SEGURIDAD EN INTRANET PARA LOS SERVICIOS WEB, CON EL SISTEMA OPERACIONAL LINUX: HTTP, DNS, POP3, SMTP, TELNET

JULIÁN DAVID PORTELA MOLINA
CARLOS EDUARDO SANTOS PINZÓN
ANGIE MILENA VIGOYA GONZÁLEZ

CORPORACIÓN UNIVERSITARIA UNITEC
FACULTAD DE INGENIERÍA DE SISTEMAS
BOGOTÁ, D.C.
27 de Julio de 2007

SEGURIDAD EN INTRANET PARA LOS SERVICIOS WEB, CON EL SISTEMA OPERACIONAL LINUX: HTTP, DNS, POP3, SMTP, TELNET

CARLOS EDUARDO SANTOS PINZÓN
JULIÁN DAVID PÓRTELA MOLINA
ANGIE MILENA VIGOYA GONZÁLEZ

Ciclo preparatorio para grado presentado al programa Ingeniería de Sistemas como requisito parcial para optar al titulo de TECNÓLOGO EN SISTEMAS

DUILIO ARNULDO BUELVAS
ASESOR

CORPORACIÓN UNIVERSITARIA UNITEC
FACULTAD DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
26 de 2007

CONTENIDO

INTRODUCCIÓN

Linux es un sistema operativo multiusuario real donde cada usuario tiene su propia Terminal. Linux dispone de todas las características de los sistemas Unix como control de acceso a los usuarios verificando una pareja de usuario y clave, ficheros, directores que tienen permisos. Hablaremos de los servicios Web en Intranet como (Correo DNS y Servidor Web) con el fin de crear una estación de trabajo Linux relativamente segura. Primero se intentará conseguir acceso como usuario "normal" para posteriormente ir incrementando sus niveles de privilegio utilizando las posibles debilidades del sistema: programas con errores, configuraciones deficientes de los servicios o el descifrado de claves cifradas.

OBJETIVO GENERAL

Proveer bases sólidas en la administración de la seguridad del sistema operativo para el uso de herramientas de búsqueda de vulnerabilidades. Con Sistemas de detección de intrusos, análisis de registros, ataques a passwords o secuestro de sesiones en donde se analiza como funciona y las medidas de protección necesarios para evitarlos.

OBJETIVOS ESPECIFICOS

- Verificar el funcionamiento de las restricciones de acceso para cada servicio.
- Utilizar métodos de protección de información y de seguridad para los servicios Web en Intranet como correo DNS, POP3, SMTP, HTTP y TELNET.
- Implementar una infraestructura de soluciones que incluyan servidores de seguridad, comunicación e Intranet.
- Utilizar métodos de rastreos y análisis para evitar intrusos maliciosos de Internet y pérdida de información.
- Utilizar técnicas de rastreo, enumeración y escaneos de puertos que permitan definir un vector de ataque desde Internet.

1. SERVIDOR HTTP

El Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol) es un protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. Atiende a las necesidades de un sistema global de distribución de información como el World Wide Web. Esta soportado sobre los servicios de conexión TCP/IP, y funciona entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (puerto 80), espera las solicitudes de conexión de los clientes Web. Protocolo TCP. Se encarga de mantener la comunicación y garantizar un intercambio d datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (HTML *HyperText Markup Language*, lenguaje de marcas de hipertexto y aplicación CGI), es conocido por su URL.

ETAPAS DE UNA TRANSACCIÓN HTTP

Para una transacción HTTP se analizara los diferentes partes de este proceso. Cuando un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL seleccionando un enlace de un documento HTML o directamente en el campo Location del cliente Web.
- El cliente Web descodifica la URL, separando sus diferentes partes. I identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional puerto 80 y el objetivo el servidor.
- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente. Se realiza la petición y se envía (GET, POST, HEAD), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor.
- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.
- Se cierra la conexión TCP.

Este proceso se repite en cada acceso al servidor HTTP. Por ejemplo, si se recoge un documento HTML en cuyo interior están insertadas cuatro imágenes, el proceso anterior se repite cinco veces, una para el documento HTML y cuatro para las imágenes.

1.1 MÉTODOS DE PETICIÓN

El estándar HTTP utiliza tres comandos que representan las operaciones de recepción, envió de información y chaqueo de estado.

GET: Se utiliza para recoger cualquier tipo de información del servidor. Se utiliza siempre que se pulsa sobre un enlace o se teclea directamente a una URL. Como resultado, el servidor HTTP envía el documento correspondiente a la URL seleccionada, o bien activa un modulo CGI, que genera a su vez la información de retorno.

GET /cgi/saludar.pl?nombre=juan&email=juan@infor.pgr.co HTTP/1.0

HEAD: Solicita información sobre un objeto fichero: tamaño, tipo, fecha de modificación. Es utilizado por los gestores de caché de paginas o los servidores Proxy, para conocer cuando es necesario utilizar la copia que se mantiene de un fichero.

POST: Sirve para enviar información al servidor, por ejemplo los datos contenidos en un formulario. El servidor pasará esta información a un proceso encargado de su tratamiento (generalmente una aplicación CGI). La operación que se realiza con la información proporcionada depende de la URL utilizada en los formularios.

POST /cgi/saludar.pl HTTP/1.0
Accept: */*
nombre=juan&email=juan@infor.pgr.co

Un cliente Web selecciona automáticamente los comandos HTTP necesarios para recoger la información requerida por el usuario. Así, ante la activación de un enlace, siempre se ejecuta una operación GET para recoger el documento correspondiente. El envío del contenido de un formulario utiliza GET o POST, en función del atributo de <FORM METHOD="..">. Además, si el cliente Web tiene un caché de páginas recientemente visitadas, puede utilizar HEAD para comprobar la ultima fecha de modificación de un fichero antes de traer una nueva copia del mismo.

Se han definido algunos comandos adicionales, que sólo están disponibles en determinadas versiones de servidores HTTP, que se pueden utilizar, como por ejemplo: para editar las páginas de un servidor Web trabajando en remoto.

1.2 ARQUITECTURA WEB

Para abrir una página Web en un navegador, se teclea el correspondiente URL en el hiperenlace oportuno. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de ficheros y la envía de vuelta al navegador que la solicitó.

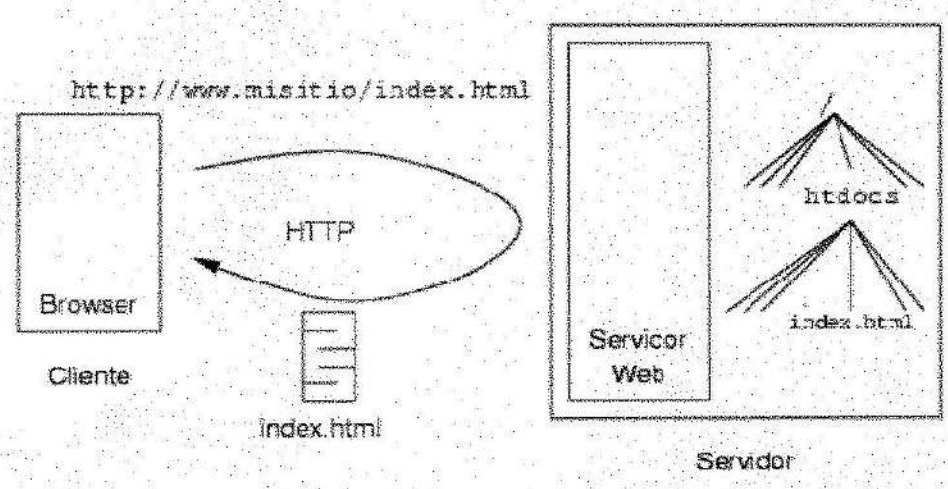

Figura 1. Arquitectura Web básica

NAVEGADOR WEB, BROWSER

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

Posibilidades de ejecución de programas de tipo *script*, con modelos de objetos que permiten manipular los contenidos de los documentos. Estos lenguajes de programación son VBScript, JScript (ambas de Microsoft) y JavaScript (de Netscape), y proporcionan las soluciones llamadas del lado del cliente, *client side* y permiten realizar validaciones de datos recogidos en las páginas antes de enviarlos al servidor y proporcionan un alto grado de interacción con el usuario dentro del documento.

Los navegadores permiten la ejecución de aplicaciones dentro de los documentos mostrados. Las dos posibilidades más populares son la tecnología ActiveX y los *applets* Java. Los applets Java son pequeños programas que se descargan del servidor Web y se ejecutan en la JVM del navegador.

APLICACIONES MULTINIVEL

Las aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma has de ser trasladada a todos los clientes.

La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola.

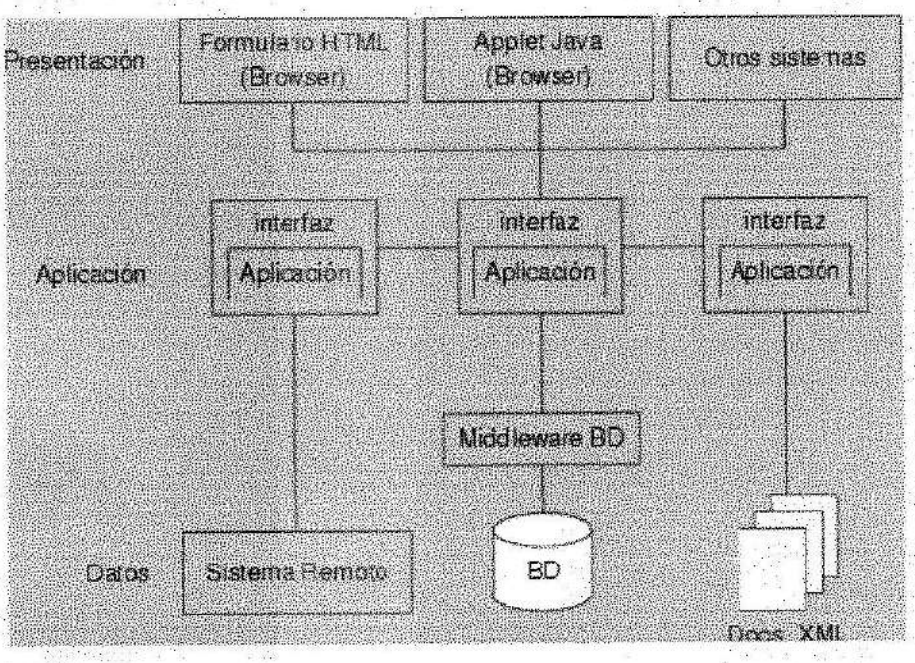

Figura 2. Arquitectura Multinivel.

El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o *script*. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel), cuyo resultado será formateado y presentado al usuario en el navegador.

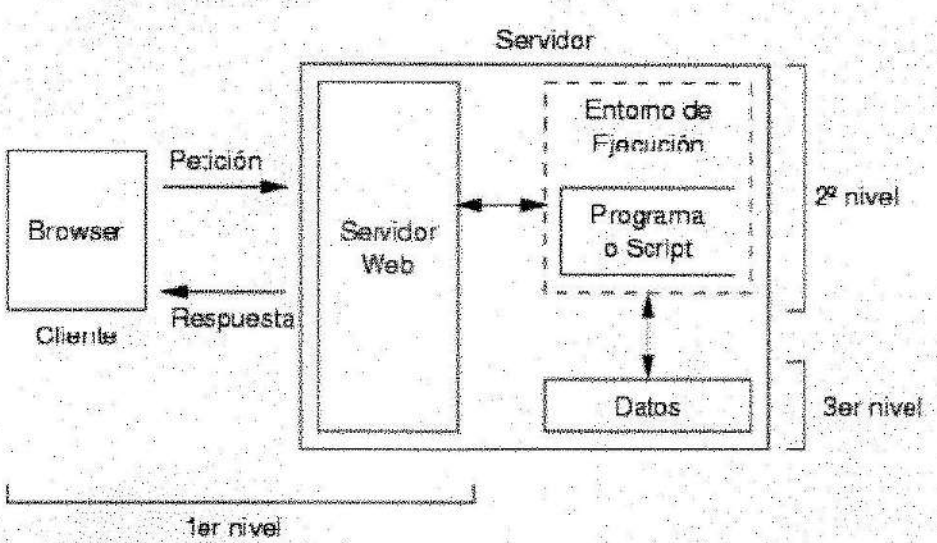

Figura 3. Arquitectura Web de tres niveles.

1.3 INSTALACIÓN DEL SERVIDOR HTTP APACHE

Linux viene con HTTP Apache instalado se debería realizar una actualización o ajuste de la configuración ya que a través de dicho proceso nos lleva a la descarga del código fuente de apache, compilación e instalación, las cuales se estudian en las últimas sesiones.

La versión actualizada del servidor HTTP Apache ya que es la ultima versión Apache 1.3.9. Descargamos el archivo Apache el cual lo situaremos en el directorio de trabajo. Normalmente, /usr/local/usr es la posición usada para almacenar y compilar programas.

Se descomprima el programa con este comando:

[root@ford src] # tar –xzf apache_1.3.9.tar.gz

Esto creara el directorio apache_1.3.9, el cual contendrá todos los archivos necesarios para compilar el programa.

1.4 COMPILACIÓN DE APACHE

El servidor HTTP apache es un sistema con funciones básicas las cuales se encargan de soportar módulos cargables dinámicamente los cuales realizan tareas como la corrección ortográfica de URL dinámica, reescritura de URL, seguimientos de cookies. Este sistema esta en desarrollo constante, por eso siempre debemos observar el archivo Install para ver cual es su configuración por defecto actual y cuales de los módulos están disponible.

1.5 INSTALACIÓN DE APACHE EN SISTEMAS OCUPADOS

Cuando tenemos un sistema con un trafico fuerte tenemos que actualizar un servidor Web lo que significa que la cantidad de tiempo en el que el servidor esta caído es el lapso de tiempo para que se reinicie la nueva versión.
Cuando se realiza estos pasos debemos crear un directorio /usr/local/apache_versión donde la versión es la que instalemos para nuestro ejemplo se crearía el directorio /usr/local/apache_1.3.9 e instalamos apache ahí, para realizarlo debemos modificarlo en la opción prefix cuando se ejecute, /configure ya después de creado el directorio se hace un enlace /usr/local/apache a su /usr/local/apache_1.3.9. Si se desea actualizar se puede instalar una versión modernizada usr/local/apache_1.4.0. Mientras que la versión anterior seguirá activa, si ya hemos configurado la versión usr/local/apache_1.4.0 se debe cambiar el enlace simbólico a usr/local/apache_1.4.0 e iniciar el apache nuevo. Esto hace que baje el tiempo de parada a lo que se tarde en introducir estos comandos, si de pronto surge un error con la nueva versión podemos regresar a la anterior y la inicie de nuevo.

INICIO Y PARADA

Linux cuenta con una opción para iniciar y parar servicios del sistema sin la necesidad de reiniciar esto se puede realizar con el servidor apache.

[rood@ford] # /usr/local/apache/bin/apachectl start → Iniciar

[rood@ford] # /usr/local/apache/bin/apachectl stop → Parar

INICIO DE APACHE EN TIEMPO DE ARRANQUE

Apache como viene preinstalado con Linux se necesita deshabilitar la configuración actual para ejecutar una versión compilada mas nueva, para esto debemos verificar que los enlaces simbólicos apunten al apache existente probablemente pueden llamarse HTTPD o APACHE.

1.6 PRUEBA DE INSTALACIÓN

Si necesitamos probarla lo podemos hacer por medio de la página de inicio por defecto. Para hacer esto reiniciamos el servidor por medio del comando:

[rood@ford/root] # /usr/local/apache/bin/apachectl start

1.7 CONFIGURACIÓN

Este sistema a diferencia de SENDMAIL posee un conjunto de instrucciones fáciles y sencillas de seguir. El cual hace que se facilite la configuración del servidor en varias configuraciones. Esta configuración básica trabaja bien y es aceptable solo debemos crear nuestros documentos HTML ya que permite varias configuraciones a la vez, después de creada la pagina de inicio observaremos como se pueden realizar estas personalizaciones en los archivos de configuración.

ARCHIVOS DE CONFIGURACIÓN

Estos archivos están localizados en el directorio /usr/local/apache/conf donde se observa estos tres archivos:

- srm.cinf
- access.conf
- httpd.conf

CAMBIO DE NOMBRE DE MAQUINAS

Es necesario dar un nombre neutral al servidor y establecer las entradas DNS CNAME o varias entradas de nombre de maquina en el archivo /etc/hosts, con esta opción se puede dar varios nombres para acceder al sistema pero un solo nombre real es el que se necesita saber.

Apache usa SERVERNAME para que lo vuelva como nombre de la maquina del servidor Web, teniendo un alias el cual se haga invisible a los usuarios del sitio.

2. SERVIDOR DNS

El Protocolo Sistema de Nombres de Dominio (*Domain Name System)* permite a los usuarios de una red TCP/IP utilizar nombres jerárquicos y descriptivos para localizar fácilmente ordenadores (*hosts*) y otros recursos en dicha red, evitando de esta manera tener que recordar la dirección IP de cada ordenador al que se desea acceder. Permite delegar el control sobre diferentes segmentos de la base de datos a distintas organizaciones, pero siempre de forma que los datos de cada segmento están disponibles en toda la red, a través de un esquema cliente-servidor.

Los programas denominados servidores de nombres (*name servers*) constituyen la parte servidora del esquema cliente-servidor. Los servidores de nombres contienen información sobre algunos segmentos de la base de datos y los ponen a disposición de los clientes, llamados solucionadores o *resolvers*.

EL ESPACIO DE NOMBRES DE DOMINIO

La base de datos distribuida de DNS está indexada por nombres de dominio. Cada nombre de dominio es esencialmente una trayectoria en un árbol invertido denominado *espacio de nombres de dominio*. La estructura jerárquica del árbol es similar a la estructura del sistema de ficheros UNIX. El árbol tiene una única raíz en el nivel superior llamada raíz (*root*). Cada nodo del árbol puede ramificarse en cualquier número de nodos de nivel inferior. La profundidad del árbol está limitada a 127 niveles.

Cada nodo en el árbol se identifica mediante una etiqueta no nula que puede contener hasta 63 caracteres, excepto el nodo raíz, identificado mediante una etiqueta nula. El nombre de dominio completo de cualquier nodo está formado por la secuencia de etiquetas que forman la trayectoria desde dicho nodo hasta la raíz, separando cada etiqueta de la siguiente mediante un punto. De esta forma, el nombre del nodo especifica su localización en la jerarquía. A este nombre de dominio completo o absoluto se le conoce como *nombre de dominio completamente cualificado* o *Fully Qualified Domain Name* (FQDN). Al ser nula la etiqueta que identifica el nodo raíz, el FQDN de cualquier nodo del árbol siempre acaba con un punto. La única restricción que se impone en el árbol de nombres es que los nodos hijos del mismo padre tengan etiquetas diferentes.

En el esquema jerárquico de nombres DNS, se denomina *dominio* a cualquier subárbol del espacio de nombres de dominio. De esta forma, cada dominio puede contener, a su vez, otros dominios. Generalmente, los hosts están representados por las hojas del árbol, aunque es posible nombrar a un host con una etiqueta correspondiente a un nodo intermedio del árbol.

EL ESPACIO DE NOMBRES DE DOMINIO EN INTERNET

El estándar DNS no impone muchas reglas sobre las etiquetas de los nombres de dominio, ni tampoco asocia un significado determinado a las etiquetas de un determinado nivel del espacio de nombres. Cuando manejamos una parte de este espacio, podemos decidir el significado y la sintaxis de nuestros nombres de dominio. Sin embargo, en el espacio de nombres Internet existente, se ha impuesto una estructura de nombres bien definida, especialmente en los dominios de primer nivel.

Los dominios originales de primer nivel dividían originalmente el espacio de nombres de Internet en siete dominios: com, edu, gov, mil, net, org, e int. Posteriormente, para acomodar el crecimiento y la internacionalización de Internet, se reservaron nuevos dominios de primer nivel que hacían referencia a países individuales.

Actualmente, los dominios originales se denominan *dominios de primer nivel genéricos* y han surgido nuevos nombres que se ajustan a los tiempos que corren.

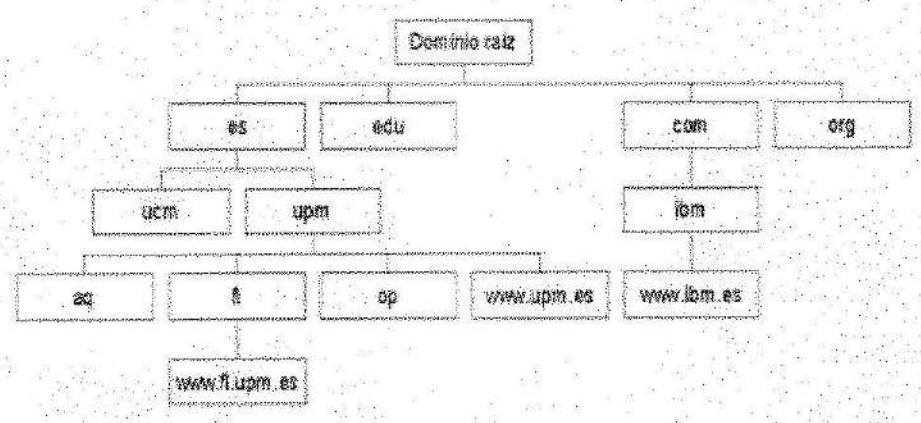

Figura 4. Dominios por niveles

DELEGACIÓN

El sistema de nombres de dominio fue su administración descentralizada. Este objetivo se consigue a través de la *delegación*. La delegación de dominios funciona de forma parecida a la delegación de tareas en una organización. Un responsable de proyecto divide el proyecto en pequeñas tareas y asigna la responsabilidad de las mismas a diferentes empleados.

Una organización que administra un dominio puede dividirla en subdominios. Cada subdominio puede ser delegado a diferentes organizaciones, lo cual implica que esa organización será responsable de mantener los datos (registros de recursos) de ese subdominio. Esa organización puede libremente cambiar los datos e incluso volver a dividir el dominio delegado en subdominios y delegarlos. El dominio padre solamente contiene enlaces a los responsables del subdominio delegado, de forma que pueda hacer referencia a ellos cuando

se le planteen consultas sobre nombres en dicho subdominio delegado.

La subdivisión de un dominio en subdominios y la delegación de dichos subdominios son cosas distintas. En primer lugar, un dominio que tenga capacidad de autogestión (autoridad), siempre puede decidir subdividirse en diferentes subdominios, manteniendo él en principio la autoridad sobre todos ellos. Posteriormente, la organización que gestiona el dominio puede decidir además delegar la autoridad de algunos (o todos) sus subdominios en otras organizaciones. La delegación es una acción que siempre decide el dominio padre, y éste puede revocarla cuando desee, volviendo a retomar la autoridad sobre el subdominio que había delegado.

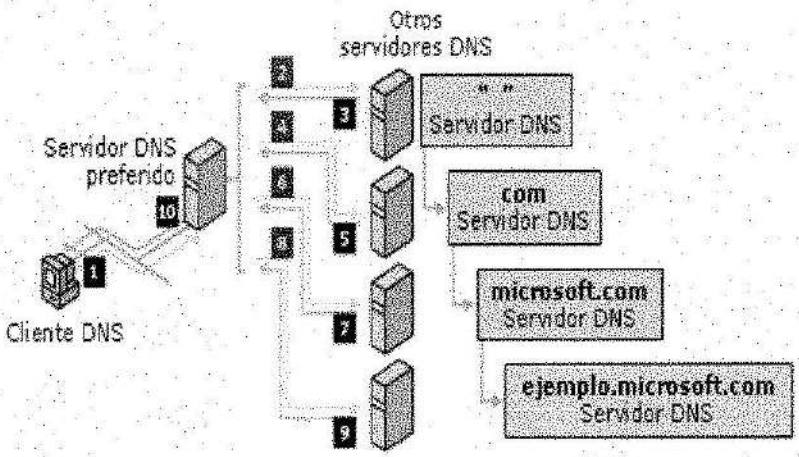

Figura 5. Resolución de nombres de dominios

2.1 INSTALACION DE UN SERVIDOR DE DNS:

El sistema natural de DNS para Linux y la mayoría de los servidores de UNIX es Bind (Berkley Internet Name Domain Server; Servidor de nombres de dominios de Internet Berkley). BIND se conectara con Internet Software Consortium o Consorcio de Software de Internet (ISC) para asegurarse de su desarrollo continuado. Cuando se escribió esto, el ISC estaba en el desarrollo de un servidor DHCP así como en el servidor de noticias INN.

Este software de servidor es estable y sufre pocas actualizaciones. Una vez se ha configurado BIND se descubre tiempo nuevos bugs y temas de seguridad que deberían ser corregidos. Las nuevas características se añaden, pero a menos que las necesite, estas correcciones son menos importantes.

Proceso de baja y compilación de BIND. Si su distribución usa software de gestión de paquetes, si se trabaja con una versión precompilada se debe bajar la fuente segura. BIND aunque esta disponible con el código fuente completo para cualquiera que lo desee para bajarse, puede beneficiarse el soporte y de la gestión de calidad comercial de ISC.
La mayoría de las distribuciones vienen con Bind configurado para no-root, utilizan chroot por defecto. Sin embargo hacer el cambio es sencillo:

-u

Especifica a qué UID cambiará Bind una vez que esté vinculado al puerto 53 para utilizar un usuario llamado 'named' sin permisos de login, similar a 'nobody'

-g

Especifica el directorio al que Bind se hará chroot a sí mismo una vez que esté arrancado. /home/named

Una forma incluso más sencilla de ejecutar Bind con chroot es descargar el paquete Bind-chroot, disponible como paquete de contribución en la mayoría de las distribuciones, e instalarlo. Antes de la instalación, se necesitará un usuario y un grupo llamados named, al cual cambiará el servidor Bind su UID/GID para utilizar groupadd y useradd para crear el usuario/grupo. Algunos paquetes utilizan holelogd para hacer un log de la información de Bind en /var/log/messages.

Otro aspecto de Bind es la información que contiene sobre la red. Cuando alguien hace una petición a un servidor DNS, por lo general envían una petición pequeña por cada información. Por ejemplo: Configurar un fichero named.conf que se ha seguro, en l que haya sido robado del paquete bind-chroot:

```
options {
// Para este chroot se necesitan los siguientes paths
directory "/var/named";
dump-file "/var/tmp/named_dump.db"; // _PATH_DUMPFILE
pid-file "/var/run/named.pid"; // _PATH_PIDFILE
statistics-file "/var/tmp/named.stats"; // _PATH_STATS
memstatistics-file "/var/tmp/named.memstats"; // _PATH_MEMSTATS
// Fin de los paths necesarios
check-names master warn; /* default. */
datasize 20M;
};
zone "localhost" {
type master;
file "master/localhost";
   check-names fail;
   allow-update {
      none;
   };
   allow-transfer {
      any;
   };
};
zone "0.0.127.in-addr.arpa" {
type master;
file "master/127.0.0";
```

```
    allow-update {
        none;
    };
    allow-transfer {
        any;
    };
};
// Denegar y registrar peticiones de versión excepto desde localhost
zone "bind" chaos {
type master;
file "master/bind";
    allow-query {
        localhost;
    };
};
zone "." {
type hint;
file "named.zone";
};
zone "ejemplo.org" {
type master;
file "zones/ejemplo.org";
    allow-transfer {
        10.2.1.1;
        10.3.1.1;
    };
};
```

Bind se ejecuta en el puerto 53, utilizando UDP y TCP. UDP se utiliza para las peticiones normales de dominios. TCP se utiliza para las transferencias de zonas y peticiones más grandes. De modo que filtrar con el cortafuegos el TCP es relativamente seguro y eliminará cualquier transferencia de zonas, pero la petición ocasional al DNS podría no funcionar.

Se podría utilizar named.conf para controlar las transferencias de zonas:

```
ipfwadm –I –a accept –P tcp –S 10.0.0.0/8 –D 0.0.0.0/0 53
ipfwadm –I –a accept –P tcp –S un.host.fiable –D 0.0.0.0/0 53
ipfwadm –I –a deny –P tcp –S 0.0.0.0/0 –D 0.0.0.0/0 53
```

O también se podría manejar así:

```
ipchains –A input –p tcp –j ACCEPT –s 10.0.0.0/8 –d 0.0.0.0/0 53
ipchains –A input –p tcp –j ACCEPT –s un.host.fiable –d 0.0.0.0/0 53
ipchains –A input –p tcp –j DENY –s 0.0.0.0/0 –d 0.0.0.0/0 53
```

En el cual bloquearía las transferencias de zonas y las peticiones grandes, lo siguiente bloquearía las peticiones normales en los que se podrían utilizar un conjunto de reglas.

ipfwadm –I –a accept –P udp –S 10.0.0.0/8 –D 0.0.0.0/0 53
ipfwadm –I –a accept –P udp –S un.host.fiable –D 0.0.0.0/0 53
ipfwadm –I –a deny –P udp –S 0.0.0.0/0 –D 0.0.0.0/0 53

O de esta manera:

ipchains –A input –p udp –j ACCEPT –s 10.0.0.0/8 –d 0.0.0.0/0 53
ipchains –A input –p udp –j ACCEPT –s un.host.fiable –d 0.0.0.0/0 53
ipchains –A input –p udp –j DENY –s 0.0.0.0/0 –d 0.0.0.0/0 53

Se utiliza los nombres de archivo siguientes para almacenar los registros de cada zona:

- Db.dominio: zona de resolución directa.
- Db.direccion: zona de resolución inversa.
- Db.cache: sugerencias de servidores raíz.
- Db.127.0.0.1: resolución inversa de bucle cerrado.

El BIND se distribuye en tres archivos:

1. El código fuente completo
2. La documentación (Bind Operators Guide; GUIA de operación de bind, conocida como "El BOG").
3. Las distribuciones de desarrolladores externos. Las contribuciones externas consiste normalmente en varias herramientas y scripts que simplifican la gestión de BIND o el trabajo con DNS en general.

Estos paquetes de terceras personas se bajan y comprueban como aplicarlos a los entornos particulares.

Por ejemplo: Para crear un directorio que tenga suficiente espacio libre:

/usr/local/src

Bind, una vez bajados estos tres archivos (bind-contrib.tar.gz, bind-doc.tar.gz y bind-src.tar.gz), se desempacan de esta forma:

{root@ford bind} · tar –xvzf bind–□ontrib..tar.gz
{root@ford bind} · tar –xvzf bind-doc.tar.gz
{root@ford bind} · tar –xvzf bind—src.tar.gz

Registros de Recursos (RR)

Para resolver nombres, los servidores consultan sus zonas. Las zonas contienen *registros de recursos* que constituyen la información de recursos

asociada al dominio DNS. Por ejemplo, ciertos registros de recursos asignan nombres descriptivos a direcciones IP.

El formato de cada registro de recursos es el siguiente:

Propietario	TTL	Clase	Tipo	RDATA

- **Propietario**: nombre de host o del dominio DNS al que pertenece este recurso. Puede contener un nombre de host/dominio (completamente cualificado o no), el símbolo "@" (que representa el nombre de la zona que se está describiendo) o una cadena vacía (en cuyo caso equivale al propietario del registro de recursos anterior).
- **TTL**: (*Time To Live*) Tiempo de vida, generalmente expresado en segundos, que un servidor DNS o un resolver debe guardar en caché esta entrada antes de descartarla. Este campo es opcional. También se puede expresar mediante letras indicando días (d), horas (h), minutos (m) y segundos (s). Por ejemplo: "2h30m".
- **Clase**: define la familia de protocolos en uso. Suele ser siempre "IN", que representa Internet.
- **Tipo**: identifica el tipo de registro.
- **RDATA**: los datos del registro de recursos.

2.2 DOCUMENTACION DE BIND

El directorio **doc** contiene varios textos de documentación de BIND y todos los documentos RFC relevantes que especifican los estándares de DNS. El directorio contiene los subdirectorios siguientes:

- **Bog:** El elemento mas notable en el archivo **doc** es la guia del operador de bind, mas conocido como BOG. Dentro del subdirectorio **bog**, encontrara dos archivos que representan el documento: **file.list**, el cual es un documento ASCII legible con un editor de texto y el **file.psf**, el cual es un archivo PostScript que se puede enviar a su impresora.
- **Html**: El subdirectorio **html** contiene la documentación Online de bind 8. Es mejor leer el archivo **index.html** mediante un navegador Web y seguir los enlaces (todos los enlaces de documentación apuntan a otros archivos del mismo directorio).
- **Man**: El subdirectorio **man** mantiene todas las páginas del **man** para las herramientas que vienen con BIND.
- **Misc**; El directorio **misc** contiene documentación que no se encuentra

acomodo junto a otros documentos. La FAQ de BIND esta aquí, así como algunas paginas blancas en forma Postscript este sirve para que en tal caso de que no haya una impresora, se puede usar Ghost View para leer estos archivos. Se comprueba las paginas del man de **gv** para mayor información.

- **Notes**: Estas son notas hechas por programadores para programadores.
- **Old**: Este archivo contiene documentación de versiones anteriores de BIND.
- **Port**: Mire aquí la información acerca de los sistemas operativos particulares. La gente de Linux no necesita preocuparse por esto: Bind es estable desde los dias de Linux 1.2.
- **Rft**: Este subdirectorio contiene documentos RFC relativos a la funcionalidad. De BIND.
- **Tmac**: Las macros troff se guardan aquí. Troff es un sistema sofisticado que le permite generar documentos con formato. Varios textos de la documentación BIND se escriben en troff (y se convierten a PostScript para visualizarlos). A menos que desee profundizar en troff, puede dejar este directorio a un lado.

2.3 COMBINACION BIND

El proceso descompilación de BIND se encuentra automatizado a través del uso de la utilidad **make.** Una vez que desempaqueta el código fuente y usa el comando **cd** para entrar en el directorio **src**, necesita decidir de posición de los archivos temporales generados por BIND durante el proceso de construcción. El director **/tmp/bind** puede cambiar la localización a otro sitio donde normalmente sitúe los archivos temporales, adoptando el comando en consecuencia. Para iniciar la compilación desde el directorio src:

[root@ford src] # make DST=/tmp/bind SRC='tmp' links
Una vez que se termine la ejecución del comando, se cambia el directorio /tmp/bind:

[root@ford src] # cd /tmp/bind

Entonces nos aseguramos de que no hay archivos, con el comando siguiente:

[root@ford bind] # make clean

Make necesita actualizar algo de su información de configuración basado en su directorio nuevo. Con el comando:

[root@ford bind] # make depend

Y luego se introduce el comando que realice la compilación final:

[root@ford bind] # make

Una vez que se compile todo, se usa el comando para copiar los archivos del programa BIND en el lugar correcto del sistema. El comando que se maneja para manejar la instalación es:

[root@ford bind] # make install.

Herramientas de instalación de Bind:

HERRAMIENTAS INSTALADAS CON LA COMPILACION/INSTALACION DE BIND	DESCRIPCION
• /usr/bin/addr	Convierte las direcciones IP a valores hexadecimales, y viceversa.
• /usr/bin/nslookup	Permite resolución de maquina a IP y de IP a nombre de maquina, así como otras búsquedas de registros relacionados con DNS, todos desde la línea de comandos.
• /usr/bin/dig	Muestra los procesos de búsqueda con los cuales se resuelven la direcciones, incluyendo todos los servidores DNS intermedios involucrados.
• /usr/bin/dnsquery	Una alternativa a nslooup.
• /usr/bin/host	Otra alternativa a nslookup.
• /usr/bin/named	El proceso servidor responsable de proporcionar los servicios de DNS.
• /usr/bin/named-xfer	Un programa auxiliar ejecutando por named para conseguir información de otros servidores DNS. Esto proporciona una herramienta poderosa para depurar las zonas de transferencia, un mecanismo por el cual un servidor DNS recibe datos de todas las maquinas en las tablas de otro servidor DNS.
• /usr/bin/ndc	Un programa para controlar el programa named.
• /usr/bin/nsupdate	Le permite actualizar la información en la tabla de DNS en un servidor en funcionamiento.
• /usr/bin/named-bootconf	Una utilidad para convertir archivos de configuración de BIND versión 4 a Bind versión 8.

Tabla 1. Archivos instalados con la compilación/instalación de BIND

2.4 CLIENTE DNS

La configuración del cliente de DNS maneja solo dos archivos involucrados: /etc/resolv.conf y /etc/nsswitch.conf.

El archivo /etc/resolv.conf: Contiene la información necesaria para que el cliente sepa cual es su servidor DNS local. Cada sitio debería tener, al menos, su propio servidor DNS de caché. Este archivo tiene dos líneas: la primera indica el dominio de búsqueda por defecto, y la segunda indica la dirección IP del servidor.

El dominio de búsqueda por defecto se aplica ala mayoría de los sitios que tienen sus propios servidores locales. Cuando se especifica el dominio de búsqueda por defecto, el cliente añade automáticamente este nombre de dominio al sitio requerido y lo comprueba primero. Por ejemplo, especificamos que nuestro dominio por defecto es yahoo.com y tratamos de conectarnos al nombre de la maquina MY, el software del cliente tratara de contactar automáticamente con my.yahoo.com. Si usamos lo mismo por defecto, y tratamos de conectarnos a www.stat.net, el software probara con www.stst.net.yahoo.com (un nombre de maquina perfectamente lega), encontrara que no existe y probara entonces con www.stat.net solo (que si existe). El proceso de búsqueda un poco porque cada dominio ser comprobado. Por ejemplo, si se especifican yahoo.com y standford.edu, y realizaremos la búsqueda de www.stat.net, realizaremos tres búsquedas: www.stat.yahoo.com, www.stat.net.standford.edu y www.stat.net.

El formato del archivo /etc/resolv.conf es como sigue:

Searcg domainname
Nameserver IP-address

Donde domainname es el nombre del dominio por defecto donde se busca, e IP-address es la dirección IP del servidor DNS. Por ejemplo este es mi /etc/resolv.conf:

Search planetoid.org
Nameserver 127.0.0.1

Cuando se conecta a zaphod.planetoid.org, solo necesito especificar zaphod como nombre de maquina. Puesto que ejecuto el servidor de nombres en mi misma maquina y especifico de maquina local.

El archivo /etc/nsswitch.conf: El archivo /etc/nsswitch.conf le dice l sistema donde buscar cierta clase de información de configuración (servicios). Cuando se identifican varias localizaciones, el archivo /etc/nsswitch.conf también especifican el orden en el cual se puede encontrar mejor información. Los archivos de configuración típicos que se configuran utilizando el etc/nsswitch.conf incluye el archivo de contraseñas, el archivo de grupos y el archivo de Hosts (para ver la lista completa, abra el archivo con un editor de textos).

El formato del archivo etc/nsswitch.conf es simple. Los nombres de servicio vienen primero en la línea (se observa que el archivo etc/nsswitch.conf se aplica a algo mas que las búsquedas de nombres de maquinas), seguido por puntos. Entonces vienen las localizaciones que contienen la información. Si se identifican varias localizaciones, las entradas se listan en el orden en el cual el sistema realiza la búsqueda.

Las entradas validas para las localizaciones son files, nis, dns, [NOTFOUND] y [NISPLUS]. Los comentarios se empiezan con el símbolo almohadilla (#). Por ejemplo, si se abre el archivo con un editor de texto, vera una línea similar a esta:

Host: files nisplus nis dns

Esta línea le dice al sistema que las búsquedas de nombre de maquinas deberían empezar por el archivo /etc/host. Si la entrada no se puede encontrar aquí, se comprueba NISPLUS. Si la maquina no se encuentra pro medio de NISPLUS, se comprueba NIS y así sucesivamente. Es posible que NISPLUS no se ejecute en su maquina y que quería que el sistema compruebe los registros de DNS antes de comprobar los registros NIS. En este caso, cambie la línea a:

Host: files dns nis

Se guarda el archivo y el sistema detectara automáticamente el cambio.

USO DE [NOTFOUND]: En el archivo /etc/nsseitch.conf, vera entradas que terminan con [NOTFOUND]. Es una directiva especial que permita parar el proceso de búsqueda de información después de que el sistema falle con todas las entradas anteriores .Por ejemplo, si su archivo contiene la línea de host: files [MOTFOUND] dns nis, el sistema tratara de buscar información de maquina solamente en el archivo /etc/hosts. Si la información de maquina solamente en el archivo /etc/hosts. Si la información pedida no esta allí, NIS y DNS no se buscara.

El archivo /etc/named.conf: Es el principal archivo de configuración de BIND. Basado en las especificaciones de este archivo, BIND determina como debería actuar y que archivos de configuración adicionales. Se encontrara una guía completa del nuevo formato del archivo de configuración del directorio html de la documentación de BIND. El formato general del archivo /etc/named.conf es el siguiente:

```
Sentencia [
        Opciones;    //comentarios
]
```

La palabra sentencia le dice a BIND que vamos a describir una faceta particular de su operación y opciones son los comandos especificados aplicados a sentencia. Las llaves se requieren para que BIND sepa que opciones se

relacionan con cada sentencia. Hay un punto y coma después de cada opción y después de cada una de las llaves de cierre.

2.5 SENTENCIAS

Puede usar las siguientes palabras clave sentencia:

ACL: Lista de control de acceso. Un mecanismo para determinar que clase de acceso tienen los otros al servidor DNS.

INCLUDE: Le permite incluir otro archivo y quien se trate a dicho archivo como parte del archivo /etc/named.conf normal.

LOGGING: Especifica sobre que información se hará log y cual se ignorara. Para la información de log, también puede especificar donde se guardara.

OPTIONS: Para temas de configuración del servidor globales.

CONTROLS: Le permite declarar canales de control para el uso de la utilidad ndc.

SERVER: Configura opciones del servidor especificas.

ZONE: Define una zona DNS.

2.6 TIPOS DE REGISTROS DE DNS

Los tipos de registros de DNS: SOA, NS, A, PTR, CNAME, MX, TXT Y RP.

SOA: START OF AUTHORITY (INICIO DE AUTORIDAD)

El registro SOA se inicia la descripción de las entradas de un sitio DNS. El formato de esta entrada es el siguiente:

```
Domain.com  in soa ns.domain.com  hostnamer.domain.com (
1999080801 ; serial number
10800       ; refresh rate in seconds (3 hours)
      1800          ; retry in seconds (30 minutes)
      1209600       ; expire in seconds (2 weeks)
      604800        ; minimum in seconds (1 week)
```

La primera línea consta de algunos detalles a los que necesitamos prestar atención:

- **Domain.com**: Es lo que será reemplazo con su nombre de dominio. El punto final es necesario para que el servidor diferencie entre nombres de

maquinas relativos de los FQDN. Por ejemplo, la diferencia entra oid y oid.planetoid.org

- **IN:** Le dice al servidor de nombres que este es un registro de Internet. Hay otros tipos de registro, pero es muy difícil que tenga necesidad de ellos.
- **SOA:** Le dice al servidor de nombres que es un registro de "Inicio de autoridad.
- **Ns.domain.com:** Es el FQDN del servidor de nombres para este dominio, (que debería ser el servidor donde reside el archivo). Se debe fijar donde se deja el punto.
- **Hostname.domain.com:** Es la dirección de correo del administrador del dominio. El símbolo @ es reemplazado por un punto. Así, la dirección de correo preferida en este ejemplo es hostiame@domain.com. El punto final se usa también.

El resto de los registros empiezan después del paréntesis de la primera línea. La primera línea es el número de serie. Se usa para decirle al servidor de nombres que el archivo se ha actualizado. No se incrementa este numero cuando haga cambio es un error frecuente en el proceso de gestión de los registros DNS.

La segunda línea es una lista de valores de tiempo de refresco en segundo. Este calor le dice a los servidores secundarios con que frecuencia deben buscar al servidor primario para ver si los registros se han actualizado.

El tercer valor es el tiempo de reintento en segundos. Si el servidor secundario trata, pero no consigue conectarse con el servidor DNS primario para comprobar las actualizaciones, el servidor secundario tratará de hacerlo de nuevo después del número de segundos especificados.

El cuarto valor esta pensando para los servidores secundarios que tienen caché en los datos de zona. Le dice a estos servidores que si no pueden conectarse con el servidor primario para una actualización, deberían descartar el valor después del número de segundos especificados. De una a dos semanas es un buen valor para este intervalo.

El valor final, mínimo, le dice a los servidores de caché cuando esperar antes de que expire una entrada sin no puede conectarse con el servidor DNS primario. De cinco a siete días es un valor bueno para esta entrada.

NS: NAME SERVER (NOMBRE DELSERVIDOR)

El registro NS se usa para especificar cual servidor de nombres mantiene los registros para esta zona. El formato de este registro es el siguiente:

```
IN    NS        ns1.domain.com.
IN    NS        ns2.domain.com.
```

Puede tener tantos servidores de respaldo como desee para un dominio. La mayoría de los ISP actuaran como servidores DNS secundarios si le proporcionan conectividad.

A: ADDRESS RECORD (REGISTRO DE DIRECCION)

El registro A se usa para proporcionar un mapeado de nombre de maquina a dirección IP. El formato de una dirección A es simple:

Nombre-maquina IN A dirección-IP

Por ejemplo, un registro A para la maquina oid.planetoid.org, cuya dirección IP es 192.168.1.2 debería parecerse a esto:

Oid IN a 192.168.1.2

Cualquier nombre de maquina se le añade un sufijo con el nombre de dominio indicado en el registro SOA, a menos que el nombre de maquina acabe con un punto. En el ejemplo de oid, el registro SOA de arriba es para planeoid.org, entonces oid que es oid.planeoid.org. si queremos cambiar esto a oid.planeoid.org se haría sin el punto, el servidor de nombres debería entenderlo comooid.planeoid.org.planeoid.org. Así que si quiere usar FQDN, se asegura de añadir el sufijo con el punto.

PTR: POINTER RECORD (REGISTRO DE PUNTERO)

El registro PTR es para realizar resolución de nombres inversa, permitiendo de esta forma que alguien especifique una dirección IP y determine el nombre de maquina correspondiente. El formato para este registro es muy similar al registro A, excepto por que los valores están invertidos:

Dirección-IP IN A nombre-maquina

La dirección IP puede tener dos formatos: el octeto ultimo de la dirección IP, dejando que el servidor de nombres añada al sufijo con la información del nombre de dominio in-add.arpa, o la dirección IP completa, La cual se añade con un punto. El nombre-maquina debe ser el FQDN completo. Por ejemplo, el registro PTR para la maquina oid debería ser el siguiente:

192.168.1.2 IN A oid.planeoid.org.

MX: MAIL EXCHANGER (INTERCAMBIADOR DE CORREO)

El registro MX esta encargado de decirles a otros sitios cual es el servidor de correo de la zona. Si una maquina de la red genera un correo saliente con su nombre de maquina en el, alguien responderá con un mensaje de vuelta directamente a esta maquina. En su lugar, el servidor de correo de respuesta debería buscar el registro MX de ese sitio y enviar el mensaje allí.
Cuando los sitios de Internet estaban principalmente compuestos de sistemas

UNIX, con el Sendmail configurado como una maquina NULL remitiendo a un concentrador de correo, la última falta de un registro MX no se nota. Pero cuando se unieron más sistemas no UNIX a la red, los registros MX se hicieron cruciales. Si el PC (el cual puede no aceptar correo SMTP), es importante que la respuesta tenga una forma fiable de saber la identidad del servidor de correo de pc.domain.com., el formato de un registro es el siguiente:

Nombre-dominio. IN peso nombre-maquina

Donde nombre-dominio, es el nombre del dominio con un punto final; el peso es la importancia del servidor de correo, si existen varios servidores de correo, el que tiene el numero mas pequeño tiene preferencia sobre los que tienen un numero mayor; y nombre-maquina es el nombre del servidor de correo. Es importante que el nombre-maquina tenga un registro A. Este es un ejemplo de entrada:

Domain.com.	IN	MX	mailserver1
	IN	MX	mailserverbackup

Los registros MX se encuentran en lo más alto de los archivos de configuración de DNS. Si no se especifica un nombre de dominio, el nombre por defecto se toma del registro SOA.

CNAME: CANONICAL NAME (NOMBRE CANONICO)

Los registros CNAME le permiten crear alias de sus nombres de maquinas. Es útil cuando quiera proporcionar un servicio disponible con un nombre fácil de recordar, pero seguir dándole a la maquina el nombre real.

Otro uso popular de CNAME es para crear un servidor nuevo con un nombre fácil de recordar sin tener que invertir en un servidor en un servidor nuevo. Un ejemplo: un sitio tiene un servidor de correo llamado mailhost, cuando la gente que no usa UNIX entre en el sitio, asumimos que el servidor de correo se llamara mail. Para acomodar esta asunción, se cera un CNAME en un lugar de renombrar el servidor, de forma que todas las peticiones a la maquina mail resuelvan transparentemente a mailhost. Este es el formato de un registro CNAME:

Name-nuevo IN CNAME nombre-antiguo

Por ejemplo, para el mapeado mail a mailhost mencionado arriba, nuestras entradas se parecían esto:

Mailhost	IN	A	192.168.1.10
Mail	IN	CNAME	mailhost

RP Y TXT: LAS ENTRADAS DE DOCUMENTACION

Algunas veces es útil proporcionar información de contacto como parte de datos, no como comentarios, sino como registros reales que otras personas puedan buscar. Esto se puede conseguir gracias a los registros RP Y TXT.

Los registros TXT son estradas de texto en las que puede situar toda la información que considere oportuna. Con mucha frecuencia, solo querrá poner toda la información de contacto en estos registros. Cada registro TXT debe relacionarse con un nombre de maquina n particular. Por ejemplo:

```
Hhgttg.planetoid.org.    IN    TXT    "Contacto: Marvin"
                         IN    TXT    "Administrador/Android"
                         IN    TXT    "Telefono: 800-55-1212"
```

El registro RP se creo como un contenedor explicito de la información del contacto de la maquina. Este registro expone quien es la persona responsable de una maquina especifica. Este es un ejemplo:

```
Hhgtt.planetoid.org.     IN    RP     marvin.domain.com  planetoid.org.
```

Las entradas que necesitamos en el archivo /etc/named.conf y sabemos todo lo relativo a los tipos de los registros DNS. Es ahora de crear la base de datos real que alimentara al servidor. El formato de un archivo de base de datos no es demasiado estricto peor se deben seguir convenciones. Ajustarse a estas convenciones le harán la vida mas fácil y aislada el camino el administrador que asuma esta creación.

Cada archivo de base de datos empezara con un registro SOA y contendrá, al menos, un registro NS. Todo lo demás es opciones; se encontraran el siguiente formato general:

```
SOA          registros
NS           registros
MX           registros
A y CNAME    registros
```

Por ejemplo, este es el archivo de configuración completo de una zona para un dominio simple con cuatro maquinas:

```
@    IN    SOA  domain.com.          hostname.domain.com.  (
                1999022300; número de serie
                10800; refresco cada 3 horas
                1800; reintento cada 30 minutos
                1209600; expira en 2 semanas
                604800); mínimo 1 semana
```

```
                IN      NS              ns.domain.com.
                IN      MX 10       mail.domain.com.
Imp             IN      A           192.168.1.1 ;       Pasarela a Internet
Mail            IN      A           192.168.1.2 ;       Servidor de correo
Technics        IN      A           192.168.1.3 ;       Servidor Web
www             IN      CNAME           technicas
ns              IN      A           192.168.1.4 ;       Servidor de nombres
peanutbutter IN         a           192.168.1.5 ;       Firewall
```

Y este es el archivo inverso correspondiente:

```
@       IN      SOA  1.168.192.in-addr.arpa.    hostname.domain.com. (
                              1999010501 ;      Serie
                              10800      ;      Tiempo de refresco (3 horas)
                              1209600    ;      Expira (2 semanas)
                              604800)    ;      Mínimo (1 semana)
        IN      NS   ns.domai.com.
1       IN      PTR  imp.domain.com.
2       IN      PTR  mail.domain.com.
3       IN      PTR  technics.domain.com.
4       IN      PTR  ns.domain.com.
5       IN      PTR  peanutbutter.domain.com
```

2.7 UNA CONFIGURACION COMPLETA

Con un poco hemos dado una idea sobre el gran esquema de DNS y le hemos proporcionado una guía para construir sus propios archivos de configuración. Esto es una configuración completa de un dominio primario (domain.com) que también actúan como secundario de una amigo, example.com. En cambio, el dominio example.com actúa como nuestro secundario. Ocurren diferentes transferencias de zona entre nuestros dos sitios, pero no con otros sitios. El ISP del sitio de domain.com proporciona servicios de DNS también a los cuales hacemos peticiones de remitente cuando no podemos resolver la información por nosotros mismos. Este es el archivo /etc/named.conf:

```
Options
        Directory "/var/named";
        Forwarders {
                        192.168.2.1;
                        192.168.2.2;
        };
        Allow-transfer {10.0.0.1; 0.0.0.2; } // el servidor e nombres de
example.com
        };
//
// Una configuración de un servidor de nombres solo de caché
//
Cone "." {
        Type hint;
```

```
        File "name.ca";
};
Zone "0.0.127.in-addr.arpa"{
        Type master;
        File "name.local";
};
//
// Información de nuestro primario
//
Zone "domain.com"
        Type master;
        File "named.comain.com";
};
Zone "1.168.in-add.arpa" {
        Type master;
        File "named.rev";
};
//
// Información de nuestro secundario para example.com
//
Zone "example.com
        Type slave
        File ""example.com.cache";
        Masters (10.0.0.1; 10.0.0.2 ;)   };
};
```

Este es el archivo /var/named/named.ca:

```
; This file Lolds the information on root name servers hended to
; initialize cache of Internet domain name servers
(e.g. reference this file in the "cache . <file>"
Configuration file of BIND domain name servers).

; This fike is made available by InterNIC registration services
; Under anonymous FTP as
;       File                    /domain/named.root
;       On server               ftp.RS.INTERNIC.NET
;       -OR- under Gpher at     RS.INTERNIC.NET
;       Under menu              InterNIC Registration Arcives
;       Submenu                 InterNIC Registration Archives
;       File                    named.root
;       last update:  Aug 22, 1997
;       related verdion of root zone:       1997082200
;
; formerly NS.INTERNIC.NET.
;
.                       3600000     IN    NS    A.ROOT-SERVER.NET.
A.ROOT-SERVERS.NET  3600000               A     198.41.0.4
;
; formerly NS1.ISI.EDU
```

```
;
.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000      A     128.9.0.107
;
; formarly C.PSI.NET.
;
.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000      A     192.33.4.12
;
.                       3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                       3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET. 3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                       3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET. 3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                       3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET. 3600000      A     192.112.36.4
;
; formerly AOS.ARLARMY.MIL
;
.                       3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET. 3600000      A     128.63.2.53
;
; formerly  NIC.NORDU.NET
;
.                       3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.  3600000      A     192.36.148.17
;
; temporarily housed at ISI (InterNIC)
;
.                       3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.  3600000      A     192.41.0.10
;
; house in LINX, operated by RIPE NCC
;
.                       3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET. 3600000      A     193.0.14.129
;
; temporality housed at ISI (IANA)
.                       3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET. 3600000      A     198.32.64.12
```

3. SERVIDOR POP3

El protocolo POP significa "Protocolo de Oficina de Correos (Post Office Protocol)" y se encarga de listar mensajes, recibirlos y borrarlos. También se manejan el IMAP que significa Internet Message Access Protocol que sirve para la recepción este protocolo esta relacionado con el POP. Existen diferentes servidores POP disponibles para Linux, el original que viene con la mayoría de las distribuciones suele ser adecuado para la mayoría de los usuarios.

POP fue creado como resultado de dos desarrollos:

- La proliferación de Internet en PC no UNIX que no tenia la capacidad de ejecutar un servidor SMTP completo SENDMAIL.
- La proliferación de estaciones de trabajo conectadas a Internet mediante facilidades de dial-up al 100 por le 100 del tiempo.

Los problemas principales con POP son similares a muchos otros protocolos; los nombres de usuarios y sus contraseñas se transmiten en texto claro, el cual es el objetivo para un SNIFFER de paquetes. El POP se puede "SSLificar", sin embargo no todos los clientes de correo soportan POP seguro mediante SSL. La mayoría de los servidores POP vienen configurados para utilizar TCP_WRAPPERS, el cual se encarga de restringir el acceso. El sistema POP es similar a root este accede a los buzones pero se han detectado varios ataques de root a varios servidores POP. POP se ejecuta en el puerto 109 y 110, también maneja el puerto 109 pero está obsoleto), siempre y cuando utilice el protocolo TCP. El servidor es en ASCII lo cual significa que es fácil de comprobar la funcionalidad de un servidor POP3 usando el TELNET.

Como por ejemplo:

```
ipfwadm –I –a accept –P tcp –S 10.0.0.0/8 –D 0.0.0.0/0 110
ipfwadm –I –a accept –P tcp –S un.host.fiable –D 0.0.0.0/0 110
ipfwadm –I –a deny –P tcp –S 0.0.0.0/0 –D 0.0.0.0/0 110
```

O se manejaría de esta manera:

```
ipchains –A input –p tcp –j ACCEPT –s 10.0.0.0/8 –d 0.0.0.0/0 110
ipchains –A input –p tcp –j ACCEPT –s un.host.fiable –d 0.0.0.0/0 110
ipchains –A input –p tcp –j DENY –s 0.0.0.0/0 –d 0.0.0.0/0 110
```

Este servicio de POP3 se encarga del correo entrante, ya que si alguien envía un mail a llmfabrega@dominio.org, POP3 recibe este email y lo guarda al directorio home del usuario correspondiente dentro de un fichero llamado mbox este quedaría guardado en /home/profes/llmfabrega/mbox. Después nosotros como usuario, desde cualquier lugar nos podríamos conectar al servidor pop3 y después de identificarnos con nuestro login y contraseña, podemos visualizar los mensajes que se encuentren en su mbox. Sí, mbox es un archivo de texto que guarda todos los emails recibidos, de tal manera que si deseas guardar los mails, se van acumulando en este fichero, y si más tarde se quiere leer un mail

de hace un mes, tendremos que buscar en este fichero hasta se encuentre guardado. También se puede decir que mbox es un sistema de almacenado un poco frágil.

También podemos decir que existe otro sistema para almacenar los mails recibidos, es el sistema de Maildir. Este sistema almacena los mensajes recibidos en una carpeta llamada "new" dentro del directorio Maildir que se encuentra en el home de cada usuario. Este sistema se diferencia del anterior por que no guarda los emails en un solo fichero como mbox, sino que cada email tiene su propio fichero. De esta manera es más fácil trabajar con los emails recibidos.

31. MODELO DE COMUNICACIÓN DE POP

El protocolo POP y su modelo de comunicaciones, se encuentra en el documento oficial RFC 1725. Este modelo de comunicaciones se basa en el concepto de buzón, que posee un espacio para almacenar los mensajes de correo hasta que se solicite la descarga de estos mensajes. El cliente POP se conecta con el servidor a través del puerto TCP, 110. Para conectarse al servidor, es necesario una cuenta de identificación en dicha máquina (lo que le permite tener un espacio reservado para sus correos). A continuación es necesario verificar que es dueño de la cuenta a través de una clave. Una vez conectado al sistema, el cliente POP puede dialogar con el servidor para saber, en otros, si existen mensajes en la casilla, cuántos mensajes son o para solicitar la descarga de alguno de ellos. Para poder ofrecer estas funciones, el modelo de comunicación POP se basa en estados. Estos son estado de autorización, estado de transacción y estado de actualización.

Después de establecer la conexión, el servidor POP se encuentra en un estado de autorización, esperando que el cliente le envíe el nombre y clave de la cuenta de usuario. Cuando se verifica que el nombre y la clave son correctos, el servidor pasa a un estado de transacción. Antes de pasar a este estado, el servidor POP bloquea el buzón para impedir que los usuarios modifiquen o borren el correo antes de pasar al estado siguiente. En este estado de transacción el servidor atiende las peticiones del cliente. Después de enviar al servidor el comando QUIT, explicado más adelante, el servidor pasa al estado de actualización (estado siguiente). En este estado el servidor elimina los mensajes que están con la marca de borrado y finaliza la conexión.

3.2 COMANDOS POP

El protocolo establece un conjunto reducido de comandos, añadiendo en versiones posteriores algunas funcionalidades adicionales. El diálogo desde el cliente al servidor, se basa en el envío de comandos, a los que el servidor responde con código y cambiando, cuando corresponda, de un estado a otro. Lo que el protocolo busca es conocer si los comandos funcionan, por tanto, sólo se establecen dos códigos de respuesta, uno para cuando el comando funciona correctamente y otro para cuando no es así. Los códigos de respuesta

que el servidor POP envía, van seguidos de una frase que explica o aclara el código, lo que puede ayudar a conocer cual es el motivo de los errores, si se producen. El Código de respuesta es el siguiente:

+OK
    El comando funcionó correctamente
+ERR
    El comando falló

Los comandos POP se pueden agrupar según el estado en el que se encuentre el servidor, así se tendrá comandos del estado de autorización, comando del estado de transacción, comandos del estado de actualización y comandos opcionales.

3.3 COMANDOS DEL ESTADO DE AUTORIZACIÓN

Al conectarse a un servidor POP, éste entra en un estado de autorización. El cliente de correo debe enviar el nombre de la cuenta y la clave para poder continuar. Si son correctos, la casilla correspondiente a esa cuenta pasa a un estado de bloqueo exclusivo, para impedir que los mensajes sean modificados o borrados antes de llegar al estado de actualización del servidor POP. Si no se consigue pasar la casilla al estado de bloqueo exclusivo, se produce un fallo y no se puede pasar al estado de transacción. PASS (Clave) señala al servidor la clave de la cuenta de usuario indicada por el comando USER. Si la clave no es correcta o la casilla no pasa al estado de bloqueo exclusivo, se produce un error. La sintaxis de este comando es la siguiente:

PASS clave#13#10

QUIT se puede usar cuando el servidor está en estado de autorización y en estado de transacción. Si se usa cuando está en estado de autorización, la sesión finaliza y se interrumpe la conexión. Si se usa cuando está en estado de transacción, se cierra la sesión y el servidor pasa a estado de actualización. La sintaxis de este comando es la siguiente:

QUIT#13#10

USER le proporciona al servidor el identificador o nombre de la cuenta de usuario. Si ese identificador existe, devuelve un código de operación correcta, de lo contrario, devuelve un código de fallo.

USER id-cuenta#13#10

3.4 COMANDOS DE ESTADO DE TRANSACCIÓN

El cliente puede enviar comandos para correo nuevo, borrar correo (marcar como borrado), recuperarlo, almacenarlo. DELE (Eliminar) marca como eliminado un mensaje, pero en realidad el servidor no lo elimina hasta que no pasa al estado de actualización, con lo que no se pierde en el caso de que la conexión falle. Cada mensaje que está en la casilla del servidor POP tiene asignado un número, que lo identifica. La sintaxis es la siguiente para este comando:

DELE numero_mensaje#13#10

LIST recupera información acerca del tamaño que ocupa un mensaje determinado o todos los mensajes. En el caso de que se aplique sobre un solo mensaje, el servidor responde con una línea indicando el número del mensaje y el tamaño. Si se refiere a más de un mensaje, el servidor responde enviando una línea por cada mensaje que incluye el número y tamaño correspondiente. El final de estas líneas es un punto y seguido por los caracteres #13#10. La sintaxis de este comando es:

LIST [numero_mensaje]#13#10

NOOP (No operación) es un comando de no operación. Cuando se envía, el servidor responde con un OK. Se utiliza para mantener activa la sesión. La sintaxis del comando es la siguiente:

NOOP#13#10

RETR (Recuperar) permite recuperar o solicitar que el servidor envíe un mensaje determinado. El mensaje se solicita enviando el número del mensaje a continuación del comando. Este número de mensaje no puede corresponder a un mensaje con marca de borrado. El servidor responde a la petición enviando el texto del mensaje, que finaliza cuando le llega al cliente un punto seguido de los caracteres de retorno de carro/avance de línea (#13#10). La sintaxis del comando es:

RETR numero_mensaje#13#10

RSET (Reiniciar) anula la marca de borrado de todos los mensajes en la casilla. No se puede eliminar la marca de borrado de un mensaje en concreto, tiene que ser de todos. La sintaxis es la siguiente:

RSET#13#10

STAT (Estado) permite obtener un resumen del contenido de la casilla. El servidor responde a este comando enviando el número de mensajes que hay en la casilla, sin contar aquellos que están marcados como borrados, y el volumen o tamaño en bytes de la casilla. La sintaxis de este comando es:

STAT#13#10

3.5 COMANDOS DEL ESTADO DE ACTUALIZACIÓN

En este estado no hay comandos. A este estado se llega desde el estado de transacción cuando se envía al servidor el comando QUIT. En este estado de actualización se eliminan los mensajes que han sido marcados en el estado anterior. A continuación se le quita el bloqueo exclusivo a la casilla para que pueda actualizarse con nuevo correo. Por ultimo, el servidor termina la conexión.

3.6 COMANDOS POP OPCIONALES

Los comandos que se han mencionados, son los comandos básicos necesarios, pero hay otros comandos que proporcionan una mayor flexibilidad en el cliente sin complicar en exceso el protocolo.
APOP (entrar en el sistema con contraseña encriptada) es una alternativa a los comandos USER y PASS. El comando APOP necesita de dos parámetros, uno es un identificador de cuenta y el otro es la clave encriptada. Al conectarse al servidor POP, éste envía una marca de tiempo. Junto con esta marca de tiempo, se aplica un algoritmo para encriptar la contraseña. Este algoritmo se encuentra definido en el
RFC 1321. Este método de autentificación POP es útil para aquellos usuarios que se conectan frecuentemente a sus servidores de correo, evitando de esta manera, que la clave de la cuenta viaje frecuentemente por la red sin encriptar. La sintaxis de este comando es la siguiente:

APOP id_cuenta clave_encriptada#13#10

TOP permite al cliente de correo recuperar la parte del encabezado del mensaje y un número de líneas del cuerpo o núcleo del mensaje. Este comando se suele utilizar cuando se desea conocer los mensajes sin leerlos. La sintaxis de este comando es:

TOP numero_mensaje numero_lineas_del_cuerpo#13#10

UIDL (lista de identificadores únicos) permite obtener del servidor una identificación (ID) de mensaje única y persistente para uno o todos los mensajes de la casilla. El servidor genera un ID de mensaje que se debe conservar entre las distintas sesiones. De esta forma, el cliente puede realizar un seguimiento de que mensajes ya se han recuperado y cuales son nuevos. La respuesta del servidor a este comando es una línea con el numero de mensaje y el identificador único si se refiere a un mensaje. Si el comando se refiere a todos los mensajes, el servidor devuelve una línea por mensaje. Al final de las líneas mencionadas aparece un punto seguido de los caracteres de retroceso de carro/avance de línea. La sintaxis de este comando es:

UIDL [numero_mensaje]#13#10

3.7 INSTALACIÓN DE POP3

Existen diferentes programas que ofrecen el servicio de POP, así que escogimos uno sencillo de instalar. Ipopd es el programa en cuestión. Tras investigar un poco, descubrimos que existía una versión segura de este software. Cuando nos conectamos a un servidor POP, necesitamos dar nuestra contraseña. Esta misma se envía normalmente como texto plano, es decir, que cualquier persona que estuviera "escuchando" nuestra línea, podría descubrir fácilmente nuestra clave de acceso. Por eso constantemente se desarrollan versiones seguras del software, que se encargan de cifrar la contraseña de tal manera que sólo el destinatario pueda descifrarla. Así hacemos seguro nuestro password. También podemos decir que esta versión segura soporta también conexiones no seguras, es decir son dos.

La versión segura de ipopd se llama ipopd-ssl, así que vayamos a instalarla:

apt-get install ipopd-ssl

Después de haber realizado esto, tenemos que abrir el puerto 110 del router y redirigirlo al 110 de nuestro servidor. Tendremos instalado nuestro servidor de POP3, y todos los usuarios del sistema gozarán de una cuenta de email para poder enviar y recibir su correo electrónico desde sus casas.

3.8 INSTALACIÓN DE QPOPPER

QPOPPER es FREEWARE, está producido por QUALCOMM. Es probable que su distribución de Linux venga con alguna versión de servidor POP3. Necesita actualizar el demonio POP3, se puede instalar para el dominio QPOPPER. Se usa en servidores de producción para un año y no se puede atribuir ninguna del servidor. Se descarga en un directorio con el espacio libre para compilarlo. Por ejemplo descargado en QPOPPER en usr/local/src, se utiliza el siguiente comando:

[root@ford src] # uncompress qpopper2.53.tar.z

Cuando ya se ha descomprimido el archive (qpopper2.53.tar), se extraen los archivos individuales del archivo tar, de esta manera:

[root@ford src] # tar –xf qpopper2.53.tar

Esto crea un subdirectorio llamado qpopper2.53, en el cual encontrara todos los archivos fuente y documentación que viene con el paquete del programa.

3.9 COMPILACIÓN QPOPPER

Viene con un strip configure que realiza esto. Todo lo que se necesita es darle alguna dirección adicional y el resto lo hará el script. Se puede todas las opciones de configuración, y se puede ejecutar el comando *configure* con el parámetro –help de esta manera:

[root@ford qpopper2.53] # ./configure –help

Daremos un ejemplo de línea de configuración para activar el Authenticated POP, POP Autenticado (APOP) que proporcione soporte para contraseñas ocultas y establezca el modo del servidor.

[root@ford qpopper2.53] # ./configure –enable-apop=/etc/pop.auth –with-Popuid=bin –enables-specialauth –enable-servermode

Cuando ejecuta el script de configuración, mostrara toda la información que encuentre. Aunque la opción del instalar los binarios en cualquier sitio, es sirve para situarlo en el directorio /usr/bin puesto que esta en el resto de las aplicaciones. Como los comandos para copia popper y popauth en usr/bin:

[root@ford qpopper2.53] # cp popper /usr/bin
[root@ford qpopper2.53] # cp popauth / usr/bin
[root@ford qpopper2.53] # chmod 755 /usr/bin/popper
[root@ford qpopper2.53] # chmod 4755 /usr/bin/popauth
[root@ford qpopper2.53] # chown root.root /usr/bin/popper
[root@ford qpopper2.53] # chown bin.root usr/bin/popauth

Para instalar el directorio en .8 /usr/man/man8 de esta forma:

[root@ford qpopper2.53] # cp .8 /usr/man/man8
[root@ford qpopper2.53] # chown root.root /usr/man/man8/popper.8
[root@ford qpopper2.53] # chown root.root /usr/man/man8/popauth.8
[root@ford qpopper2.53] # chmod 644 /usr/man/man8/popper
[root@ford qpopper2.53] # chmod 644 /usr/man/man8/popauth

CONFIGURACIÓN QPOPPER

El servicio QPOPPER se ejecuta del demonio inetd, esta es la manera de configurarlo así:

Borrar otras entradas de pop: El primer paso es editar el archivo /etc/inetd.conf borrando las entradas existentes de otros servicios POP que pudiera haber. Para eliminar la entrada existente se busca el nombre del servicio que corresponda al servidor POP listado en el archivo /etc/services. Una forma más rápida de encontrar la información es mediante un uso de la utilidad **grep** y especificar la cadena de búsqueda. Por ejemplo:

[root@ford /root] # grep '110/tcp' /etc/services
Si el nombre del servicio esta definido, se definirá así:

Pop-3 110/tcp # pop versión 3

Si el comando **grep** con su cadena de búsqueda no genera salida, entonces necesitara editar el archivo /etc/services e insertar la línea:

Pop-3 110/tcp # pop versión 3

Configurar QPOPPER como servicio POP3: se puede saber e l nombre del servicio esta definido el archivo /etc/inetd.conf que el programa necesita. Después de que designe como el programa del servidor POP3, se coloca la siguiente línea:

Pop-3 stream tcp nowait root /usr/bin/popper popper

Si se tiene TCPWAPPERS para hacer log (/usr/sbin/tcpd o /usr/local/sbin/tcpd). Se puede usar de esta manera:

Pop-3 stream tcp nowait root /usr/sbin/tcpd /usr/bin/popper

Para dar un demonio en inetd, siempre y cuando el sistema guarde ID de procesos (PID) se usaría de esta manera:

[root@ford /root] # kill -1 'cat /var/run/inetd.pid'

Y sino usa el método de PID se manejaría de esta forma:

[root@ford /root] # ps –no-heading –c inetd I awk '{print $1}'

Y mostraría un número como salida. Así que se llamaría el número ***pid***, se introduce la sentencia:

[root@ford /root] # kill -1 pid

OPCIONES DE COMANDOS PARA QPOPPER

Una de las características de QOPPOERE son los comandos, para esto se edita el archivo /etc/inetd.conf de forma que esas opciones aparezcan al final de cada línea. Y dando el demonio inetd una señal de comando **kill -1.** Astas son las opciones de QPOPPER:

OPCIÓN	DESCRIPCIÓN
-b bulldir	En lugar de usar el valor de la opción **--enable-bulletin** de ./configure para designar el directorio del boletín, se configura con el valor de bulldir.
-T timeout	configura el timeout de la conexión a timeout a segundos en lugar del valor por defecto de 600 segundos. Los valores

recomendados van de 30 a 120 segundos.

Opciones de comando *configure*:

Opción de configuración	Descripción
• --enable-apoppath • --with-popuid=usuario	Estas dos opciones se usan a la vez. La opción --enable-apop=path activa el servicio APOP, el cual es una extensión del POP3 normal. La extensión APOP permite a los usuarios enviar sus contraseñas en un formato encriptado en lugar de en texto. El parámetro *path* especifica el archivo donde reside el archivo de autenticación /etc/pop.auth. El parámetro usuario especifica el nombre de usuario que será el propietario del archivo. El usuario *bin* es una buena elección, que es la configuración de la base de datos de APOP.
• --enable-bulletind=path	Bulletins son un mecanismo útil para hacer anuncios a todos los usuarios sin tener que mandar a todos los buzones el mismo mensaje. Solo se escribe un mensaje, y cada usuario y cada usuario lo vera como parte de su correo. Ahorraría espacio en el disco. El *path* es el directorio donde se envía los boletines nuevos.
• -enable-servermode	El modo del servidor se designa para hacer que el demonio QPOPPER se ejecute más efectivo en entornos de carga alta. Se considera usar esta opción si tiene un gran número de usuarios activos.
• --enable-specialauth	Esta opción se necesita si usa contraseñas ocultas.

Tabla 2. Comandos configure.

Se puedes conectarte manualmente al servidor POP3 haciendo Telnet al puerto 110. Es muy útil cuando te envían un mensaje con un fichero muy largo que no quieres recibir.

- USER <nombre> Identificación de usuario (Solo se realiza una vez).
- PASS <password> Envías la clave del servidor.
- STAT Da el número de mensajes no borrados en el buzón y su longitud total.
- LIST Muestra todo los mensajes no borrados con su longitud.

- RETR <número> Solicita el envío del mensaje especificando el número (no se borra del buzón).
- TOP <número> <líneas> Muestra la cabecera y el número de líneas requerido del mensaje especificando el número.
- DELE <número> Borra el mensaje especificando el número.
- RSET Recupera los mensajes borrados (en la conexión actual).
- QUIT Salir.

4. SERVIDOR SMTP

El protocolo SMTP significa simple de transferencia de correo electrónico (Simple Mail Transfer Protocol) es uno de los servicios más importantes que proporciona Internet. Actualmente la mayoría de las compañías tienen o dependen del correo para manejar su información interna de la compañía. Se encuentran disponibles muchos paquetes SMTP como por ejemplo SENDMAIL que es uno de los mas viejos y mas probados. Pero hay dos nuevos paquetes POSTFIX y QMAIL ambos escritos desde ceros teniendo la seguridad. Se ejecuta en el puerto 25, TCP.

```
ipfwadm –I –a accept –P tcp –S 10.0.0.0/8 –D 0.0.0.0/0 25
ipfwadm –I –a accept –P tcp –S un.host.fiable –D 0.0.0.0/0 25
ipfwadm –I –a deny –P tcp –S 0.0.0.0/0 –D 0.0.0.0/0 25
```

O también se podría utilizar:

```
ipchains –A input –p tcp –j ACCEPT –s 10.0.0.0/8 –d 0.0.0.0/0 25
ipchains –A input –p tcp –j ACCEPT –s un.host.fiable –d 0.0.0.0/0 25
ipchains –A input –p tcp –j DENY –s 0.0.0.0/0 –d 0.0.0.0/0 25
```

41. FUNCIONAMIENTO DEL PROTOCOLO SMTP

Cuando un cliente establece una conexión con el servidor SMTP, espera a que éste envíe un mensaje "220 Service ready" o "421 Service non available"

- Se envía un HELLO desde el cliente. Con ello el servidor se identifica. Esto puede usarse para comprobar si se conectó con el servidor SMTP correcto.

- El cliente comienza la transacción del correo con la orden MAIL. Como argumento de esta orden se puede pasar la dirección de correo al que el servidor notificará cualquier fallo en el envío del correo. El servidor responde "250 OK".

- Ya le hemos dicho al servidor que queremos mandar un correo, ahora hay que comunicarle a quien. La orden para esto es RCPT TO:<destino@host>. Se pueden mandar tantas órdenes RCPT como destinatarios del correo queramos. Por cada destinatario, el servidor contestará "250 OK" o bien "550 No such user here", si no encuentra al destinatario.

- Una vez enviados todos los RCPT, el cliente envía una orden DATA para indicar que a continuación se envían los contenidos del mensaje. El servidor responde "354 Start mail input, end with <CRLF>.<CRLF>" Esto indica al cliente como ha de notificar el fin del mensaje.

- Ahora el cliente envía el cuerpo del mensaje, línea a línea. Una vez

finalizado, se termina con un <CRLF>.<CRLF> (la última línea será un punto), a lo que el servidor contestará "250 OK", o un mensaje de error apropiado.

- Tras el envío, el cliente, si no tiene que enviar más correos, con la orden QUIT corta la conexión. También puede usar la orden TURN, con lo que el cliente pasa a ser el servidor, y el servidor se convierte en cliente. Finalmente, si tiene más mensajes que enviar, repite el proceso hasta completarlos.

Puede que el servidor SMTP soporte las extensiones definidas en el RFC 1651, en este caso, la orden HELO puede ser sustituida por la orden EHLO, con lo que el servidor contestará con una lista de las extensiones admitidas. Si el servidor no soporta las extensiones, contestará con un mensaje "500 Syntax error, command unrecognized".

De los tres dígitos del código numérico, el primero indica la categoría de la respuesta, estando definidas las siguientes categorías:

- 2XX, la operación solicitada mediante el comando anterior ha sido concluida con éxito
- 3XX, la orden ha sido aceptada, pero el servidor esta pendiente de que el cliente le envíe nuevos datos para terminar la operación
- 4XX, para una respuesta de error, pero se espera a que se repita la instrucción
- 5XX, para indicar una condición de error permanente, por lo que no debe repetirse la orden

Una vez que el servidor recibe el mensaje finalizado con un punto puede bien almacenarlo si es para un destinatario que pertenece a su dominio, o bien retransmitirlo a otro servidor para que finalmente llegue a un servidor del dominio del receptor.

FORMATO DEL MENSAJE

El mensaje es enviado por el cliente después de que envié la orden DATA al servidor. El mensaje está compuesto por dos partes:

- Cabecera: en el ejemplo las tres primeras líneas del mensaje son la cabecera. En ellas se usan unas palabras clave para definir los campos del mensaje. Estos campos ayudan a los clientes de correo a organizarlos y mostrarlos. Los más típicos son *subject* (asunto), *from* (emisor) y *to* (receptor). Estos dos últimos campos no hay que confundirlos con las órdenes MAIL FROM y RCPT TO, que pertenecen al protocolo, pero no al formato del mensaje.
- Cuerpo del mensaje: es el mensaje propiamente dicho. En el SMTP básico está compuesto únicamente por texto, y finalizado con una línea en la que el único carácter es un punto.

4.2 COMANDOS SMTP

HELO	Lo envía un cliente para identificarse a sí mismo, normalmente con un nombre de dominio.
EHLO	Permite al servidor identificar su compatibilidad con los comandos del Protocolo simple de transferencia de correo extendido (ESMTP).
MAIL FROM	Identifica al remitente del mensaje; se utiliza con el formato MAIL FROM:.
RCPT TO	Identifica a los destinatarios del mensaje; se utiliza con el formato RCPT TO:.
TURN	Permite que el cliente y el servidor intercambien las funciones, y envíen correo en la dirección contraria sin tener que establecer una conexión nueva.
ATRN	El comando ATRN (TURN autenticado) toma uno o más dominios como parámetro de forma opcional. El comando ATRN debe rechazarse si la sesión no se ha autenticado.
SIZE	Proporciona un mecanismo por el cual el servidor SMTP puede indicar el tamaño máximo del mensaje aceptado. Los servidores compatibles deben proporcionar extensiones de tamaño para indicar el tamaño máximo de mensaje que pueden aceptar. Los clientes no deben enviar mensajes mayores que el tamaño indicado por el servidor.
ETRN	Una extensión de SMTP. ETRN lo envía un servidor SMTP para solicitar que otro servidor envíe todos los mensajes de correo electrónico que tenga.
PIPELINING	Permite enviar una secuencia de comandos sin esperar una respuesta de cada comando.
CHUNKING	Un comando ESMTP que reemplaza al comando DATA. Como el host SMTP no tiene que buscar continuamente el fin de los datos, este comando envía un comando BDAT con un argumento que contiene el número total de bytes de un mensaje. El servidor de recepción cuenta los bytes del mensaje y, cuando el tamaño del mensaje es igual que el valor enviado por el comando BDAT, supone que ha recibido todos los datos del mensaje.
DATA	Lo envía un cliente para iniciar la transferencia del contenido del mensaje.
DSN	Un comando ESMTP que permite la entrega de notificaciones de estado.
RSET	Anula toda la transacción del mensaje y restablece el búfer.

TABLA 3. Comandos SMTP

4.3 PUERTOS UTILIZADOS POR EXCHANGE

SMTP	TCP: 25	El servicio SMTP utiliza el puerto TCP 25.
DNS	TCP/UDP: 53	DNS escucha en el puerto 53. Los controladores de dominio utilizan este puerto.
LSA	TCP: 691	El servicio Motor de enrutamiento de Microsoft Exchange (RESvc) escucha la información de estado de los vínculos de enrutamiento en este puerto.
LDAP	TCP/UPD: 389	El Protocolo ligero de acceso a directorios (LDAP) utilizado por el servicio de directorio Microsoft Active Directory®, el Conector de Active Directory y el directorio de Microsoft Exchange Server 5.5 utilizan este puerto.
LDAP/SSL	TCP/UDP: 636	LDAP sobre Secure Sockets Layer (SSL) utiliza este puerto.
LDAP	TCP/UDP: 379	El Servicio de replicación de sitios (SRS) utiliza este puerto.
LDAP	TCP/UDP: 390	Éste es el puerto alternativo recomendado para configurar el protocolo LDAP de Exchange Server 5.5 cuando Exchange Server 5.5 está ejecutándose en un controlador de dominio de Active Directory.
LDAP	TCP: 3268	Catálogo global. El catálogo global de Active Directory (una "función" de controlador de dominio) de Windows 2000 y Windows Server 2003 escucha en el puerto TCP 3268.
LDAP/SSLPort	TCP: 3269	Catálogo global sobre SSL. Las aplicaciones que se conectan al puerto TCP 3269 de un servidor de catálogo global pueden transmitir y recibir datos cifrados mediante SSL.
IMAP4	TCP: 143	El Protocolo de acceso a correo de Internet (IMAP) utiliza este puerto.
IMAP4/SSL	TCP: 993	IMAP4 sobre SSL utiliza este puerto.
POP3	TCP: 110	El Protocolo de oficina de correo versión 3 (POP3) utiliza este puerto.
POP3/SSL	TCP: 995	POP3 sobre SSL utiliza este puerto.
NNTP	TCP: 119	El Protocolo de transferencia de noticias a través de la red (NNTP) utiliza este puerto.
NNTP/SSL	TCP: 563	NNTP sobre SSL utiliza este puerto.
HTTP	TCP: 80	HTTP utiliza este puerto.
HTTP/SSL	TCP: 443	HTTP sobre SSL utiliza este puerto.

TABLA 4. Puertos por Exchange

4.3 PAQUETE SMPT

SENDMAIL

SENDMAIL es uno de los servicios que se manejaban con mayor frecuencia en las compañías para el protocolo SMTP. Hacer un CHROOT del SENDMAIL es una buena opción, se ejecuta como Root. SENDMAIL sólo tiene que estar accesible para que pueda ser utilizado y reciba correo de otras máquinas y repartirlo localmente, o solo para que funcione como reparto local y que se pueda enviar con facilidad el correo a otras máquinas. Simplemente filtrar con el cortafuegos a el SENDMAIL, o ejecutarlo en modo demonio.

Para configurar la ejecución del SENDMAIL en modo cola: Se edita el Script de inicio del SENDMAIL y cambia la línea que contiene:

sendmail –bd –q1h
por:
sendmail –q1h

Si se utiliza el sistema para enviar mucho correo posiblemente se disminuye el tiempo de refresco. Ahora el correo saliente y el correo interno del sistema se comportarán bien, lo cual a menos que se ejecute un servidor de correo. Los ficheros de configuración del SENDMAIL consisten en aplicar al SENDMAIL 8.9.x. Como por ejemplo:

/etc/sendmail.cf: El fichero de configuración principal, también dice dónde se encuentran el resto de ficheros de configuración.

/etc/mail/: Se puede definir la localización de los ficheros de configuración en sendmail.cf, generalmente se coloca en /etc/ o en /etc/mail.

Access: La base de datos de la lista de accesos, permite rechazar el correo proveniente de ciertas fuentes (IP o dominio), y controlar con facilidad las transmisiones. Por ejemplo:

RELAY
spam.com REJECT

Lo que quiere decir que a 10.0.0.* los Hosts de mi red interna se les permite utilizar el servidor de correo para enviar correo pero *.spam.com se rechaza. Hay listas en línea de Spammers conocidos, generalmente suele tener entre 5-10.000 entradas, lo cual puede afectar seriamente el rendimiento del SENDMAIL, pues cada conexión se comprueba contra esta lista, y por otro lado al tener otra maquina SENDMAIL podría enviar Spam pero podría ser peor.

POSTFIX

El POSTFIX es un agente de transferencia de correo (MTA) orientado a la seguridad, velocidad, y facilidad de configuración, cosas en las que SENDMAIL suele fallar por lo general. POSTFIX que se ejecuta como Root es un programa de control maestro, llamado "master", que llama a otros programas para procesar el correo a la cola ("Pickup"), un programa para gestionar la cola, espera conexiones entrantes, repartos de correo retrasados, ("qmgr"), un programa que en realidad envía y recibe el correo ("SMTPD"), Cada parte de POSTFIX está muy bien pensada, y generalmente hace una o dos tareas.

Por ejemplo, en lugar del modelo de SENDMAIL, donde el correo simplemente se volcaba a /var/spool/mqueue, en POSTFIX existe un directorio accesible llamado "maildrop" el cual se comprueba mediante "Pickup", que alimenta los datos a "cleanup", y mueve el correo a un directorio seguro de cola para el procesado real.

Los ficheros primarios de configuración están en /etc/postfix, y existen varios ficheros primarios de configuración que es necesario tener:

master.cf: Controla el comportamiento de varios programas de "ayuda", están hechos Chroot, el máximo número de procesos que pueden ejecutar. Probablemente sea mejor dejar las configuraciones por defecto en la mayoría de los servidores de correo, a menos que se necesite ajustar algo por altas cargas o asegurar el servidor.

main.cf: Este fichero está muy próximo al sendmail.cf, en cuanto a propósito y en cuanto al diseño es bastante diferente. Está bien comentado y configura todas las variables principales, las situaciones y formato de diferentes ficheros que contienen información tal como los mapeos a usuarios virtuales e información relativa.

Aquí se encuentra una lista de variables y localización de ficheros que se suele tener que configurar, el fichero /etc/postfix/main.cf por lo general suele estar comentado. Daremos unos ejemplos de entradas main.cf pero teniendo en cuenta que no son un main.cf completo.

```
# ¿Cuál es el nombre de la máquina?
myhostname = correo.ejemplo.org

# ¿Cuál es el nombre de dominio?
mydomain = ejemplo.org

# ¿Cómo etiqueta el "from" del correo?
myorigin = $mydomain

# ¿En qué interfaces lo ejecuto? En todas
inet_interfaces = all
```

un fichero que contiene una lista de nombres de hosts y nombres de

dominios cualificados desde los cuales recibo correo.
# Habitualmente están listados así:
mydestination = localhost, $myhostname.

# Pero prefiero mantener el listado en un fichero.
mydestination = /etc/postfix/mydestination
# Mapa de nombres de usuarios entrantes. "man 5 virtual"
virtual_maps = hash:/etc/postfix/virtual

# Mapeo de alias (como /etc/aliases en sendmail), "man 5 alias"
alias_maps = hash:/etc/postfix/aliases

# Base de datos de alias, se pueden tener diferentes configuraciones. "man 5 alias"
alias_database = hash:/etc/postfix/aliases

# ¿Dónde repartir el correo, formato Mailbox o Maildir (el tradicional /var/spool/mail)?
home_mailbox = Maildir/

# ¿Dónde guardar el correo, generalmente en /var/spool/mail/ pero se puede cambiar con facilidad?
mail_spool_directory = /var/spool/mail

# ¿Qué comando utilizamos para repartir el correo? /usr/bin/procmail es el comando por defecto, pero yo utilizo scanmail que es el sello del software antivirus AMaViS.
mailbox_command = /usr/sbin/scanmails

# ¿Para quién retransmito el correo, se pueden listar o guardarlos en un fichero (uno por línea)?
relay_domains = /etc/postfix/relaydomains

# Lista de redes locales (por defecto se retransmite el correo de estos hosts).
mynetworks = 10.0.0.0/24, 127.0.0.0/8

# ¿Qué se le muestra a la gente que conecte al puerto 25? Por defecto muestra el número de versión.
smtpd_banner = $myhostname ESMTP $mail_name

En cualquier fichero que simplemente liste un elemento por línea como /etc/postfix/mydestination o /etc/postfix/relaydomains, se suelen almacenar como simple texto llano. Los ficheros que contienen mapeados como por ejemplo cuando se maneja un alias donde se tienen entradas como Root: Cualquier Usuario que en este caso se deberían transformar en ficheros Hash de base de datos por velocidad y se especifica el tipo de fichero como Hash, dbm. Una de las ventajas de Postfix es que cuenta con una licencia y tiene el código abierto y libre.

5. SERVIDOR TELNET

Telnet es un protocolo estándar siendo su número STD de 8. Su status es recomendado. Se describe en el RFC 854 - Especificaciones del protocolo Telnet y RFC 855 - "Telnet Option Specifications". El protocolo Telnet proporciona una interfaz estandarizada, a través de la cual un programa de un Host (el cliente de Telnet) puede acceder a los recursos de otro Host (el servidor de Telnet) como si el cliente fuera una Terminal local conectada al servidor.

5.1 FUNCIONAMIENTO DEL TELNET

TELNET es un protocolo basado en tres ideas:

- El concepto de NVT (Network Virtual Terminal) (NVT). Una NVT es un dispositivo imaginario que posee una estructura básica común a una amplia gama de terminales reales. Cada Host mapea las características de su propia Terminal sobre las de su correspondiente NVT, y asume todos los demás Hosts harán lo mismo.
- Una perspectiva simétrica de las terminales y los procesos.
- El protocolo TELNET usa el principio de opciones negociadas, ya que muchos Host pueden desear suministrar servicios adicionales. El cliente y el servidor utilizan una serie de convenciones para establecer las características operacionales de su conexión TELNET a través de los mecanismos Do y Don't.

5.2 ESTRUCTURA DE COMNADOS EN TELNET

La comunicación entre cliente y servidor es manejada por comandos internos, que no son accesibles a los usuarios. Todos los comandos internos de TELNET consisten en secuencias de 2 o 3 bytes, dependiendo del tipo de comando. El carácter IAC ("Interpret As Command"; Interpretar Como Comando) es seguido de un código de comando. Si este comando trata con opciones de negociación, el comando tendrá un tercer byte para mostrar el codigo asociado a la opción indicada.

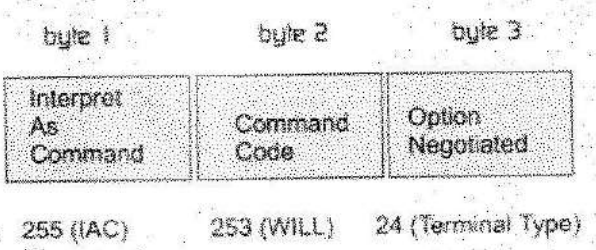

FIGURA 6. Estructura Telnet

5.3 PROBLEMAS MÁS FRECUENTES EN TELNET

- Tipo de la configuración del terminal, cada computadora acepta que las terminales que se conectan a ella sean de algún tipo determinado (normalmente VT100 o VT200) y si nuestro software de Telnet no es capaz de emular estos tipos de terminales lo suficientemente bien, pueden aparecer caracteres extraños en la pantalla o que no consigamos escribir con nuestro teclado un determinado carácter.

- Autentificación en texto claro, nombre de usuario y contraseña.

- Ataques a contraseñas.

La mejor solución para estos problemas es desactivar el Telnet y utilizar SSH. Sin embargo esto no es práctico en todas las situaciones. Si es necesario utilizar Telnet, sugeriría filtrarlo mediante un cortafuegos, , tener reglas para permitir a los Hosts/redes acceso a puerto 23, y después tener una regla general denegando acceso al puerto 23, al igual que utilizar TCP_WRAPPERS. El cual comprueba cada conexión de Telnet y no cada paquete contra las reglas del cortafuegos. Sin embargo utilizar TCP_WRAPPERS les permitirá a los usuarios dar por hecho que se está ejecutando Telnet si les permite conectar, se evalúa la conexión y después se cierra si no se está listo como permitido el acceso.

Daremos un breve ejemplo de reglas de Cortafuegos:

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 23
ipfwadm -I -a accept -P tcp -S un.host.fiable -d 0.0.0.0/0 23
ipchains -A input -p all -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 23
```

Ahora daremos un ejemplo utilizando TCP_WRAPPERS:

En /etc/hosts.allow
in.telnetd: 10.0.0.0/255.0.0.0, un.host.fiable
Y en /etc/hosts.deny
in.telnetd: ALL

Existen varias alternativas cifradas al Telnet, como SSH, SSLEAY Telnet y otras utilidades de terceros.

Para asegurar las cuentas de los usuarios con respecto a Telnet, se pueden hacer varias cosas. La primera sería no permitir al Root hacer login vía Telnet, lo cual se controla mediante el /etc/securetty y por defecto en la mayoría de las distribuciones el Root tiene restringido el acceso a la consola. Para que un usuario haga login con éxito, su shell tiene que ser válido: El cual viene determinado por la lista de shells de /etc/shells, de modo que configurar cuentas de usuario a las que se les permita hacer login es simplemente cuestión de configurar su shell a alguno de los listados en /etc/shells.

5.4 COMANDOS TELNET

Telnet establece una representación estándar para algunas funciones:

- IP: Interrumpir proceso
- AO: Abortar la salida
- AYT: Se encuentra
- EC: Borrar carácter
- EL: Borrar línea
- SYNCH: Sincronizar

5.4 SHELL

Este protocolo fue diseñado para dar seguridad al acceso a computadores en forma remota. Cumple la misma función que Telnet o rlogin pero además, usando criptografía, logra seguridad con los datos. A diferencia de telnet u otro servicio similar, SSH utiliza el puerto 22 para la comunicación y la forma de efectuar su trabajo es muy similar al efectuado por SSL. Para su uso se requiere que por parte del servidor exista un demonio que mantenga continuamente en el puerto 22 el servicio de comunicación segura, el sshd. El cliente debe ser un software tipo TeraTerm o Putty que permita la hacer pedidos a este puerto 22 de forma cifrada.

La forma en que se entabla una comunicación es en base la misma para todos los protocolos seguros:

- El cliente envía una señal al servidor pidiéndole comunicación por el puerto 22.
- El servidor acepta la comunicación en el caso de poder mantenerla bajo encriptación mediante un algoritmo definido y le envía la llave pública al cliente para que pueda descifrar los mensajes.
- El cliente recibe la llave teniendo la posibilidad de guardar la llave para futuras comunicaciones o destruirla después de la sesión actual.

Permite definir variables en un fichero de comandos en la forma:

USER=/mnt/mecan/juanto
TERM=hp2392
...

Es una práctica habitual el utilizar nombres con letras mayúsculas para estas variables. Para recuperar el valor de una variable hay que precederla con el carácter. Por ejemplo, utilizando en otra parte del programa TERM, en dicho lugar se sustituiría TERM por su valor, esto es, hp2392. El shell del Linux las tiene definidas para cada usuario unas variables estándar. Para saber cuáles son se digita el comando siguiente:

Set

Sirve para definir otras variables propias de cada usuario puede utilizarse el

fichero .profile, que es un fichero de comandos propio de cada usuario que se ejecuta automáticamente al hacer el login. Para definir variables que contengan espacios en blanco deben encerrarse entre caracteres (') o ("), como por ejemplo:

FECHA="31 de Diciembre de 1986" más adelante en la que se vera la diferencia entre el carácter (') y el carácter (").

Para un PSI que quiere permitir a sus clientes cambiar sus contraseñas con facilidad, pero no permitirles acceso al sistema. PSI utiliza Ultrasparcs y por alguna razón se niega a distribuir cuentas de usuario en /etc/shells se lista:

/usr/bin/passwd

Y se cambia el shell de los usuarios por /usr/bin/passwd, de modo que se tiene algo así:

nombreusuario:x:1000:1000::/home/nombreusuario:/usr/bin/passwd

El usuario hace un Telnet al servidor, se le pregunta su nombre de usuario y contraseña, y después se le pide cambiar la contraseña. Si se hace correctamente, password termina y se les desconecta. Lo que sigue es una transcripción de tal configuración cuando un usuario hace telnet:

Trying 1.2.3.4...
Connected to localhost
Escape character is '^]'.
Fedora Linux release 5.2 (Apollo)
Kernel 2.2.5 on an i586
login: tester
Password:
Changing password for tester
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
password: all authentication tokens updated successfully
Connection closed by foreign host.

Telnet también muestra un banner por defecto cuando se conecta alguien. El banner suele contener información del sistema, como el nombre, el SO, la versión y a veces otro tipo de información detallada, como la versión del kernel. Telnet muestra los contenidos del fichero /etc/issue.net. Este fichero se suele volver a crear al arrancar, en la mayoría de las distribuciones de Linux, desde el fichero de arranque rc.local. Edita el fichero rc.local, ya sea modificando lo que pone en /etc/issue y /etc/issue.net, comentando las líneas que crean esos ficheros, y después editando los ficheros con información estática.

Los contenidos de un fichero rc.local típico pertenecientes a /etc/issue y /etc/issue.net:

```
# This will overwrite /etc/issue at every boot. So make any changes
# you want to make to /etc/issue here or you will lose them when you
# reboot.
echo "" > /etc/issue
echo "$R" >> /etc/issue
echo "Kernel $(uname –r) on $a $(uname –m)" >> /etc/issue
cp –f /etc/issue /etc/issue.net
echo >> /etc/issue
```

Comenta las líneas o elimina los comandos uname, si es absolutamente necesario habilitar el Telnet para hacer logins de usuarios, se asegura de mostrar una advertencia. Este sistema es exclusivamente para usos autorizados. Si algún usuario intenta abusar de este servicio legalmente las empresas tienen una posición más fuerte al sistema y le harían restricciones al sistema.

5.5 SEGURIDAD

Hay tres razones principales por las que el Telnet es una mala opción para los sistemas modernos desde el punto de vista de la seguridad de la computadora:

- Los demonios de uso general del Telnet tienen varias vulnerabilidades descubiertas sobre los años.
- El Telnet, por defecto, no cifra ningunos de los datos enviados sobre la conexión (incluyendo las contraseñas), así que mejor asegurar las contraseñas utilizar la contraseña más adelante para los propósitos de que no descubran los passwords de los usuarios.
- El Telnet carece un esquema de la autentificación que permitir asegurar que la comunicación esté realizada entre los las personas encargadas de esta seguridad.

Por ejemplo en el Internet público el Telnet no debe ser utilizado. Las sesiones del Telnet son unencrypted. Esto significa que cualquiera que tiene acceso a cualquier rebajadora, el interruptor, o la entrada donde se localiza la red si dos usuarios están utilizando el Telnet puede interceptar los paquetes del Telnet que pasan cerca y obtener fácilmente la información de la conexión y de la contraseña, en las que habrían varias utilidades como tcpdump y Wireshark.

Además Telnet no se debe utilizar generalmente siempre en redes con conectividad del Internet.

5.6 PROTOCOLO SSL

El Protocolo SSL (Secure Socket Layer) fue desarrollado por Netscape para permitir confidencialidad y autenticación en Internet. SSL es una capa por debajo de HTTP y tal como lo indica su nombre esta a nivel de Socket por lo que permite ser usado no tan solo para proteger documentos de hipertexto sino también servicios como FTP, SMTP, Telnet entre otros. La idea que persigue SSL es encriptar la comunicación entre servidor y cliente mediante el uso de llaves y algoritmos de encriptación. El protocolo TLS esta basado en SSL y son similares en el modo de operar. Es importante señalar que ambos protocolos se ejecutan sobre una capa de transporte definida, pero no determinada. Esto indica que pueden ser utilizados para cualquier tipo de comunicaciones. La capa de transporte más usada es TCP cobre la cual pueden implementar seguridad en HTTP.

5.7 ¿COMO FUNCIONA?

Se solicita una comunicación segura, servidor y el cliente se deben poner de acuerdo en como se comunicaran (SSL Handshake) para luego comenzar la comunicación encriptada. Luego de terminada la transacción, SSL termina.

Solicitud de SSL:

Típicamente este proceso ocurre en el momento que un cliente accede a un servidor seguro, identificado con "https://...", pero como se mencionó, no necesariamente es usado para HTTP. La comunicación se establecerá por un puerto distinto al utilizado por el servicio normalmente. Luego de esta petición, se procede al SSL Handshake.

SSL Handshake:

En este momento, servidor y cliente se ponen de acuerdo en varios parámetros de la comunicación. Se puede dividir el proceso en distintos pasos:

- Client Hello: El cliente se presenta. Le pide al servidor que se presente (certifique quien es)y le comunica que algoritmos de encriptación soporta y le envía un número aleatorio para el caso que el servidor no pueda certificar su validez y que aun así se pueda realizar la comunicación segura.
- Server Hello: El servidor se presenta. Le responde al cliente con su identificador digital encriptado, su llave pública, el algoritmo que se usará, y otro número aleatorio. El algoritmo usado será el más poderoso que soporte tanto el servidor como el cliente.
- Aceptación del cliente: El cliente recibe el identificador digital del servidor, lo desencripta usando la llave pública también recibida y verifica que dicha identificación proviene de una empresa certificadora segura. Luego se procede a realizar verificaciones del certificado (identificador) por medio de fechas, URL del servidor, etc. Finalmente el cliente genera una llave aleatoria usando la llave pública del servidor y el

algoritmo seleccionado y se la envía al servidor.

- Verificación: Ahora tanto el cliente y el servidor conocen la llave aleatoria (El cliente la generó y el servidor la recibió y desencriptó con su llave privada). Para asegurar que nada ha cambiado, ambas partes se envían las llaves. Si coinciden, el Handshake concluye y comienza la transacción.

Intercambio de Datos:

Desde este momento los mensajes son encriptados con la llave conocida por el servidor y el cliente y luego son enviados para que en el otro extremo sean desencriptados y leídos.

Terminación de SSL

Cuando el cliente abandona el servidor, se le informa que terminara la sesión segura para luego terminar con SSL.

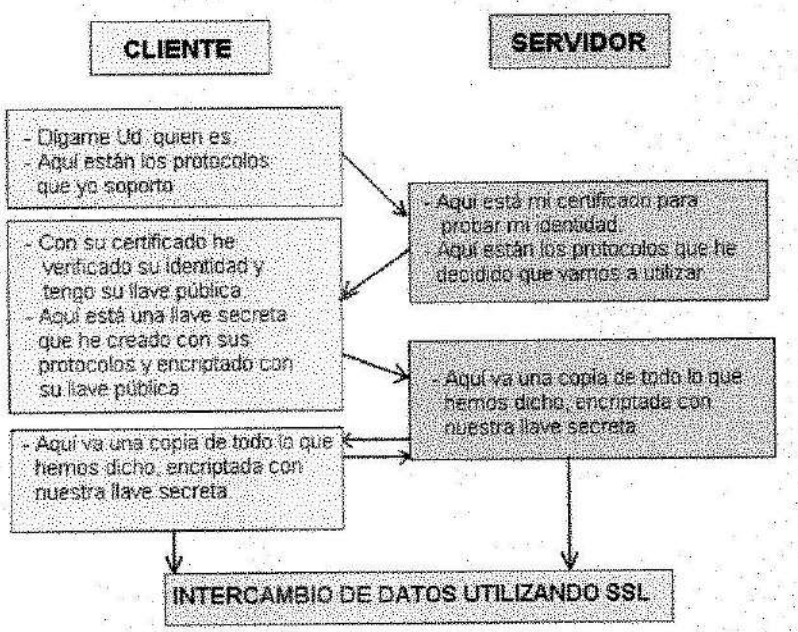

FIGURA 7 .Proceso del Handshake

5.8 SESIÓN DE SSL

Una sesión SSL se establece por medio de un handshake entre el cliente y el servidor, esta secuencia puede variar dependiendo de si el servidor entrega un certificado o solicita el cliente. Una vez que la sesión SSL ha sido establecida, se puede reutilizar con el fin de evitar una pérdida de recursos en un nuevo handshake. Para esto cada servidor asigna a cada sesión SSL un identificador de sesión único el cual se guarda en el caché del servidor y que utilizará el cliente para las conexiones venideras, esto hasta que el identificador de sesión expire en el caché del servidor.

Los elementos de un handshake son los siguientes:

1. Negociar la suite de cifrado a ser utilizada durante la transferencia de datos. Una suite de cifrado tiene los siguientes componentes:
   - Método de intercambio de llaves.
   - Cifrado para la transferencia de datos.
   - Elección del algoritmo de Digest.

2. Establecer y compartir la llave de sesión entre el cliente y el servidor.
3. Opcionalmente autenticación del servidor al cliente.
4. Opcionalmente autenticación del cliente al servidor.

5.9 TRANSFERENCIA DE DATOS

La transferencia de datos se realiza por medio de registros de protocolo SSL fragmentando los datos en pequeñas unidades, probablemente comprimiéndolos, atando las signaturas y luego encriptando esas unidades antes de transmitirlas por medio del protocolo de red escogido.

HTTPS

Uno de los usos comunes de SSL es el de establecer una comunicación Web segura entre un browser y un Webserver. Es aquí donde se usa HTTPS que es básicamente HTTP sobre SSL con un esquema de invocación por medio de URL. Es importante hacer notar que el uso del protocolo HTTPS no impide en caso alguno que se pueda utilizar HTPP, por lo que la mayoría de los browsers advierten cuando una página tiene elementos que no son seguros en entornos seguros, como también advierten cuando se invoca un protocolo distinto al de la actual (HTTP -> HTTPS o HTTPS -> HTPP).

6. PRIVILEGIOS Y ACCESOS A FICHEROS PARA LOS USUARIOS

DERECHOS DE USUARIO

Es un atributo de un usuario o grupo de usuarios que le confiere la posibilidad de realizar una acción concreta sobre el sistema en conjunto. , la lista de derechos de cada usuario se añade explícitamente a la acreditación (SAT) que el sistema construye cuando el usuario se conecta al sistema. . Esta lista incluye los derechos que el usuario tiene concedidos a título individual más los que tienen concedidos todos los grupos a los que el usuario pertenece.

Distingue entre dos tipos de derechos: los *derechos de conexión* (*logon rights*) y los *privilegios* (*privileges*). Los primeros establecen las diferentes formas en que un usuario puede conectarse al sistema de forma interactiva, a través de la red. Mientras que los segundos hacen referencia a ciertas acciones predefinidas que el usuario puede realizar una vez conectado al sistema.

Es importante hacer notar lo siguiente: cuando existe un conflicto entre lo que concede o deniega un permiso y lo que concede o deniega un derecho, este último tiene prioridad.

OTRAS DIRECTIVAS DE SEGURIDAD

Dentro de esta herramienta de administración podemos establecer, entre otras, los siguientes tipos de reglas de seguridad para el equipo local:

Cuentas: En este apartado podemos establecer cuál es la *política de cuentas* o de contraseñas que sigue el equipo para sus usuarios locales. Dentro de este apartado se pueden distinguir reglas en tres epígrafes: *Contraseñas*, *Bloqueo* y *Kerberos*. Entre ellas, las dos primeras hacen referencia a cómo deben ser las contraseñas en el equipo (longitud mínima, vigencia máxima, historial, etc.) y cómo se debe bloquear una cuenta que haya alcanzado un cierto máximo de intentos fallidos de conexión local.

Directiva local: Dentro de este apartado se encuentra, por una parte, la *Auditoría* del equipo, que permite registrar en el visor de sucesos ciertos eventos que sean interesantes, a criterio del administrador (por ejemplo, inicios de sesión local). Por otra parte, este apartado incluye los *derechos y privilegios* que acabamos de explicar.

Claves públicas: Este apartado permite administrar las opciones de seguridad de las claves públicas emitidas por el equipo.

DERECHOS DE CONEXIÓN	
Nombre	**Significado**
Acceder a este equipo desde la red	Permite/impide al usuario conectar con el ordenador desde otro ordenador a través de la red.
Inicio de sesión local	Permite/impide al usuario iniciar una sesión local en el ordenador, desde el teclado del mismo.
PRIVILEGIOS	
Nombre	**Significado**
Añadir estaciones al dominio	Permite al usuario añadir ordenadores al dominio actual.
Hacer copias de seguridad	Permite al usuario hacer copias de seguridad de archivos y carpetas.
Restaurar copias de seguridad	Permite al usuario restaurar copias de seguridad de archivos y carpetas.
Atravesar carpetas	Permite al usuario acceder a archivos a los que tiene permisos a través de una ruta de directorios en los que puede no tener ningún permiso.
Cambiar la hora del sistema	Permite al usuario modificar la hora interna del ordenador.
Instalar manejadores de dispositivo	Permite al usuario instalar y desinstalar manejadores de dispositivos *Plug and Play*.
Apagar el sistema	Permite al usuario apagar el ordenador local.
Tomar posesión de archivos y otros objetos	Permite al usuario tomar posesión (hacerse propietario) de cualquier objeto con atributos de seguridad del sistema (archivos, carpetas, objetos del Directorio Activo, etc.).

TABLA 5. Permisos y derechos para usuario Linux Cliente/Servidor.

6.1 ASOCIACIÓN DE PERMISOS A RECURSOS

La asociación de permisos a archivos y carpetas sigue una serie de reglas:

- Cuando se crea un nuevo archivo o carpeta, este posee por defecto permisos heredados (de la carpeta o unidad donde se ubica) y ningún permiso explícito.
- Cualquier usuario que posea control total sobre el archivo o carpeta (por defecto, su propietario) puede incluir nuevos permisos (positivos o negativos) en la lista de permisos explícita.
- Copiar un archivo o carpeta a otra ubicación se considera una creación, y por tanto el archivo copiado recibe una lista de permisos explícitos vacía y se activa la herencia de la carpeta (o unidad) padre correspondiente a la nueva ubicación.
- Mover un archivo distingue dos casos: si movemos una carpeta o

archivo a otra ubicación dentro del mismo volumen (partición) NTFS, se desactiva la herencia y se mantienen los permisos que tuviera como explícitos en la nueva ubicación. Si el volumen destino es distinto, entonces se actúa como en una copia (sólo se tienen los permisos heredados de la carpeta padre correspondiente a la nueva ubicación).

6.2 PERMISOS ESTÁNDAR E INDIVIDUALES

Los permisos estándar son combinaciones predefinidas de *permisos individuales*, que son aquellos que controlan cada una de las acciones individuales que se pueden realizar sobre carpetas y archivos. La existencia de estas combinaciones predefinidas es el resultado de una agrupación ``lógica'' de los permisos individuales para facilitar la labor del administrador y de cada usuario cuando administra los permisos de sus archivos.

CARPETAS	
Nombre	**Significado**
Listar	Permite listar la carpeta: ver los archivos y subcarpetas que contiene.
Leer	Permite ver el contenido de los archivos y subcarpetas, así como su propietario, permisos y atributos (sistema, sólo lectura, oculto, etc.).
Escribir	Permite crear nuevos archivos y subcarpetas. Permite modificar los atributos de la propia carpeta, así como ver su propietario, permisos y atributos.
Leer y Ejecutar	Permite moverse por la jerarquía de subcarpetas a partir de la carpeta, incluso si no se tienen permisos sobre ellas. Además, incluye todos los permisos de Leer y de Listar.
Modificar	Permite eliminar la carpeta más todos los permisos de Escribir y de Leer y Ejecutar.
Control Total	Permite cambiar permisos, tomar posesión y eliminar subcarpetas y archivos (aun no teniendo permisos sobre ellos), así como todos los permisos anteriores.

Tabla 6. Permisos estándar sobre carpetas en Linux

Nombre	Significado
Atravesar carpeta/ejecutar archivo	Aplicado a una carpeta, permite moverse por subcarpetas en las que puede que no se tenga permiso de acceso. Aplicado a un archivo, permite su ejecución.
Leer carpeta/Leer datos	Duplicado a una carpeta, permite ver los nombres de sus ficheros y subcarpetas. Aplicado a un archivo, permite leer su contenido.
Leer atributos	Permite ver los atributos del fichero/carpeta, tales como oculto o sólo lectura, definidos en NTFS.
Leer atributos extendidos	Permite ver los atributos extendidos del archivo o carpeta. (Estos atributos están definidos por los programas y pueden variar).
Crear ficheros/escribir datos	Aplicado a unas carpetas, permite crear archivo en ella. Aplicado a un archivo, permite modificar y sobreescribir su contenido.
Crear carpetas/anexar datos	Aplicado a una carpeta, permite crear subcarpetas en ella. Aplicado a un archivo, permite añadir datos al mismo.
Escribir atributos	Permite modificar los atributos de un archivo o carpeta.
Escribir atributos extendidos	Permite modificar los atributos extendidos de un archivo o carpeta.
Borrar subcarpetas y archivos	Sólo se puede aplicar a una carpeta, y permite borrar archivos o subcarpetas de la misma, aun no teniendo permiso de borrado en dichos objetos.
Borrar	Permite eliminar la carpeta o archivo.
Leer permisos	Permite leer los permisos de la carpeta o archivo.
Cambiar permisos	Permite modificar los permisos de la carpeta o archivo.
Tomar posesión	Permite tomar posesión de la carpeta o archivo.

Tabla 7. Permisos individuales en Linux

Permiso	C. Total	Modificar	Leer y Ej.	Listar	Leer	Escribir
Atravesar carpeta/ejecutar archivo	√	√	√	√		
Leer carpeta/Leer datos	√	√	√	√	√	
Leer atributos	√	√	√	√	√	
Leer atributos extendidos	√	√	√	√	√	
Crear ficheros/escribir datos	√	√				√
Crear carpetas/anexar datos	√	√				√
Escribir atributos	√	√				√
Escribir atributos extendidos	√	√				√
Borrar subcarpetas y archivos	√					
Borrar	√	√				
Leer permisos	√	√	√	√	√	√
Cambiar permisos	√					
Tomar posesión	√					

Tabla 8. Correspondencia de permisos estándar a individuales en Linux.

CONCLUSIÓN

- Linux soporta muchas maneras de trabajo en red, se tendría que tener el suficiente cuidado para asegurar el servidor Linux. Solo se activar los servicios que se necesiten.
- Tener actualizaciones para los parches de seguridad.
- Muchas distribuciones de Linux incluyen programas 'password' que no permiten establecer una contraseña fácilmente.
- Asegurar que el programa password este al día y que tenga características. Y comprobar los logs del sistema diariamente para descubrir actividades anormales como el escaneado de puertos.
- Familiarizarse con los procesos que se ejecutan normalmente en el sistema y comprobar regularmente que no haya procesos inusuales.
- Escanear sistemas para descubrir archivos o directorios sospechosos y nombres de dispositivos inusuales
- Seguridad implementada con el Protocolo SSH para una comunicación más segura.

LISTA DE TABLAS

ILUSTRACIONES

ABREVIATURAS

BIND	Berkley Internet Name Domain Server
SOA	Start of authority
NS	Name server
A	Address record
PTR	Pointer record
MX	Mail exchanger
CNAME	Canonical name
POP	Protocolo de Oficina de Correos
TCP	Protocolo de Control de Transmisión
HTTP	Protocolo de Transferencia de Hipertexto
DNS	Servidor de Nombres de Dominio
SMTP	Servidor de correo Saliente
HTML	Lenguaje de marcas de Hipertexto.
URL	Localizador Universal de Recursos

FUENTES

Kevin Fenzi (kevin@scrye.com) & Dave Wreski (dave@nic.com). *Linux Security HOWTO* v0.9.11, 1 May 1998.

Matt Welsh, Phil Hughes, David Bandel, Boris Beletsky, Sean Dreilinger, Robert Kiesling, Evan Liebovitch, Henry Pierce. *Linux Installation and Getting Started* Red Hat Version 3.2, 20 Feb 1998.

Terry Dawson, VK2KTJ, Alessandro Rubini (maintainer),alessandro.rubini@linux.it. *Linux NET-3-HOWTO, Linux Networking.* v1.3, 1 April 1998.

Maintained by Peter Baer Galvin *The Solaris Security FAQ* SunWorld, URL: http://www.sunworld.com/common/security-faq.html, Last modified: Thursday, April 01, 1999.