de ventanas. Entonces podemos ejecutar desde un shell:
/opt/SUNWfw/bin/fwui &

Políticas de seguridad e implementación de Firewall-1

Para usar de forma adecuada el producto Firewall-1, el orden en que se deberían hacer las cosas es el siguiente:

En primer lugar definir una política de seguridad: es decir, escribir qué es lo que está permitido hacer, y qué es lo que está prohibido. Básicamente existen dos aproximaciones: lo que no está prohibido está permitido, o lo que no está permitido está prohibido.

Después, definir objetos que intervienen esa política: máquinas y usuarios

Por último, definir reglas que implementan la política de seguridad descrita en primer lugar, utilizando para ello los objetos definidos. Las reglas se aplican en el orden en que están definidas, y se ejecuta la primera que coincide.

Definición de objetos Firewall-1

El firewall forma una base de datos con distintos elementos de la red:

- hosts
- gateways
- networks
- domains
- routers
- grupos de los anteriores

Los grupos sirven para propósitos de administración, y permiten dar permisos comunes a distintos elementos.

Los elementos se definen con el interfaz gráfico del firewall. Se comienza por lanzar el editor de los mismos: desde la ventana *Rule Base Editor* se marca en la casilla *Windows: network objects*. Aparece entonces la ventana de control de objetos: *Network Objects Manager*.

Para definir un nuevo objeto introducimos su nombre en el campo *Edit/Create*. En el menú desplegable desde el botón seleccionamos el tipo de objeto (router, host, etc). Pasamos entonces a la ventana de Propiedades del Objeto. Todos los objetos menos los grupos presentan los siguientes campos:

IP Address**:** Debe ser rellenada. Si se pulsa el botón *Get* el programa intentará rellenar el campo automáticamente con la información disponible en las bases de datos del host (/etc/hosts, NIS,...)

- Location**:** Se define el objeto como interno o externo al firewall. Este considera que su misión es proteger los objetos internos.

Definir un host o un gateway

- Host/Gateway: Seleccionamos gateway si el ordenador actúa como pasarela entre dos redes.
- Firewall-1 Installed/Not Installed

Se incluyen luego campos que identifican la autentificación y la encriptación entre el host remoto y el local, así como información. Por último existe un campo para añadir interfaces a la descripción del host.

Para añadir interfaces a la red se pulsa el botón *Add* y se rellenan los campos *Name, Net Address y Net Mask.*

Definir una red:

- Net Mask: Máscara de la red
- Gateway de la red.

Definir un dominio DNS

En este caso en vez del campo de dirección IP se debe rellenar el campo *Name*, con el nombre completo del dominio.

Definir un router

- Type**:** El firewall puede manejar dos tipos de routers: los Cisco y los Wellfleet.

Siguen los campos referidos a SNMP, comentarios, e información. También se pueden definir múltiples interfaces tal y como se hizo con los hosts.

Definir un grupo:

Para formar un grupo aparece una ventana con los elementos que lo componen, junto a la de control de objetos. Para añadir un elemento al grupo se selecciona y se pulsa el botón *Add Objects*. Esto nos permite transferir un grupo de manera que quede anidado en el nuevo. Si se selecciona un grupo, y se pulsa el botón *Add Group Elements* se copian al nuevo grupo todos los elementos del anterior. Esta copia es plana, es decir, no hay anidamiento de grupos.

Cuando se ha terminado de transferir objetos se pulsa *Apply* y finaliza la edición del grupo.

Definición de reglas Firewall-1

El editor de reglas es la ventana principal que aparece en el interfaz de usuario del firewall. El editor contiene un cuadro con el conjunto de las reglas en aplicación. estas aparecen en orden de prioridad, de arriba a abajo. Además de las que se pueden ver aquí existe un conjunto de reglas que también se aplican: las propiedades de control. Estas propiedades

controlan los paquetes de control de la red: DNS, ICMP, etc... Esto permite regularlos aparte teniendo en la ventana sólo las reglas principales, referidas a servicios.

Las reglas de la base de reglas se aplican en su orden de prioridad. Cuando el firewall recibe un paquete, lo compara con la primera regla. Si el paquete entra dentro del ámbito de esta regla, se toma la acción que esta especifica. Si no, se pasa a la siguiente regla. Si ninguna regla casa con el paquete este se descarta. No debemos olvidar que también se le aplican las propiedades de control (un error común al construir la base de datos).

La base de reglas

Los campos de la base de reglas son los siguientes:

Number**:** El número de orden o prioridad de la regla.

- Source**:** La fuente de los paquetes. Cuando esta regla se refiere a un servicio de tipo cliente-servidor, se refiere al cliente, iniciador de la petición o de la conexión.
- Destination**:** El ordenador receptor del paquete. Si la regla describe un servicio de tipo cliente-servidor, se refiere al servidor, el receptor de la petición o de la conexión.
- Services**:** Tipo de paquetes a los que se refiere la regla. Hay un número grande de servicios ya definidos, pero se pueden definir nuevos servicios.
- Action**:** La acción a realizar con este paquete. Puede ser:

Accept: Aceptar el paquete o conexión.

Drop: Descartar el paquete o conexión.

Reject: Rechazar el paquete notificando su rechazo al emisor.

Authenticate: Autentificar el servicio. Algunos servicios (ftp, telnet...) pueden ser autentificados por el firewall, es decir, este interroga al usuario para comprobar su identidad, antes de permitirle proseguir la sesión con una máquina protegida.

Encrypt: Encripta la comunicación. Es necesario que exista un firewall protegiendo también el ordenador remoto, pues la encriptación se realiza entre los dos.

Track**:** Este campo describe si se va a registrar o no el paso de este paquete. El valor de este campo puede ser:

- *:* No se registra este evento.

short: Se anota un registro corto en el fichero “log” del firewall.

long: Se anota un registro largo en el log.

alert: Se dispara una señal de alarma en el monitor.

mail: Se envía un correo al administrador.

snmptrap

user defined

Install on: Las reglas pueden instalarse en todos los ordenadores de la red que tengan instalado el software del firewall o puedan ser controlados por éste. Este campo describe en que elementos de la red se instalarán las reglas:

Src (source): La regla se instalará en las máquinas que recoge el campo *source* de esta regla.

Dst (Destination): La regla se instalará en las máquinas que recoge el campo *destination* de esta regla.

Gateways: Fuerza la regla en las pasarelas. Esto incluye el firewall.

Routers: Instala la regla en los routers de la red.

Comments**:** En este campo se pueden anotar los comentarios o la descripción de la regla.

Edición de la base de reglas

Para crear una regla nueva se despliega el menú *Rule* y se selecciona la entrada *New rule...* Dentro de esta escogemos la prioridad de la regla dentro de la base. Una vez insertada la nueva regla se editan los campos de ésta. Para editarlos se despliega desde cada uno de ellos un menú contextual.

Propiedades de control Firewall-1

Las propiedades de control configuran el comportamiento del firewall e incluyen reglas acerca de los protocolos de gestión. Las propiedades de control abarcan cinco categorías:

- Security Police: Aquí se controlan reglas acerca de diversos protocolos de red: paquetes UDP, RIP, RPC, ICMP... Para cada regla definida se debe especificar el lugar de la base de datos en que se debe aplicar: al principio, al final o antes de la última regla.
- Logging and alerting: Se definen aspectos relacionados con las alertas.
- Names Resolving: Se define el orden en que se deben consultar las distintas bases de datos (NIS, DNS, etc...) para resolver los nombres de los programas.
- Routers: Al igual que en *Security Police* se definen reglas acerca de protocolos de red, pero unicamente aquellas relevantes para los routers.
- Autentificación: Se controlan parámetros y especificaciones relacionados con la autentificación.

NAT: Network Address Translator de Firewall-1

El firewall “Solstice Firewall-1 Version 2.0” incluye un servicio de NAT o Network Address Translator. Este servicio, una vez configurado puede realizar traducciones de direcciones de red: cambia la dirección de origen de paquetes salientes y la de destino de los paquetes entrantes que llegan como respuesta.

El servicio NAT tiene dos utilidades principales:

- Ocultar las direcciones de la red interna para que no sean conocidas desde el exterior.
- Traducir direcciones privadas por direcciones públicas. En gran número de redes privadas se emplean direcciones no-públicas según el modelo descrito en la RFC1597.

Modos de traducción del Firewall-1

El firewall de Solstice puede traducir direcciones según cuatro esquemas:

- FWXT_HIDE: Traduce un conjunto de direcciones IP internas por una dirección externa. La traducción por tanto es N->1. La discriminación entre las direcciones de destino de los paquetes entrantes se realiza en función del puerto. Sólo es aplicable en servicios iniciados desde la red interna.
- FWXT_SRC_STATIC: Traduce la dirección de origen de un paquete saliente por otra dirección. La correspondencia entre direcciones es de uno a uno.
- FWXT_DST_STATIC: Traduce la dirección de destino de un paquete entrante por otra. La correspondencia es también de uno a uno. Se debe emplear asociada con la regla anterior para traducir univocamente una dirección interna por otra externa.
- DPORT_STATIC: Cambia el puerto de destino de un paquete originado en la red interna.

Configuración del NAT de Firewall-1

El servicio NAT se configura desde un shell de Unix ejecutando el programa:

.../bin/fwxlconf.

Este programa consulta al usuario con un sistema de menús formando una tabla de traducción. Cada entrada de la tabla de traducción tiene los siguientes campos:

Número
- Primera dirección original
- Ultima dirección original
- Método de traducción
- Primera dirección traducida (y única, en el método FWXT_HIDE).

Para que la traducción se realice, las direcciones implicadas deben pertenecer a redes definidas como objetos en el firewall. Una de las redes será interna (la de las direcciones originales) y la otra (la de las direcciones traducidas) externa.

Una vez configurada la tabla con fwxlconf se debe parar y arrancar el firewall para que tengan efecto los cambios en la configuración (ejecutar fwstop y fwstart)

Control de listas de acceso de routers CISCO con Firewall-1

Desde el interfaz de usuario del Firewall-1 es posible configurar las listas de acceso de un router CISCO (o Wellfleet). En este apartado se dan las instrucciones para hacerlo. Las listas de acceso se transmiten al router mediante un TELNET, y no mediante SNMP como pudiera dar a entender la documentación del producto. Para el caso de un router CISCO, los pasos son los siguientes:

Se define el router a configurar como un objeto, en el que se deben especificar la versión del sistema operativo del router, su dirección IP, y los passwords de acceso y para ejecutar el comando enable.

Se define la/las reglas que se quieren instalar en el router

En el apartado 'Install On' de la regla se le especifica el objeto router definido.

Se chequea la regla para ver si es correcta.

Se ejecuta la opción de menú Routers, se selecciona Install, y en la ventana que aparece se pulsa sobre Apply.

NOTA IMPORTANTE: La configuración del router desde el Firewall-1 se hace invocando unos scripts de un shell llamado *expect* que sirven para intercambiar comandos con el router. En algunas versiones del Firewall-1 hemos detectado que existe un error en el script /opt/SUNWfw/cisco/fwciscoput, en concreto el router nunca responde 'Edit with DELETE', por lo que la configuración nunca se llega a completar. La solución consiste en comentar esas líneas.

Otro punto que es interesante es el uso de SNMP para recabar y cambiar información del router (comandos *SNMP Fetch* y *SNMP Set* en la ventana de configuración del router), sobre todo en lo que se refiere a sus interfaces. Para poder usar el agente SNMP de un router CISCO se tiene que definir una lista de acceso y asociarla al agente SNMP (el servidor), esto se puede hacer con los comandos:

access-list <N> permit <DIRECCION>
snmp-server community <PASSWD> RW <N>

siendo ...
<N> el nº de la lista de acceso creada
<DIRECCION> la dirección IP donde está el Router Inspection Module
<PASSWD> el que se configure en el objeto router

Análisis de logs de **Firewall-1**

La ventana de los logs generados por el firewall, muestra información sobre número del log, fecha, hora, origen, destino, interface, tipo, prototipo, información. regla, acción, servicio, usuario, etc.

Se puede filtrar para ver logs que cumplan una característica determinada, generalmente lo más interesante serán aquellos que se refieran a acciones prohibidas. Si se detecta una serie de puertos altos consecutivos explorados probablemente estamos ante un ataque de un SATAN que está intentando ver nuestros fallos de seguridad (naturalmente no hay ningún peligro con el Firewall-1 funcionando).

9.3. PROXY

Un servidor proxy es una pasarela a nivel de aplicación, que se comporta como servidor y como cliente a la vez. Esta pasarela actúa como un servidor respecto al cliente final, y como un cliente del servidor final, existiendo una pasarela específica (proxy) para cada servicio usado (HTTP, FTP, Wais, Gopher, etc).

Estos servicios no permiten el tráfico directo entre la red interna y la red externa, sino que el cliente debe establecer primero un circuito con el proxy, y éste después abrirá otro circuito distinto con el servidor. Esto es útil para una organización que tiene un esquema de direcciones IP privado para sus máquinas, pues les permite acceder a determinados servicios de Internet (HTTP, FTP, Wais, Correo electrónico) a través de este servidor proxy, manteniendo la privacidad de su red interna (las máquinas no pueden ser accedidas porque no tienen direcciones IP válidas).

Esta solución es más potente que los filtros de paquetes para seguridad, pero inferior respecto los filtros en lo que respecta a transparencia y rendimiento. La ventaja fundamental sobre la tecnología descrita en el apartado anterior es que en el caso de que el proxy falle, no hay una ruta directa hacia la red que está detrás del firewall. En el caso de un filtro de paquetes, si por alguna razón se elimina un filtro (accidental o intencionada), el filtro seguirá rutando los paquetes. El inconveniente es que cada aplicación que se use en el firewall necesita su propio proxy. La mayoría de los proxies permite la utilización de clientes y servidores sin modificar, pero en cualquier caso los usuarios finales deben aprender utilizar procedimientos especiales para negociar con el proxy.

Algunos servicios proxy requieren la instalación de un software específico en las máquinas clientes. El ejemplo más conocido es el toolkit SOCKS, que permite crear servicios proxy genéricos, pero que necesita que las aplicaciones de todos los clientes sean compatibles con SOCKS.

Hay servidores proxy que implementan una caché para reducir el tráfico a través de las redes (su funcionamineot es el mismo que en el caso de la caché de servidores Web).

9.3.1. Servidores Proxy disponibles

En el campo del dominio público, el único servidor proxy existente es el desarrollado por el CERN europeo. Está incluido con el servidor HTTP.

Entre los servidores comerciales, no existe ningún servidor Proxy para Windows NT. Netscape Corporation sí dispone de un servidor comercial.

Servidores de dominio público

La distribución del servidor del CERN es de dominio público, y puede obtenerse en la URL http://www.w3.org.

Existen versiones precompiladas para una variedad de plataformas y sistemas operativos. En este caso, lo único que se necesita es descargar el fichero adecuado, descomprimirlo (uncompress) y expandirlo (tar).

Si no se dispone de una versión precompilada para nuestra plataforma, es necesario descargar los ficheros fuenet y compilarlos, ejecutando previamente unos scripts que personalizan los makefiles.

Servidor de Netscape

El servidor proxy de Netscape se configura en base a formularios html. Para utilizarlos es preciso disponer de un navegador, por ejemplo Netscape Navigator. Este mecanismo permite además la configuración remota de los servidores.

El servidor de administración es un demonio httpd independiente que atiende peticiones por el puerto tcp 11111 (en la configuración actual) y es preciso arrancarlo de modo independiente. Este servidor incorpora además un demonio de SOCKS.

Configuración de proxy de Netscape

La configuración de los servidores de Netscape está registrada en varios ficheros:

```
#Fichero magnus.conf
Port 8080
LoadObjects obj.conf
RootObject default
ErrorLog /opt/ns-proxy/logs/errors
PidLog /opt/ns-proxy/logs/pid
User nobody
ServerName sol.midominio.es
MaxProcs 50
Init fn=load-types mime-types=mime.types
Init fn=init-clf global=/opt/ns-proxy/logs/access
Init fn=init-proxy log-format="common" timeout="120"
Init fn=init-cache cache-root="/var/opt/ns-proxy/cache" cache-size="200" lock-
timeout="1200" top-dirs="4" cache-protocols="http,ftp" cache-mode="all" max-
uncheck="0" lm-factor="0.06" gc-times="3:00" gc-nice="4" connect-mode="normal"
```

El fichero magnus.conf se encarga de guardar la configuración de control del servidor, en aspectos que no sean de manejo de documentos o directorios (de ello se encarga el fichero obj.conf). Todas las líneas de este fichero tienen el formato:
Directiva Valor

Cada directiva especifica un aspecto del funcionamiento del servidor, y el formato del campo valor depende de la directiva de que se trate. Las directivas son las mismas que para el caso de un servidor HTTP, y se relacionan a continuación.

La lista de las posibles directivas y su significado:
- **ServerName**. Nombre del host.
-**Port**. Número del puerto TCP en el que el servidor escucha peticiones.
-**User**. Nombre de la cuenta UNIX propia del servidor.
-**MaxProcs**. Número máximo de procesos activos.
-**MinProcs**. Número mínimo de procesos activos.
-**ProcessLife**. Número de peticiones que que puede servir un proceso hijo antes de morir.
-**ErrorLog**. Directorio en el que el servidor guarda los registros de los errores.
-**PidLog**. Nombre del fichero en el que se guarda el identificador del proceso servidor principal.
-**LoadObjects**. Especifica el fichero de configuración de objetos.
-**RootObject**. Define el objeto por defecto del servidor.

-**Chroot**. Permite restringir los ficheros accedidos a los de un directorio, por razones de seguridad.
-**Init**. Directiva especial. Sirve para inicializar los subsistemas del servidor (por ejemplo, logs de los accesos).
-**DNS**. Servidor de DNS.
-**Security**. Sólo para el Commerce Server. Especifica el tipo de seguridad.

```
<Object name=default>
NameTrans fn=map from=file: to=ftp:
NameTrans fn=pfx2dir from=/admin/bin dir="/opt/ns-proxy/admin/bin" name=cgi
NameTrans fn=pfx2dir from=/admin dir="/opt/ns-proxy/admin/html" name=file
NameTrans fn=pfx2dir from=/ns-icons dir="/opt/ns-proxy/ns-icons" name=file
Service fn=deny-service
AddLog fn=proxy-log iponly=1
</Object>

<Object name=cgi>
PathCheck fn=unix-uri-clean
PathCheck fn=find-pathinfo
ObjectType fn=force-type type=magnus-internal/cgi
Service fn=send-cgi
</Object>

<Object name=file>
PathCheck fn=unix-uri-clean
PathCheck fn=find-index index-names=index.html
ObjectType fn=type-by-extension
ObjectType fn=force-type type=text/plain
Service fn=send-file
</Object>

<Object ppath="/opt/ns-proxy/admin/*">
AuthTrans fn=basic-ncsa auth-type=basic userfile="/opt/ns-proxy/admin/config/admpw"
<Client dns="*~sol.midominio.es" ip="*~127.0.0.1">
PathCheck fn=deny-service
</Client>
PathCheck fn=require-auth realm="Netscape Proxy Administration" auth-type=basic
</Object>
<Object ppath="http://*">
Service fn=proxy-retrieve
</Object>
<Object ppath="ftp://*">
Service fn=proxy-retrieve
</Object>
<Object ppath="gopher://*">
Service fn=proxy-retrieve
</Object>
<Object ppath="https://*">
Service fn=proxy-retrieve
</Object>
<Object ppath="connect://*:443">
Service fn="connect" method="CONNECT"
</Object>
```

El fichero obj.conf indica al servidor cómo deben manejarse los documentos, ejecutables, etc. En el caso de un servidor proxy este fichero sólo es necesario si se utilizan formularios para administrar el sistema. Si se utiliza, es obligatorio que el fichero contenga las descripciones de cuatro objetos:

- default

- cgi
- file
- formularios de administración

Las descripciones de otros objetos pueden ser incluidas por el administrador.

```
#--Netscape Communications Corporation MIME Information
# Do not delete the above line. It is used to identify the file type.
#
# This is a pruned MIME types file for Netsite Proxy since it has most
# of the MIME types already compiled in.  Types that are part of the
# admin interface (basically HTML and GIF) still have to appear here,
# or they will not be known to the part of the server that manages the
# admin interface calls.
#
# Icons (internal-gopher-...) are references to Netscape's compiled-in
# internal icons.  If the client doesn't support those icons, the
# proxy will provide them for it.

type=application/oda          exts=oda
type=application/pdf          exts=pdf
type=application/x-mif        exts=mif
type=application/x-dvi        exts=dvi
type=application/x-hdf        exts=hdf
type=application/x-netcdf     exts=nc,cdf
type=application/x-texinfo    exts=texinfo,texi   icon=internal-gopher-text
type=application/zip          exts=zip
type=application/x-tar        exts=tar
type=application/x-macbinary  exts=bin
type=application/x-stuffit    exts=sit

type=image/gif              exts=gif          icon=internal-gopher-image
type=image/jpeg             exts=jpeg,jpg,jpe   icon=internal-gopher-image
type=image/x-xwindowdump    exts=xwd          icon=internal-gopher-image

type=text/html              exts=htm,html,shtml icon=internal-gopher-text
type=text/plain             exts=txt          icon=internal-gopher-text
type=text/richtext          exts=rtx          icon=internal-gopher-text
type=text/tab-separated-values exts=tsv          icon=internal-gopher-text
type=text/x-setext          exts=etx          icon=internal-gopher-text

type=application/x-tar enc=x-gzip exts=tgz

enc=x-gzip                exts=gz
enc=x-compress            exts=z
```

El fichero mime.types le indica al servidor cómo convertir ficheros discriminados por su extensión a ficheros MIME. En general, las líneas de este fichero tienen tres campos:
- **type/subtype**. Identifica el tipo de objeto MIME.
- **exts**. Identifica la extensión de los ficheros del tipo o subtipo.
- **icon**. Indica el icono que el navegador presenta para el tipo de fichero.

9.3.2. SOCKS

El demonio de SOCKS es un demonio genérico que permite conexiones punto a punto a través de un firewall. El servidor proxy de Netscape soporta la versión 4 de SOCKS.

SOCKS no diferencia entre ordenadores internos a una red y externos a la misma. Esto quiere decir, que el demonio de SOCKS permite tanto el acceso de los usuarios de la red interna al exterior, como de usuarios exteriores al interior.

Por seguridad, es conveniente que los usuarios externos no tengan acceso a la red interna. El demonio de SOCKS debe configurarse para impedir toda entrada desde el exterior.

El servidor de proxy utiliza el fichero sockd.conf para controlar el acceso al servidor proxy de SOCKS.

9.4. Software de Seguridad

TAMU: Colección de herramientas para contruir firewalls y detectar ataques.
ftp://net.tamu.edu/pub/security/TAMU

COPS: Herramientas de auditoría
ftp://ftp.cert.org/pub/tools/cops

Tripwire: Paquete de evaluación de sistemas. comprueba ficheros alterados.
ftp://ftp.cs.purdue.edu/pub/spaf/COAST/Tripwire

ISS: Comprueba redes completas
ftp://ftp.uu.net/usenet/comp.sources.misc/volume39/iss

SATAN: Comprueba redes completas, incluyendo redes remotas.
ftp://ftp.win.tue.nl/pub/security/satan.tar.Z

Crack: Comprueba la vulnerabilidad de las passwords de un sistema
ftp://ftp.cert.org/pub/tools/crack

CERT (Computer Emergency Response Team)

Fundado por DARPA a raiz del Internet Worm.
Mantiene parches y avisos sobre problemas de seguridad.
En caso de necesidad se les puede pedir ayuda en cert@cert.org. Se les debe enviar una descripción completa del problema, así como la asistencia que necesitamos.
También mantiene herramientas diversas de seguridad accesibles por ftp:
ftp://ftp.cerg.org/pub/tools

Listas de correo:

firewalls: Enviar mensaje a majordomo@greatcircle.com con el siguiente cuerpo:
subscribe firewalls
o
subscribe firewalls_digest (para una versión resumida)

bugtraq: Lista sobre bugs en programas. Suscribirse enviando un mensaje a:
bugtraq-request@fc.net

Newsgroups:

Existen varios foros sobre seguridad:
comp.security.announce
comp.security.misc
comp.security.unix
alt.security
sci.crypt

9.5. Bibliografía

Cheswick, William R.; Bellovin, Steven M. *Firewalls and Internet Security*. Addison-Wesley 1994
C.Pfleeger; *Security in computing.* Prentice Hall.

9.6. Prácticas

- ***SATAN***
- ***crack***
- ***Firewall-1***

ANEXO 1. CONFIGURACION DE UN ROUTER

Rutado

El rutado en el interior de la red asi como en las máquinas que se han instalado para la conexión a Intenet es mediante rutas estáticas.

Rutas en el Firewall

Routing Table:

Destination	Gateway
127.0.0.1	127.0.0.1
195.76.12.132	192.168.5.204
195.76.12.128	195.76.12.130
195.76.12.192	195.76.12.194
192.168.5.0	192.168.5.200
224.0.0.0	195.76.12.130
default	195.76.12.129

Configuración del router

El router se ha configurado de la siguiente manera:

- Su rutado es estático.
- La red 192.168.5.x privada de MIDOMINIO se alcanza siempre a través del firewall.
- La ruta por defecto se alcanza a través del router de Internet conectado mediante una linea Frame Relay a Ibernet.

```
Current configuration:
!
version 11.1
service udp-small-servers
service tcp-small-servers
hostname MIDOMINIO
interface Ethernet0
 ip address 195.76.12.129 255.255.255.128
interface Serial0
 no ip address
 encapsulation frame-relay IETF
 no fair-queue
 frame-relay lmi-type q933a
interface Serial0.16 multipoint
 ip address 194.179.6.49 255.255.255.0
 frame-relay interface-dlci 16
 frame-relay map ip 194.179.6.129 16 broadcast
interface Serial1
 no ip address
 shutdown
!
ip default-network 194.179.2.0
ip route 194.179.2.0 255.255.255.0 194.179.6.129
ip route 195.76.12.131 255.255.255.255 195.76.12.130
ip route 195.76.12.189 255.255.255.255 195.76.12.130
ip route 195.76.12.190 255.255.255.255 195.76.12.130
ip route 195.76.12.192 255.255.255.192 195.76.12.130
!
end
```

ANEXO 2. LINEA FRAME-RELAY

Averias en la linea Internet

En caso de mal funcionamiento de la linea Frame Relay que conecta MIDOMINIO a Internet se dara un parte de averia a Telefónica (teléfono 900112002) comunicando los siguientes datos de la instalación:

- Número Administrativo del router de MIDOMINIO: 2810000 1065422
- NRI del router en la Red-1 : 31613991

ANEXO 3. CONFIGURACION DE LINUX COMO SERVIDOR INTERNET O INTRANET.

A.3.8. Configuración de sendmail.

Los ficheros relacionados con el correo son:

Arranque:	/etc/rc.d/rc.inet2
Ejecutable:	/usr/lib/sendmail
Configuración:	/etc/sendmail.cf
Help file:	/etc/sendmail.hf
Status file:	/etc/sendmail.st
Log file:	/var/log/sendmail.log
Ident. proceso:	/etc/sendmail.pid

Los directorios son:

Cola de salida:	/var/spool/mqueue
Buzón usuarios (entrada):	/var/spool/mail

El fichero más importante es sendmail.cf, que determina la elección de los agentes de entrega, las reglas de reescritura de direcciones y los formatos de las cabecera.

Las opciones de arranque del proceso sendmail son:

/usr/lib/sendmail -bd	corre en modo background, aceptando conexiones SMTP
/usr/lib/sendmail -q1h	procesa mensajes salvados en la cola cada hora
/usr/lib/sendmail -bt	testea las reglas de reescritura introducidas manualmente
/usr/lib/sendmail -bp	imprime la cola de correo (como mailx)

Configuración básica

Al arrancar, la máquina lanza automáticamente el demonio del correo (/usr/lib/sendmail), desde el script de inicio /etc/rc.d/rc.inet2, que lo lanza en modo background (para que atienda peticiones externas del protocolo SMTP) y para que consulte la cola de correo cada 15 minutos. El correo electrónico se configura modificando el fichero /etc/sendmail.cf.