UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FES ARAGÓN

INGENIERÍA EN COMPUTACIÓN

LINUX COMO PROPUESTA ANTE LA FALTA DE UNA LEGISLACIÓN INFORMÁTICA EN MÉXICO

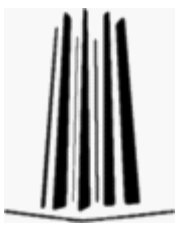
Tesis
Que para obtener el título de:

INGENIERO EN COMPUTACIÓN

Presenta:

GUILLERMO ALEJANDRO VINIEGRA PÉREZ.

DIRECTOR DE TESIS: ING. JUAN GASTALDI PÉREZ.



San Juan de Aragón, Edo., de México **Octubre 2006**

Dedicatoria
A mis hermanos que muchas
Veces me ayudaron a lo largo de mi vida
Académica, A mis abuelitos Maria, Lupita y Guillermo

Agradecimientos

Primero me gustaría agradecer a dios por darme la oportunidad de concluir la carrera y tener unos padres como los que tengo (José Viniegra Morales y Jezabel Pérez Ortiz), que siempre me han apoyado no solo en la vida académica si no también en todo lo que deseo emprender y me han enseñado que siempre hay que ir mas allá, no ponernos obstáculos, también agradecer a toda mi familia que me sabido apoyar a cada momento, agradezco también a mis amigos que han compartido su amistad conmigo y me han ayudado en los buenos y los malos momentos por los que he pasado y que nunca me han negado su amistad, también agradezco a Victoria Oropeza Ocampo que ha sido especial y única, que me ha dado su amor y apoyo incondicional, porque ha estado a mi lado y me ha motivado a finalizar este trabajo, sin importarle las dificultades, Por ultimo y no menos importantes a mis profesores que día con día fueron inculcándome en las aulas de la FES Aragon el interés por esta gran carrera y en especial a mi asesor el Ing. Juan Gastaldi Pérez que me apoyo en la elaboración de esta tesis y también a todos los que indirectamente ayudaron en la elaboración de este trabajo.

ÍNDICE

II. LINUX

III. SEGURIDAD EN LINUX

INTRODUCCIÓN.

Seguridad[1].

Es una característica de cualquier sistema (informático o no) que nos indica una situación libre de todo peligro, daño o riesgo, y de cierto carácter, infalible. Esta característica, especificando en el caso de sistemas operativos o redes de computadoras, es difícil de lograr (según la mayoría de expertos, imposible), proporcionando una tendencia flexible en dichos aspectos resultando así el concepto de *fiabilidad*[2] más que de seguridad; por consecuencias, se especula como sistemas fiables en sucesión de sistemas seguros.

A grosso modo se entiende que mantener un sistema seguro (o fiable) consiste básicamente en garantizar tres aspectos: *confidencialidad, integridad y disponibilidad*. Algunos estudios constituyen la seguridad dentro de una propiedad más general de los sistemas. La *confiabilidad*, entendida como el nivel de calidad del servicio ofrecido. Consideran la disponibilidad como un aspecto semejante a la seguridad y sin ser parte de ella, implicando la división de esta última en sólo las dos facetas restantes, *confidencialidad e integridad*.

La ***confidencialidad*** indica que los objetos de un sistema tienen acceso únicamente por elementos autorizados a ello, y que esos elementos autorizados no será información disponible para otras entidades.

La ***integridad*** expresa que los objetos sólo pueden ser modificados[3] por elementos autorizados, y de una manera controlada, por lo tanto la disponibilidad indica que los objetos del sistema tienen que permanecer accesibles a elementos autorizados; refutando la negación de servicio. Generalmente tienen que existir los tres aspectos descritos para que exista seguridad. Por ejemplo, en un sistema militar se antepondrá la confidencialidad de los datos almacenados o transmitidos sobre su disponibilidad: seguramente, es preferible que alguien borre información confidencial (que se podría recuperar después desde una cinta de backup) a correr el riesgo de que un atacante pueda leerla, o que dicha información esté disponible en un instante dado para los usuarios autorizados. En contraste, un servidor NFS de un departamento es mejor por su disponibilidad frente a la confidencialidad: importa poco que un atacante lea una unidad, pero que esa misma unidad no sea leída por usuarios autorizados va a suponer una pérdida de tiempo y dinero. En un entorno bancario, la faceta que más ha de preocupar a los responsables del sistema es la integridad de los datos, frente a su disponibilidad o su confidencialidad: es menos grave que un usuario consiga leer el saldo de otro que el hecho de que ese usuario pueda modificarlo.

[1] Definición de Villalón Huerta Antonio del libro segurida en Unix y Redes.
[2] Se entiende como la probabilidad de que un sistema se comporte tal y como se espera de él
[3] Por modificar entendemos escribir, cambiar, cambiar el estado, borrar y crear.

Que protegemos

Los tres elementos principales a proteger en cualquier sistema informático son el *software*[4], el *hardware*[5] y los *datos*[6]. Habitualmente los datos constituyen el principal elemento de los tres a proteger, ya que es el más amenazado y seguramente el más difícil de recuperar.
Sin embargo, en caso de pérdida de una base de datos o de un proyecto de un usuario, no tenemos un medio "original" desde el cual restaurar: Pasando obligatoriamente por un sistema de copias de seguridad, en el caso que la política de copias sea muy estricta, es difícil devolver los datos al estado en que se encontraban antes de la pérdida[7].
Contra cualquiera de los tres elementos descritos anteriormente (pero principalmente sobre los datos) se pueden realizar multitud de ataques o, dicho de otra forma, están expuestos a diferentes amenazas. Como la *interrupción, interceptación, modificación y fabricación*.
Un ataque se clasifica como ***interrupción*** si hace que un objeto del sistema se pierda, quede inutilizable o no disponible.
Se tratará de una ***interceptación*** si un elemento no autorizado consigue un acceso a un determinado objeto del sistema, en el caso de una ***modificación*** si además de conseguir el acceso consigue modificar el objeto; se considera un caso especial de la modificación: la ***destrucción***[8],
Por último, se dice que un ataque es una fabricación si se trata de una modificación destinada a conseguir un objeto similar al atacado de forma que sea difícil distinguir entre el objeto original y el "fabricado".

De que nos tenemos que proteger

A continuación se presenta una relación de los elementos que potencialmente pueden amenazar a nuestro sistema.

Personas

La mayoría de ataques a nuestro sistema van a provenir en última instancia de individuos que, intencionada o inintencionadamente, pueden causarnos enormes pérdidas. Generalmente se tratará de piratas que intentan conseguir el máximo nivel de privilegio posible aprovechando algúno (o algunos) de los riesgos lógicos.
Hoy en día cualquier administrador mínimamente preocupado por la seguridad va a conseguir un sistema relativamente fiable de una forma lógica (permaneciendo atento a vulnerabilidades de su software, restringiendo servicios, utilizando cifrado de datos entre otros.), pocos administradores

[4] Comprende el conjunto de programas lógicos que hacen funcional al hardware, tanto sistemas operativos como aplicaciones.
[5] Es el conjunto formado por todos los elementos físicos de un sistema informático, como CPUs, terminales, cableado, medios de almacenamiento secundario (cintas, CD-ROMs, diskettes. . .) o tarjetas de red.
[6] Conjunto de información lógica que manejan el software y el hardware, como por ejemplo paquetes que circulan por un cable de red o entradas de una base de datos.
[7] Quizás no es el mas caro pero si el mas difícil.
[8] Entendiéndola como una modificación que inutiliza al objeto afectado.

tienen en cuenta factores como la ingeniería social o el basurero a la hora de diseñar una política de seguridad.

- **Personal**
  Aunque los ataques pueden ser intencionados (en cuyo caso sus efectos son extremadamente dañinos, recordemos que nadie mejor que el propio personal de la organización conoce mejor los sistemas así como sus debilidades), lo normal es que más que de ataques se trate de accidentes causados por un error o por desconocimiento de las normas básicas de seguridad: un empleado de mantenimiento que corta el suministro eléctrico para hacer una reparación puede llegar a ser tan peligroso como el más experto de los administradores que se equivoca al teclear una orden y borra todos los sistemas de ficheros; en el primer caso, el "atacante" ni siquiera ha de tener acceso lógico (¡ni físico!) a los equipos, ni conocer nada sobre seguridad.
- **Ex-empleados**
  Generalmente, se trata de personas resentidas con la organización que pueden aprovechar debilidades de un sistema que conocen perfectamente para dañarlo como venganza por algún hecho que no consideran justo: amparados en excusas como "No me han pagado lo que me deben" o "Es una gran universidad, se lo pueden permitir" pueden insertar troyanos, bombas lógicas, virus, o simplemente conectarse al sistema como si aún trabajaran para la organización (muchas veces se mantienen las cuentas abiertas incluso meses después de abandonar la universidad o empresa); conseguir el privilegio necesario para dañarlo de la forma que deseen, incluso extorsionando a sus ex-compañeros o ex-jefes.
- **Curiosos**
  Recordemos también que las personas suelen ser curiosas por naturaleza; esta combinación produce una avalancha de estudiantes o personal intentando conseguir mayor privilegio del que tienen o intentando acceder a sistemas a los que oficialmente no tienen acceso.
- **Crackers**
  Los entornos de seguridad son un objetivo de los intrusos, ya sea para fisgonear, para utilizarlas como enlace hacia otras redes o simplemente por diversión. Por un lado, son redes generalmente abiertas, y la seguridad no es un factor tenido muy en cuenta en ellas; por otro, el gran número y variedad de sistemas conectados a estas redes provoca, casi por simple probabilidad, que al menos algunos de sus equipos (cuando no la mayoría) sean vulnerables a problemas conocidos de antemano. De esta forma un atacante sólo ha de utilizar un escáner de seguridad contra el dominio completo y luego atacar mediante un simple exploit los equipos que presentan vulnerabilidades para piratas con cualquier nivel de conocimientos, desde los más novatos (y a veces más peligrosos) hasta los expertos, que pueden utilizar toda la red para probar nuevos ataques o como nodo intermedio en un ataque a otros organismos, con el consiguiente deterioro de imagen.
- **Intrusos remunerados**
  Suele afectar más a las grandes empresas o a los organismos de defensa. Se trata de piratas con gran experiencia en problemas de

seguridad y un amplio conocimiento del sistema, que son pagados por una tercera parte generalmente para robar secretos o simplemente para dañar la imagen.

Amenazas lógicas

Encontramos todo tipo de programas que de una forma u otra pueden dañar a nuestro sistema, creados de forma intencionada para ello o simplemente por error (bugs o agujeros).

- **Software incorrecto**
  Son errores cometidos de forma involuntaria por los programadores de sistemas o de aplicaciones. Una situación no contemplada a la hora de diseñar el sistema de red del kernel o un error accediendo a memoria en un fichero situado pueden comprometer local o remotamente a cualquier sistema operativo.
  A estos errores de programación se les denomina bugs, y a los programas utilizados para aprovechar uno de estos fallos y atacar al sistema se les denomina exploits. Como hemos dicho, representan la amenaza más común, ya que cualquiera puede conseguir un exploit y utilizarlo contra nuestra máquina sin saber cómo funciona y sin conocimientos mínimos; existen exploits que dañan seriamente la integridad de un sistema.
- **Herramientas de seguridad**
  Estas representan un arma de doble filo: de la misma forma que un administrador las utiliza para detectar y solucionar fallos en sus sistemas o en la sub red completa, un potencial intruso las puede utilizar para detectar esos mismos fallos y aprovecharlos para atacar los equipos. Cuando las utilizan crackers que buscan información sobre las vulnerabilidades de un host o de una red completa.
  Ha quedado bastante claro que no se puede basar la seguridad de un sistema en el supuesto desconocimiento de sus problemas por parte de los atacantes.
- **Puertas traseras**
  Los programadores insertan “atajos” en los sistemas habituales de autentificación del programa o del núcleo que se está diseñando. A estos atajos se les denomina puertas traseras, y con ellos se consigue mayor velocidad a la hora de detectar y depurar fallos: por ejemplo, los diseñadores de un software de gestión de bases de datos en el que para acceder a una tabla se necesiten cuatro claves diferentes de diez caracteres cada una pueden insertar una rutina para conseguir ese acceso mediante una única clave “especial”, con el objetivo de perder menos tiempo al depurar el sistema.
- **Bombas lógicas**
  Son partes de código de ciertos programas que permanecen sin realizar ninguna función hasta que son activadas; en ese punto, la función que realizan no es la original del programa, sino que generalmente se trata de una acción perjudicial.
  Los activadores más comunes de estas bombas lógicas pueden ser la ausencia o presencia de ciertos ficheros, la ejecución bajo un

determinado UID o la llegada de una fecha concreta; cuando la bomba se activa va a poder realizar cualquier tarea que pueda realizar la persona que ejecuta el programa: si las activa el root, o el programa que contiene la bomba está setuidado a su nombre, los efectos obviamente pueden ser fatales.

- **Canales cubiertos**
  Son canales de comunicación que permiten a un proceso transferir información de forma que viole la política de seguridad del sistema; dicho de otra forma, un proceso transmite información a otros (locales o remotos) que no están autorizados a leer dicha información.
- **Virus**
  Es una secuencia de código que se inserta en un fichero ejecutable (denominado huésped), de forma que cuando el archivo se ejecuta, el virus también lo hace, insertándose a sí mismo en otros programas.
- **Gusanos**
  Es un programa capaz de ejecutarse y propagarse por sí mismo a través de redes, en ocasiones portando virus o aprovechando bugs de los sistemas a los que conecta para dañarlos.
  Al ser difíciles de programar su número no es muy elevado, pero el daño que pueden causar es muy grande: el mayor incidente de seguridad en Internet fue precisamente el Internet Worm, un gusano que en 1988 causó pérdidas millonarias al infectar y detener más de 6000 máquinas conectadas a la red.
  Un gusano puede automatizar así como ejecutar en unos segundos todos los pasos que seguiría un atacante humano para acceder a nuestro sistema: mientras una persona, por muchos conocimientos y medios que posea, tardaría como mínimo horas en controlar nuestra red completa (un tiempo razonable para detectarlo), un gusano puede realizar la misma función en pocos minutos: de ahí su enorme peligro además de sus devastadores efectos.
- **Caballos de Troya**
  Los troyanos o caballos de Troya son instrucciones escondidas en un programa de forma que éste parezca realizar las tareas que un usuario espera de él, pero que realmente ejecute funciones ocultas (generalmente en detrimento de la seguridad) sin el conocimiento del usuario; como el Caballo de Troya de la mitología griega, al que deben su nombre, ocultan su función real bajo la apariencia de un programa inofensivo que a primera vista funciona correctamente.
- **Programas conejo o bacterias**
  Bajo este nombre se conoce a los programas que no hacen nada útil, sino que simplemente se dedican a reproducirse hasta que el número de copias acaba con los recursos del sistema (memoria, procesador, disco, etc.), produciendo una negación de servicio. Por sí mismos no hacen ningún daño, siendo lo que realmente perjudica es el gran número de copias suyas en el sistema, en algunas situaciones pueden llegar a provocar la parada total de la máquina.
- **Técnicas salami**
  Es conocido como el robo automatizado de pequeñas cantidades de bienes (generalmente dinero) de una gran cantidad origen. El hecho de que la cantidad inicial sea grande y la robada pequeña hace

extremadamente difícil su detección: si de una cuenta con varios millones se roban unos centavos, nadie va a darse cuenta de ello; si esto se automatiza para, descontar un peso de cada nómina pagada en la universidad o de cada beca concedida, tras un mes de actividad seguramente se habrá robado una enorme cantidad de dinero sin que nadie se haya percatado de este hecho, ya que de cada origen se ha tomado una cantidad mínima.

Las técnicas salami no se suelen utilizar para atacar sistemas normales, sino que su uso más habitual es en sistemas bancarios; sin embargo, como en una red con requerimientos de seguridad medios es posible que haya computadoras dedicados a contabilidad, facturación de un departamento o gestión de nóminas del personal.

Catástrofes[9]

Son la amenaza menos probable contra los entornos habituales: simplemente por su ubicación geográfica, a nadie se le escapa que la probabilidad de sufrir un terremoto o una inundación que afecte a los sistemas informáticos en una gran ciudad.

Cómo protegernos

Para proteger nuestro sistema hemos de realizar un análisis de las amenazas potenciales que puede sufrir, las pérdidas que podrían generar, y la probabilidad de su ocurrencia; a partir de este análisis hemos de diseñar una política de seguridad que defina responsabilidades así como reglas a seguir para evitar tales amenazas o minimizar sus efectos en caso de que se produzcan. A los dispositivos utilizados para implementar esta política de seguridad se les denomina **mecanismos de seguridad**.

Estos se dividen en tres grandes grupos: *prevención*, *detección* y *recuperación*.

Los ***mecanismos de prevención*** son aquellos que aumentan la seguridad de un sistema durante el funcionamiento normal de éste, previniendo la ocurrencia de violaciones a la seguridad; por ejemplo, el uso de cifrado en la transmisión de datos se puede considerar un mecanismo de este tipo, ya que evita que un posible atacante escuche las conexiones hacia o desde un sistema en la red.

Por ***mecanismos de detección*** se conoce a aquellos que se utilizan para detectar violaciones de la seguridad o intentos de violación; ejemplos de estos mecanismos son los programas de auditoria como Tripwire.

Finalmente, los ***mecanismos de recuperación*** son aquellos que se aplican cuando una violación del sistema se ha detectado, para retornar a éste a su funcionamiento correcto; ejemplos de estos mecanismos son la utilización de copias de seguridad o el hardware adicional. Dentro de este último grupo de mecanismos de seguridad encontramos un subgrupo denominado *mecanismos de análisis forense*, cuyo objetivo no es simplemente retornar al sistema a su modo de trabajo normal, sino averiguar el alcance de la violación, las actividades de un intruso en el sistema, y la puerta utilizada para

[9] Pueden ser naturales o artificiales

entrar; de esta forma se previenen ataques posteriores y se detectan ataques a otros sistemas de nuestra red.
Además de los mecanismos ya mencionados; existen:

- *Mecanismos de autenticación e identificación*
  Hacen posible identificar entidades del sistema de una forma única, y posteriormente, una vez identificadas, autenticarlas (comprobar que la entidad es quien dice ser). Son los dispositivos más importantes en cualquier sistema, ya que forman la base de otros mecanismos que basan su funcionamiento en la identidad de las entidades que acceden a un objeto.
  Un grupo especialmente importante de estos son los denominados *Sistemas de Autenticación de Usuarios.*
- *Mecanismos de control de acceso*
  Cualquier objeto del sistema ha de estar protegido mediante mecanismos de control de acceso, que inspeccionan todos los tipos de acceso sobre el objeto por parte de cualquier entidad del sistema.
- *Mecanismos de separación*
  Cualquier sistema con diferentes niveles de seguridad ha de implementar dispositivos que permitan separar los objetos dentro de cada nivel, evitando el flujo de información entre objetos y entidades de diferentes niveles siempre que no exista una autorización expresa del mecanismo de control de acceso.
  Estos se dividen en cinco grandes grupos, en función de como separan a los objetos: *separación física, temporal, lógica, criptográfica y fragmentación*.
- *Mecanismos de seguridad en las comunicaciones*
  Es especialmente importante para la seguridad de nuestro sistema el proteger la integridad y la privacidad de los datos cuando se transmiten a través de la red. Para garantizar esta seguridad en las comunicaciones, hemos de utilizar ciertos mecanismos, la mayoría de los cuales se basan en la **Criptografía**[10]. Aunque cada vez se utilizan más los protocolos seguros[11], aún es frecuente encontrar conexiones en texto claro ya no sólo entre máquinas de una misma sub red, sino entre redes diferentes. Una de las mayores amenazas a la integridad de las redes es este tráfico sin cifrar, que hace extremadamente fáciles ataques encaminados a robar contraseñas o suplantar la identidad de máquinas de la red.

Los problemas de seguridad diarios son intrusiones, virus, negaciones de servicio contra una máquina que sirve páginas Web. Es en este tipo de entornos donde los mecanismos que estudiaremos se pueden aplicar más fácilmente, tanto por las características de los sistemas utilizados como por el relativamente bajo peligro de nuestros atacantes: los intrusos potencialmente interesados en nuestras máquinas serían solo curiosos u ociosos que sólo buscan un cierto status social en un grupo de aficionados a la piratería. Gente que ante la más mínima dificultad para acceder a nuestra red, la abandonarán y se dedicarán a objetivos más fáciles. Lo que debemos buscar es

[10] Por ejemplo: cifrado de clave pública, de clave privada, firmas digitales.
[11] Como SSH o Kerberos en red

defendernos contra la mayoría de los atacantes y los factores que nos puedan afectar.

En cualquier tipo de red, la seguridad es siempre un factor a tener en cuenta a la hora de administrar la propia red y sus máquinas, aunque con demasiada frecuencia su seguridad es mínima o ni siquiera existe; mereciendo invertir tiempo, y por qué no, dinero, para garantizar un mínimo nivel de seguridad que proporcione un entorno de trabajo aceptable.

No podemos limitarnos a establecer una férrea política de filtrado de paquetes o a restringir servicios, ya que los usuarios no van a aceptarlo.
La mejor defensa contra un cracker o curiosos consiste simplemente en cerrar los servicios que no sean estrictamente necesarios y mantener actualizado el software de nuestras máquinas que se pueda considerar crítico[12]. Si conseguimos que sus primeros intentos por acceder no sean fructíferos directamente dejarán el ataque para dedicarse a objetivos más fáciles.

Las empresas dedicadas a ofrecer acceso a Internet a través de la línea telefónica, así como otros servicios de red[13] son los conocidos ISPs; distinguidos tanto por sus servicios como por su inseguridad. Y es que realmente no es fácil compaginar una amplia oferta de servicios con una buena seguridad. Si estos viven justamente de permitir accesos a Internet o a sus propios servidores parece obvio que no podrán aplicar estrictas políticas de seguridad en las máquinas: mientras que por ejemplo en una empresa el administrador puede obligar relativamente a sus usuarios a utilizar protocolos cifrados, si un ISP no permite acceso FTP a los clientes que deseen colocar sus páginas Web y les obliga a usar un protocolo de transferencia de archivos que aplique criptografía, es muy probable que muchos de esos clientes abandonen y se vayan a la competencia: es más fácil utilizar el FTP clásico que instalar software adicional para poder actualizar una página Web.

[12] Por mencionar solo algunos: núcleos, demonios, ficheros
[13] Principalmente, hospedaje de páginas web

Justificación.

En la actualidad es clara la necesidad de rebasar los niveles aceptables de seguridad en cualquier red. Pues de este modo se da la circulación de todo tipo de datos aún refiriéndose a información confidencial (solo por mencionar algunos, nóminas, expedientes, presupuestos, etc.) así como privados (por ejemplo correos electrónicos, proyectos de investigación, etc.)
El quebrantamiento a dicha información nos trae como consecuencia perdidas económicas, dada la presunción de nuevas tecnologías en el caso de accesos internos no autorizados; por consiguiente los delitos informáticos en general se incrementan de manera exorbitante año tras año, alcanzando índices de 80%.
Los responsables de realizar este boicot principalmente son los Crackers por Intereses propios, como resultado se ha tenido una mayor difusión y beneficiándose con la creciente tecnología ya sea informática y de telecomunicaciones.
Con todo esto el conocimiento de ciertos aspectos informáticos o la curiosidad los lleva a realizar actos ilícitos; al romper con la privacidad o confidencialidad de cierta información.

Objetivos.

- Conocer los preceptos necesarios para lograr un nivel de seguridad más eficaz en los sistemas informáticos y redes, así como de igual manera los de la supercarretera de la información.
- Crear un procedimiento para la fácil evaluación de la seguridad de la red y detección de intrusos.
- Implementar y proponer El sistema Linux como una vía de seguridad informática.

- Crear una nueva cultura de seguridad en el tráfico de información al navegar por la red local o por Internet.
- Hacer una guía tanto para expertos como para gente que tiene el deseo de saber más sobre redes y que además sepan que puede hacer Linux por ellos.
- Proponer el uso de políticas y prácticas seguras como una alternativa ante la falta de una legislación informática en nuestro país.
- Resaltar la importancia de implantar programas de seguridad en la sociedad como en las organizaciones

I. CONCEPTOS DE REDES Y TELECOMUNICACIONES

La gran rapidez con la que Internet se ha difundido y acreditado en los últimos años nos ha llevado a una revolución en cuanto al mundo de las telecomunicaciones, originando cambios en diferentes aspectos de la sociedad. Enfocándonos a dicho auge en las telecomunicaciones se hará referencia a Internet que es en realidad un conjunto de redes independientes[14] que se encuentran enlazadas entre sí, permitiendo el intercambio de datos así como constituyendo una red mundial que resulta el medio idóneo para la captación de información, distribución de datos de todo tipo e interacción personal.

La supercarretera de la información tiene su origen en la red informática ARPAnet que comenzó a desarrollarse en los Estados Unidos como un proyecto del DARPA (*Defense Advanced Research Projects Agency*) sobre la década de los 60´s, aunque hasta el inicio de la década de los 70´s se comenzaron a crear las primeras aplicaciones. A finales de 1969 cuatro *hosts*[15] fueron conectados en esta red inicial, la cual fue creciendo rápidamente durante los años siguientes, pero fue a partir de 1972 cuando se comenzó a investigar la movilidad de los paquetes de información a través de varias redes de diferentes tipos y no necesariamente compatibles.

De esta manera se llega a enlazar redes independientes permitiendo que se puedan comunicar de forma transparente dichas computadoras. Este proyecto recibió el nombre de "Interneting", de esto resultando el concepto de referencia al sistema de redes funcionando conjuntamente y formando una red mayor; se utilizó el nombre de "***Internet***".

La red continuó extendiéndose por todo el país con gran rapidez, conectando a universidades e instituciones de investigación y educación, organizaciones gubernamentales, ONG´s además de redes privadas y comerciales.

De esta manera continuó su desarrollo durante los años 80´s extendiéndose internacionalmente, pero ha sido hasta la década de los 90´s cuando Internet se ha convertido en un nuevo y revolucionario medio de comunicación a escala mundial. Los nuevos medios desarrollados para hacer el acceso a Internet mucho más sencillo y agradable para cualquier usuario han influido notablemente en esta expansión, convirtiendo a Internet en la gran red mundial.

El uso creciente de la tecnología de la información en la actividad económica ha dado lugar a un incremento sustancial en el número de puestos de trabajo informatizados, con una relación de terminales por empleado que aumenta

[14] En las que se encuentran de área local y área extensa

[15](sistema anfitrión) Esto es un servidor o computadora muy potente que por medio de protocolos TCP/IP permite a los usuarios la comunicación con otros servidores en Internet.

constantemente en todos los sectores industriales. De ahí que sea señalada la importancia que tiene el uso adecuado tanto de la información como su gestión que poseen en la actividad de cualquier empresa.

Cuando se realiza un estudio en el ámbito de la indagación de un puesto de trabajo se encuentran modelos de distribución que indican que alrededor de un 90% de la información generada tiene como destino el propio departamento, así un 75% está consignada a un punto distante no más de 200 metros del punto de generación, y hasta un 90% queda dentro del propio edificio, lo que sólo le concede un 10% a la información dirigida a destinos remotos. Independientemente del carácter estimativo de las cifras anteriores es evidente que un esquema de distribución como el mencionado, incita a acciones contundentes para optimizar la difusión de la información que fluye en un ámbito local.

La reubicación física de los puestos de trabajo es una realidad connatural con el dinamismo de las empresas actuales. Esta movilidad lleva a unos porcentajes de cambio anual entre un 20 y un 50% del total de puestos de trabajo. Los costos de traslado pueden ser notables[16]. Por tanto, se hace necesaria una racionalización de los medios de acceso de estos equipos con el objeto de minimizar dichos costes.

Las Redes de Área Local se han creado para responder a ésta problemática. El éxito de las LAN reside en que cada día es mayor la cantidad de información que se procesa de una manera local, y a su vez mayor el número de usuarios que necesitan estar conectados entre sí, con la posibilidad de compartir recursos comunes. Por ejemplo, acceder a una base de datos general o compartir una impresora de alta velocidad.

El progreso de las redes locales a mediados de los años ochenta hizo que se reemplazará nuestra forma de comunicarnos con las computadoras y la forma en que las computadoras se comunicaban entre sí. La importancia de las LAN reside en que en un principio se puede conectar un número pequeño de computadoras que puede ser ampliado a medida que se incrementan dichas necesidades. Son el factor medular de las micro y medianas empresas ya que predeterminan la solución a un entorno distribuido.

1.1 ¿Que es una red?

Es un sistema de comunicaciones entre computadoras. Como tal, consta de un soporte físico que abarca cableada y placas adicionales en las computadoras, y un conjunto de programas que forma el sistema operativo de la red. Las redes constan de dos o más computadoras conectadas entre sí y permiten compartir recursos e información. La información por compartir suele consistir en archivos y datos. Los recursos son los dispositivos o las áreas de almacenamiento de datos de una computadora.

[16] Estos pueden ser: nuevo tendido para equipos informáticos, teléfonos, etc.

1.2 Modelo o Estándar IEEE.

Debido al gran avance en las comunicaciones se dio a la par la proliferación de redes de área local (LAN) muchos productos aparecieron, y con ello la necesidad de una uniformidad, entonces la **IEEE**[17] empezó a definir estándares de red. El proyecto fue llamado 802, en referencia al año y al mes que fue emprendido: febrero de 1980
Del proyecto 802 resultaron numerosos documentos incluyendo los tres principales estándares para las topologías de red. A continuación solo se hará mención de los conceptos ya que se profundizará en algunos de ellos posteriormente:

- 802.1.- Normalización del interfaz con niveles superiores (HLI, higher Layer Interfaz Estándar).
- 802.2.-Normalización para el control de enlace lógico (LLC, logical link control).
- 802.3.- CSMA/CD acceso por detección de portadora/ detección de colisiones la cual es uno de los principales estándares.
- 802.4.- Token Bus paso del testigo por el bus.
- 802.5.- Token Ring paso de testigo en anillo.

Estos son solo algunos de los frutos del proyecto 802 de la IEEE pero no entraremos en detalle en este tema.
Como no había un orden en cuanto a dichos estándares los fabricantes de redes trabajaban libremente por ello no había en un inicio una compatibilidad surgiendo varias formas de hacer una comunicación entre las redes y los distintos estándares como los que se enumeran a continuación:

- **Conmutadas por Circuitos:** Redes en las cuales, para establecer comunicación se debe efectuar una llamada y cuando se establece la conexión, los usuarios disponen de un enlace directo a través de los distintos segmentos de la red.
- **Conmutadas por Mensaje:** En este tipo de redes el conmutador suele ser una computadora que se encarga de aceptar tráfico de las computadoras y terminales conectados a él. La computadora examina la dirección que aparece en la cabecera del mensaje hacia el DTE[18] que debe recibirlo. Esta tecnología permite grabar la información para atenderla después. El usuario puede borrar, almacenar, redirigir o contestar el mensaje de forma automática.
- **Conmutadas por Paquetes[19]:** En este tipo de red los datos de los usuarios se descomponen en fragmentos más pequeños. Estos fragmentos o paquetes, están insertados dentro de informaciones del protocolo y recorren la red como entidades independientes.

[17] Es el instituto de ingenieros eléctricos y en electrónica

[18] Equipo terminal de datos, nombre que suele recibir en una comunicación el ordenador que recibe o envía los datos (Data Terminal Equipment).

[19] Es la segmentación de la información que se va a transmitir estos paquetes deben contener la información necesaria para llegar a su destino

Una red de computadoras permite conectar a las computadoras que la forman con la finalidad de compartir información, como documentos o bases de datos, o recursos físicos, como impresoras o unidades de disco. Las redes suelen clasificarse según su extensión en:

- ***LAN (Local Area Network):*** Son las redes de área local. La extensión de este tipo de redes suele estar restringida a una sala o edificio, aunque también podría utilizarse para conectar dos o más edificios próximos.
- ***WAN (Wide Area Network):*** Son redes que cubren un espacio muy amplio, conectando a computadoras de una cuidad o un país completo. Para ello se utilizan las líneas de teléfono y otros medios de transmisión más sofisticados, como pueden ser las microondas. La velocidad de transmisión suele ser inferior que en las redes locales.

1.2.1 Redes de Área Local (LAN)

Es un sistema de comunicaciones de alta velocidad que conecta microcomputadoras o PC´s que se encuentran cercanos, por lo general dentro del mismo edificio. Está consta de hardware y software de red; sirve para conectar las que están aisladas. Da la posibilidad de que las PC´s interactúen entre sí compartiendo programas, información y recursos, como unidades de disco, directorios e impresoras. El proceso para incorporar una PC´s o microcomputadora a una LAN consiste en la instalación de una tarjeta de interfase de red NIC[20] en cada computadora.

Los NIC de cada computadora se conectan con un cable especial de red. El último paso para implantar una LAN, es cargar a cada PC un software conocido como sistema operativo de red NOS[21]. Este trabaja con el software del sistema operativo de la computadora y permite que el software de aplicación[22] que sé esta ejecutando se comunique a través de la red con otra computadora.
Así pues se puede concluir que una ***red de área local*** es un medio de transmisión para la información que proporciona la interconexión, entre diversas computadoras terminales y periféricos situados en un entorno reducido además de pertenecientes a una sola organización.

Características de las LAN's:

- El radio que abarca es de pocos kilómetros, Por ejemplo: edificios, un campus universitario, un complejo industrial, etc.
- Utilizan un medio privado de comunicación.

[20](Centro de Información de la Red) Organismo que ofrece entre otros servicios: información, asistencia para los usuarios de Internet.

[21]Network Operating System
[22] Como el procesador de palabras, las hojas de cálculo, entre otros.

- La velocidad de transmisión es de varios millones de bps. Las velocidades más habituales van desde 1 hasta 16 Mbits, aunque se está elaborando un estándar para una red que alcanzará los 100 Mbps.
- Pueden atender a cientos de dispositivos muy distintos entre sí[23].
- Ofrecen la posibilidad de comunicación con otras redes a través de pasarelas o Gateways.
- Para el caso concreto de una red local, NOVELL NETWARE 3.12: Soporta hasta 250 usuarios trabajando de forma concurrente.
- Permite hasta 100.000 ficheros abiertos simultáneamente.
- El mismo servidor sirve de puente o Gateways con otras redes.

1.3 Redes de área extensa (WAN)

Cuando se llega a un cierto punto deja de ser poco práctico seguir ampliando una LAN. A veces esto es impuesto por limitaciones físicas, aunque suele haber formas más adecuadas o económicas de ampliar una red de computadoras. Dos de los componentes importantes de cualquier red son: teléfono y la de datos. Son enlaces para grandes distancias que amplían la LAN hasta convertirla en una red de área extensa[24]. Casi todos los operadores de redes nacionales ofrecen servicios para interconectar redes de computadoras, que van desde los enlaces de datos sencillos y a baja velocidad que funcionan basándose en la red pública de telefonía hasta los complejos servicios de alta velocidad[25] adecuados para la interconexión de las LAN. Estos servicios de datos a alta velocidad suelen denominarse *conexiones de banda ancha*. Se prevé que proporcionen los enlaces necesarios entre LAN para hacer posible lo que han dado en llamarse autopistas de la información.

Una WAN se extiende sobre un área geográfica amplia, a veces un país o un continente; contiene una colección de máquinas dedicadas a ejecutar programas de usuario[26], estás máquinas se llaman *Hosts*. Los hosts están conectados por una sub red de comunicación; su trabajo es conducir mensajes de un host a otro. La separación entre los aspectos exclusivamente de comunicación de la red (la subred) y los aspectos de aplicación (hosts), simplifica enormemente el diseño total de la red.
En muchas redes de área amplia, la subred tiene dos componentes distintos: las líneas de transmisión y los elementos de conmutación. Las líneas de transmisión (también llamadas circuitos o canales) mueven los bits de una máquina a otra.

Los elementos de conmutación son computadoras especializadas que conectan dos o más líneas de transmisión. Cuando los datos llegan por una línea de entrada, el elemento de conmutación debe escoger una línea de salida para enviarlos. Como término genérico para las computadoras de conmutación, le llamaremos enrutadores.

[23] Impresoras, computadoras, discos, teléfonos, módems, etc.
[24] WAN
[25] Como framerelay y SMDS-Synchronous Multimegabit Data Service.
[26] Es decir, aplicaciones.

1.3.1 CONSTITUCIÓN DE UNA RED DE ÁREA AMPLIA (WAN)

La red consiste en ECD[27] interconectados por canales alquilados de alta velocidad (por ejemplo, líneas de 56 kbit / s). Cada ECD utiliza un protocolo responsable de encaminar correctamente los datos y de proporcionar soporte a los computadoras además terminales de los usuarios finales conectados a los mismos. La función de soporte ETD[28] se denomina a veces PAD[29]. Para los ETD, el ECD es un dispositivo que los aísla de la red. El centro de control de red[30] es el responsable de la eficiencia y fiabilidad de las operaciones de la red.

1.3.2 CARACTERÍSTICAS DE UNA RED WAN

- Los canales suelen proporcionarlos las compañías telefónicas, con un determinado coste mensual, si las líneas son alquiladas; un costes proporcional en el caso de la utilización de líneas normales conmutadas.
- Los enlaces son relativamente lentos (de 1200 Kbit / s a 1.55Mbit / s).
- Las conexiones de los ETD con los ECD son generalmente más lentas (150 bit / s a 19.2 kbit / s).
- LOS ETD y los ECD están separados por distancias que varían desde algunos kilómetros hasta cientos de kilómetros.
- Las líneas son relativamente propensas a errores.

Las redes de área local (LAN) son significativamente diferentes de las redes de cobertura amplia. El sector de las LAN es uno de los que consta con un crecimiento más rápido en la industria de las comunicaciones. Las redes de área local poseen las siguientes las características:

- Generalmente, los canales son propiedad del usuario o empresa.
- Los enlaces son líneas (desde 1 Mbit / s hasta 400 Mbit / s). Los ETDs se conectan a la red vía canales de baja velocidad (desde 600 bit / s hasta 56 Kbit / s).
- Los ETD están cercanos entre sí, generalmente en un mismo edificio.
- Puede utilizarse un ECD para conmutar entre diferentes configuraciones, pero no tan frecuentemente como en las WAN.
- Las líneas son de mejor calidad que los canales en las WAN.

Debido a las diferencias entre las redes de área local y las redes de cobertura amplia, sus topologías pueden tomar formas muy diferentes.

[27] computadoras de conmutación.
[28] Terminales / computadoras de usuario.
[29] Packet Assembly / Disasembly – ensamblador / desensamblador de paquetes.
[30] Abreviado como CCR

La estructura de las **WAN** tiende a ser más irregular, debido a la necesidad de conectar múltiples terminales, computadoras y centros de conmutación. Como los canales están alquilados mensualmente, las empresas y organizaciones que los utilizan tienden a mantenerlos lo más ocupados posible. Para ello, a menudo los canales "serpentean" por una determinada zona geográfica para conectarse a los ETD allí donde estén. Debido a eso la topología de las WAN suele ser más irregular.

Por el contrario el propietario de una **LAN** no tiene que preocuparse de utilizar al máximo los canales, ya que son baratos en comparación con su capacidad de transmisión (los cuellos de botella en las LAN suelen estar en el SOFTWARE). Por tanto, no es tan severa la necesidad de esquemas muy eficientes de multiplexado y multidistribución. Además, como las redes de área local que residen en un mismo edificio, la topología tiende a ser más ordenada y estructurada, con configuraciones en forma de *bus, anillo o estrella*.

1.4 Componentes de una red.

Una red de computadoras esta conectada tanto por hardware como por software. El hardware incluye tanto las tarjetas de interfaz de red como los cables que las unen, y el software incluye los controladores como los programas que se utilizan para gestionar los dispositivos además del sistema operativo de red que gestiona la misma. A continuación se listan los componentes:

Tarjetas de Conexión a la Red (NIC):

Tarjeta electrónica que conectan a las estaciones de trabajo a la red. Normalmente se insertan en una de las ranuras de expansión del motherboard de la microcomputadora suministrando de esta forma acceso directo a memoria[31]. El NIC tiene las siguientes funciones:

- Forman los paquetes de datos
- Dan acceso al cable, con la conversión eléctrica y ajuste de velocidad
- Son el transmisor y el receptor de la estación
- Chequean las tramas para chequear errores
- Conversión Serie / paralelo
- Identificación o dirección única en la red que permite saber cual es físicamente la terminal

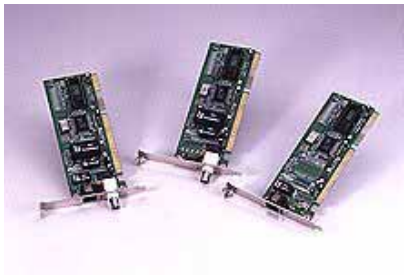


Fig. 1.1 Tarjetas de conexión a red.

[31] Mejor conocido como DMA.

Estaciones de Trabajo:

Cuando una computadora se conecta a una red, la primera se convierte en un nodo de la última y se puede tratar como una estación de trabajo o cliente. Las estaciones de trabajos pueden ser computadoras personales con el DOS, Macintosh, Unix, OS/2 o estaciones de trabajos sin discos.

Fig. 1.2 Estación de trabajo

Servidores:

Computadoras que proporcionan servicios a las estaciones de trabajo de la red tales como almacenamiento en discos, acceso a las impresoras, unidades para respaldo de archivos, acceso a otras redes o computadoras centrales.

Fig. 1.3 servidores

Repetidores:

Son equipos que trabajan a nivel 1 de la pila OSI, es decir, repiten todas las señales de un segmento a otro a nivel eléctrico.
Se utilizan para resolver los problemas de longitudes máximas de los segmentos de red (su función es extender una red Ethernet más allá de un segmento). No obstante, hay que tener en cuenta que, al retransmitir todas las señales de un segmento a otro, también retransmitirán las colisiones. Estos equipos sólo se aíslan entre los segmentos los problemas eléctricos que pudieran existir en algunos de ellos.
El número máximo de repetidores en cascada es de cuatro, pero con la condición de que los segmentos 2 y 4 sean IRL[32], es decir, que no tengan ningún equipo conectado que no sean los repetidores. En caso contrario, el número máximo es de 2, interconectando 3 segmentos de red.
El repetidor tiene dos puertas que conectan dos segmentos Ethernet por medio de transceivers[33] y cables drop.

[32] Interactive Reader Language + Inter-Repeater Link
[33] Instalando diferentes transceivers es posible interconectar dos segmentos de diferentes medios físicos.

El repetidor tiene como mínimo una salida Ethernet para el cable amarillo y otra para teléfono.
Con un repetidor modular se pude centralizar así como estructurar todo el cableado de un edificio, con diferentes medios, adecuados según el entorno, además de las conexiones al exterior.
Un Concentrador es un equipo igual a un multiport repeater pero con salida RJ-45.
Los repetidores con buffers es la unión de dos redes por una línea serie mediante una pareja de repetidores.

Puentes o Bridges.

Estos equipos se utilizan asimismo para interconectar segmentos de red[34], se utilizan cuando el tráfico no es excesivamente alto en las redes pero interesa aislar las colisiones que se produzcan en los segmentos interconectados entre sí.
Los Bridges trabajan en el nivel 2 de OSI, con direcciones físicas, por lo que filtra tráfico de un segmento a otro.
Esto lo hace de la siguiente forma: Escucha los paquetes que pasan por la red de igual manera va configurando una tabla de direcciones físicas de equipos que tiene a ambas partes (generalmente tienen una tabla dinámica), de tal forma que cuando escucha en un segmento un paquete de información que va dirigido a ese mismo segmento no lo pasa al otro, y viceversa.
No filtra los broadcasts, que son paquetes genéricos que lanzan los equipos a la red para que algún otro les responda, aunque puede impedir el paso de determinados tipos de broadcast. Por ejemplo; esto es típico para solicitar las cargas de software. Por tanto, al interconectar segmentos de red con Bridges, podemos tener problemas de tormentas de broadcasts, de saturación del puente por sobrecarga de tráfico, etc.
El número máximo de puentes en cascada es de siete; no pueden existir bucles o lazos activos, es decir, si hay caminos redundantes para ir de un equipo a otro, sólo uno de ellos debe estar activo, mientras que el redundante debe ser de backup. Para esto, cuando se está haciendo bridging en las redes, se usa el algoritmo de spanning-tree, mediante el cual se deshacen los bucles de los caminos redundantes.
Las posibles colisiones no se transmiten de un lado a otro de la red. El bridge sólo deja pasar los datos que van a un equipo que él conoce.
El bridge generalmente tiene una tabla dinámica, aíslan las colisiones***, pero no filtran protocolos***. Además trabaja en el nivel 2 de OSI y aísla las colisiones
La primera vez que llega un paquete al bridge lo transmitirá, pero aprende (ya que, si el paquete no lo recibe nadie, significa que no está).
Su peligro consiste en que cuando hay exceso de broadcast y se colapsa la red. A esto se le llama tormenta de broadcast, y se produce porque un equipo está pidiendo ayuda (falla).

[34] Amplía una red que ha llegado a su máximo, ya sea por distancia o por el número de equipos.

Routers.

Estos equipos trabajan a nivel 3 de la pila OSI, es decir pueden filtrar protocolos y direcciones a la vez. Los equipos de la red saben que existe un router es por ello que les envía los paquetes directamente a él cuando se trate de equipos en otro segmento.
Además los routers pueden interconectar redes distintas entre sí; eligen el mejor camino para enviar la información, balancean tráfico entre líneas, etc.
Este trabaja con tablas de encaminamiento o enrutado con la información que generan los protocolos, deciden si hay que enviar un paquete o no, deciden cual es la mejor ruta para enviar la información de un equipo a otro, pueden contener filtros a distintos niveles, etc.
Poseen una entrada con múltiples conexiones a segmentos remotos, garantizan la fiabilidad de los datos así permiten un mayor control del tráfico de la red. Su método de funcionamiento es el *encapsulado de paquetes*.
Para interconectar un nuevo segmento a nuestra red, sólo hace falta instalar un router que proporcionará los enlaces con todos los elementos conectados.

Brouters

Dispositivos con funciones combinadas de bridge y router. Cuando se configura se le indica la modalidad en la cual va a funcionar, como bridge o como router.

Transceivers.

Son equipos que son una combinación de transmisor / receptor de información, es por ello que transmite paquetes de datos desde el controlador al bus y viceversa.
En una Ethernet, los transceivers se desconectan cuando el equipo al que están conectados no está funcionando, sin afectar para nada al comportamiento de la red.

Switchs

Divide la red LAN en varios segmentos limitando el tráfico a uno o más segmentos en vez de permitir la difusión de los paquetes por todos los puertos. Dentro del Switch, un circuito de alta velocidad se encarga del filtrado además de permitir el tránsito entre segmentos que tengan la intención de hacerlo.

Gateways.

También llamados traductores de protocolos, son equipos que se encargan, como su nombre indica, a servir de intermediarios entre los distintos protocolos de comunicaciones para facilitar la interconexión de equipos distintos entre sí.
Su forma de funcionar es que tienen duplicada la pila OSI, es decir, la correspondiente a un protocolo y, paralelamente, a la del otro. Reciben los

datos encapsulados, los van desencapsulando hasta el nivel más alto, para posteriormente ir encapsulando los datos en el otro protocolo desde el nivel más alto al nivel más bajo, y vuelven a dejar la información en la red, pero ya traducida. Los Gateways también pueden interconectar redes entre sí.

1.5 Concentradores

1.5.1 MAU (Multistation Access Unit)

Concentrador que permite insertar en el anillo o eliminar derivándolas, hasta 8 estaciones. Detecta señales procedentes de las estaciones de trabajo, en caso de detectarse un dispositivo defectuoso o un cable deteriorado y elimina, derivándola, la estación en cuestión para evitar perdidas de datos y del TOKEN.

1.5.2 Hubs

Concentradores de cableado en estrella integrados por microprocesadores, memoria y protocolos como SNMP, características que lo convierten en un nodo inteligente en la red capaz de controlar y diagnosticar, incluso por monitoreo remoto.

Fig. 1.4 hub 3com

1.6 Topologías de una Red

1.6.1 Topologías en Bus

En la topología en bus, al contrario que en la topología de Estrella, no existe un nodo central, si no que todos los nodos que componen la red quedan unidos entre sí linealmente, uno a continuación del otro.

El cableado en bus presenta menos problemas logísticos, puesto que no se acumulan montones de cables en torno al nodo central, como ocurriría en una disposición en estrella. Pero, por contra, tiene la desventaja de que un fallo en una parte del cableado detendría el sistema, total o parcialmente, en función del lugar en que se produzca.
Es además muy difícil encontrar y diagnosticar las averías que se producen en esta topología.

Debido a que en el bus la información recorre todo el bus bidireccionalmente hasta hallar su destino, la posibilidad de interceptar la información por usuarios no autorizados es superior a la existente en una Red en estrella debido a la modularidad que ésta posee.
La red en bus posee un retardo en la propagación de la información mínimo, debido a que los nodos de la red no deben amplificar la señal, siendo su función pasiva respecto al tráfico de la red. Esta pasividad de los nodos es debida al método de acceso empleado que a la propia disposición geográfica de los puestos de red.
La Red en Bus necesita incluir en ambos extremos del bus, unos dispositivos llamados terminadores, los cuales evitan los posibles rebotes de la señal, introduciendo una impedancia característica (50 Ohm.)
Añadir nuevos puesto a una red en bus, supone detener al menos por tramos, la actividad de la red. Sin embargo es un proceso rápido y sencillo. Es la topología tradicionalmente usada en redes Ethernet.

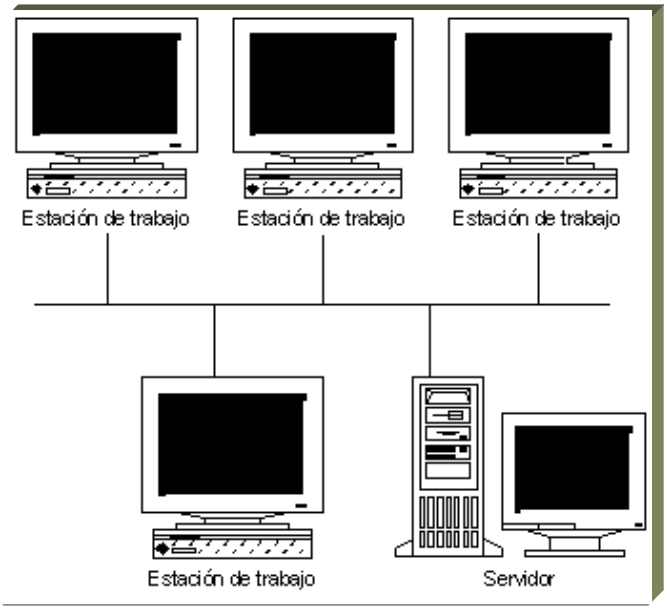


Fig. 1.5 Topología Bus.

1.6.2 Topología en Anillo

Como su propio nombre indica, consiste en conectar linealmente entre sí todas las computadoras, en un bucle cerrado. La información se transfiere en un solo sentido a través del anillo, mediante un paquete especial de datos, llamado **testigo**, que se transmite de un nodo a otro, hasta alcanzar el nodo destino.

El cableado de la red en anillo es el más complejo de los tres enumerados, debido por una parte al mayor coste del cable, así como a la necesidad de emplear unos dispositivos denominados Unidades de Acceso Multiestación[35] para implementar físicamente el anillo.
A la hora de tratar con fallos y averías, la red en anillo presenta la ventaja de poder derivar partes de la red mediante los MAU's, aislando dichas partes defectuosas del resto de la red mientras se determina el problema. Un fallo,

[35] Sus siglas son MAU

pues, en una parte del cableado de una red en anillo, no debe detener toda la red. La adición de nuevas estaciones no supone una complicación excesiva, puesto que una vez más los MAU's aíslan las partes a añadir hasta que se hallan listas, no siendo necesario detener toda la red para añadir nuevas estaciones. Dos buenos ejemplos de red en anillo serían Token-Ring y FDDI (fibra óptica)

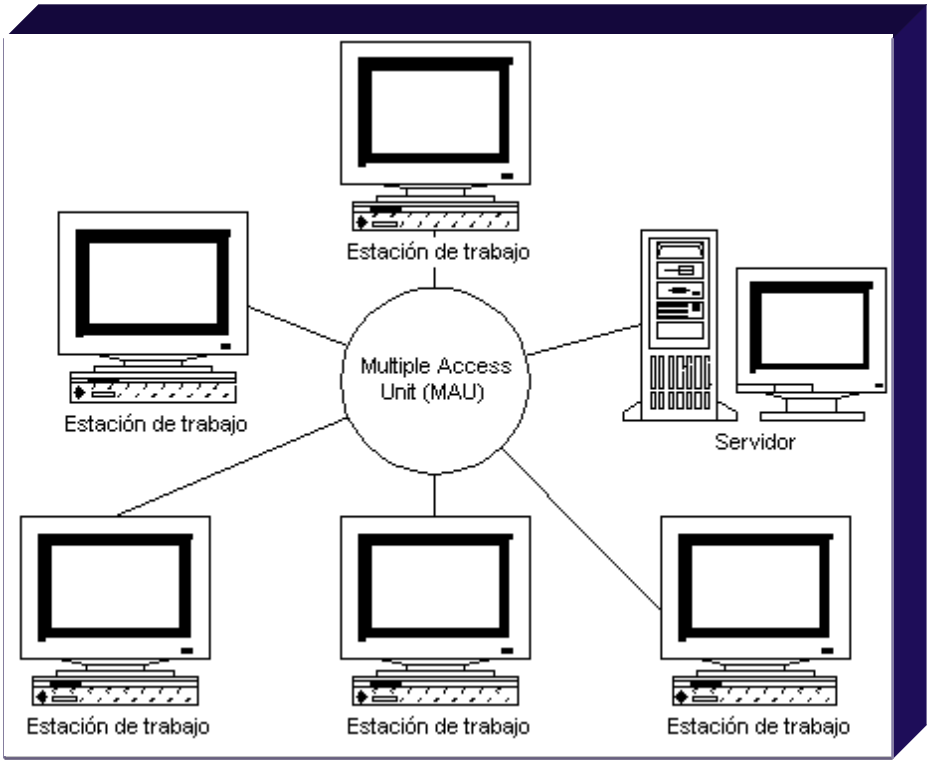


Fig. 1.6 Topología Anillo

1.6.3 Topología en Estrella.

Esta topología se caracteriza por existir en ella un punto central, o más propiamente nodo central, al cual se conectan todos los equipos, de un modo muy similar a los radios de una rueda.

De esta disposición se deduce el inconveniente de esta topología, y es que la máxima vulnerabilidad se encuentra precisamente en el nodo central, ya que si este falla, toda la red fallaría.

Este posible fallo en el nodo central, aunque posible, es bastante improbable, debido a la gran seguridad que suele poseer dicho nodo. Sin embargo presenta como principal ventaja una gran modularidad, lo que permite aislar una estación defectuosa con bastante sencillez y sin perjudicar al resto de la red.

Para aumentar el número de estaciones, o nodos, de la red en estrella no es necesario interrumpir, ni siquiera parcialmente la actividad de la red, realizándose la operación casi inmediatamente. La topología en estrella es empleada en redes Ethernet y ArcNet.

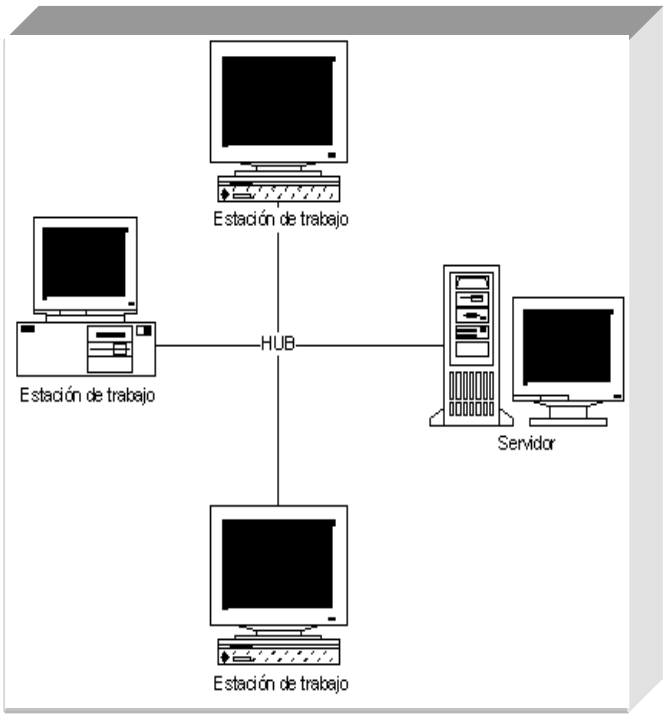


Fig. 1.7 Topología estrella

1.7 Medios de transmisión

1.7.1 Par trenzado

- Grosor de 1mm.
- El ancho de banda depende del grosor y de la distancia.
- Velocidad del orden de 10-100 Mbps.
- Categorías de cable par trenzado:
  - **STP** (blindado): 2 pares de hilo, recubierto por malla.
  - **UTP** (no blindado): 4 pares de hilos.

El UTP y el STP se utilizan a velocidades de hasta 150Mbps con longitudes no superiores a 100 metros, con una atenuación de 30 dB/300 mts. A 10 mhz. Su impedancia es de 100 ohms para el UTP[36] y 120 a 150 para los STP[37] con conectores RJ45 y RJ11. Con el cable trenzado se reducen las corrientes parásitas, para trabajar con el cable blindado tanto el conector de la tarjeta como el RJ45 deben ser blindados.

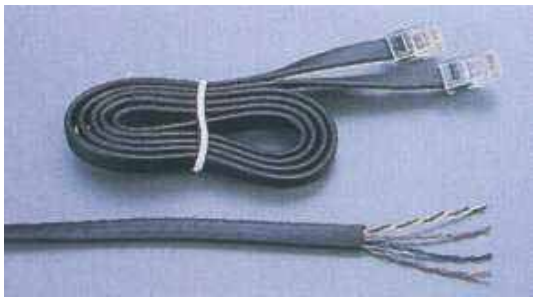


Fig. 1.8 cable par trenzado

[36] Es decir, Unshield twisted pair.
[37] Que quiere decir Shiel Twisted Pair.

- ✓ **Categoría 3:** van de 4 en 4 (8 cables), alcanzando 30 Mbps .
- ✓ **Categoría 5:** por el se pueden enviar texto multimedia imágenes, más retorcidos y mejor aislante (teflón), alcanzando 100 Mbps

Numeración del conector RJ45

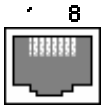
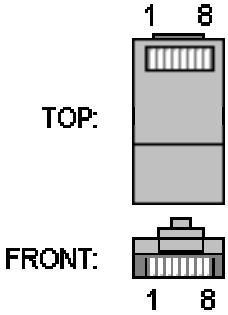
Hembra	Macho
Visto de frente	Conector visto de frente y desde arriba
	

Fig. 1.9 configuración conector RJ45

1.7.2 Cable coaxial

Este consta de un par de conductores de cobre o aluminio. Uno de ellos forma el alma central y esta rodeado por el segundo conductor constituido por una malla muy fina de hilos trenzados o una lamina metálica cilíndrica. La separación y asilamiento entre los conductores se efectúa por un material dieléctrico de teflón o plástico. El cable esta cubierto para reducir las emisiones eléctricas.
Los hay de 2 impedancias:

- **75 ohmios**: banda ancha, utilizado en TV, distintos canales, 300MHz, 1800m uno de transmisión y otro de recepción un total de 3600m.
- **50 ohmios**: banda base, utilizado en Ethernet, un canal.
- **10BASE5:** coaxial grueso, con una impedancia de 50 ohms, 500 metros, 10Mbps, conector "N".
- **10BASE2:** coaxial fino, con una impedancia de 50 ohms, 185 metros, 10 Mbps, conector "BNC".

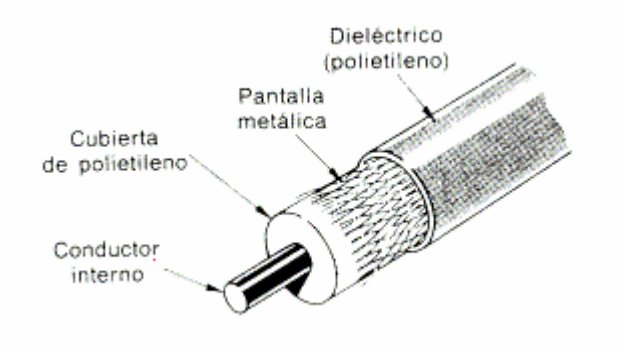


Fig. 1.10 Cable Coaxial

CABLE	CARACTERÍSTICAS
10-BASE-5	Cable coaxial grueso (Ethernet grueso). **Velocidad de transmisión:** 10 Mb/seg. **Segmentos:** máximo de 500 metros.
10-BASE-2	Cable coaxial fino(Ethernetfino). **Velocidad de transmisión:** 10Mb/seg. **Segmentos:** máximo de 185 metros.
10-BROAD-36	Cable coaxial **Segmentos:** máximo de 3600 metros. **Velocidad de transmisión:** 10 Mb/seg.
100-BASE-X	Fast Ethernet. **Velocidad de transmisión:** 100 Mb/seg.

Tabla 1.1 características de cable coaxial

1.7.3 Fibra óptica

Surgió por tratar de transmitir a mayor velocidad. El modo de transmisión es duplex por ello se deben de tener dos fibras ópticas.
Tiene dos formas de transmitir:

- Mono modo (manda una sola señal a la vez): se alcanzan distancias mayores y velocidades altas y es la más usada.
- Multimodo (manda varias señales ala vez): distancias cortas.

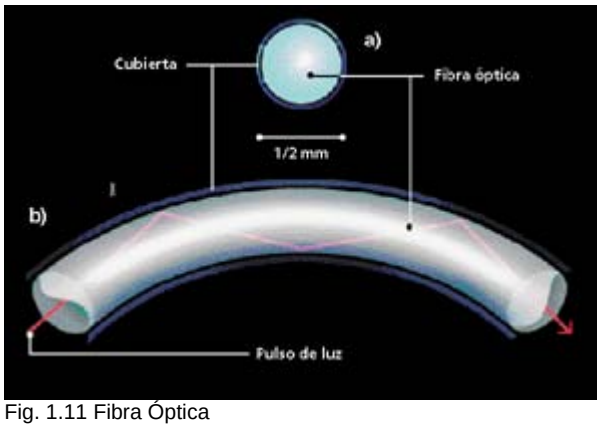


Fig. 1.11 Fibra Óptica

Para transmitir en una fibra óptica se hace sin pasar de cierto ángulo establecido, una de las ventajas es que es inmune a los campos magnéticos y es mas seguro en que no se roben los datos que circulan en ella.

- Se necesita una fuente de luz: láser o LED.
- Se transmite por fibra y se capta por foto diodos.
- La topología típica es el anillo
- Alcanza un ancho de banda de 30000GHz .
- Sólo necesita repetidores cada 30 kms.
- No hay interferencias.
- Pesa 8 veces menos que el cable par trenzado.

1.8 Métodos de acceso

Se denomina así; a la posibilidad de transmitir datos por la red; hay dos formas básicas:
CSMA/CD[38] y el Token passing.

1.8.1 CSMA/CD

Para facilitar este estudio dividiremos el siguiente proceso en cuatro fases:
Fase a. En este caso, cualquier máquina puede iniciar una comunicación (acceso múltiple) con sólo verificar que no haya ninguna otra comunicación en el cable; para ello detecta la presencia de portadora[39].
Fase b. La información que se está transmitiendo tarda un cierto tiempo en recorrer la red. Una estación a la que todavía no le llegaron los primeros bits podría iniciar una transmisión basada en que en ese momento no hay señal.
Fase c. Un instante después le empezarán a llegar dichos bits, pero como la transmisión ya había comenzado, las estaciones comprendidas entre ambas máquinas recibirán la suma de las dos señales. Esto se denomina "colisión".
Fase d. El segundo transmisor debe seguir transmitiendo un tiempo suficiente como para que el primero se entere de la colisión. Esta acción recibe el nombre de atascamiento (jamming).

Análisis de una colisión.

El peor caso de colisión se produce cuando las estaciones están a la mayor distancia posible y la segunda comienza a transmitir justo antes de recibir el primer bit, pues al tiempo de propagación de la señal de la primera estación a la segunda, hay que sumarle el de propagación del atascamiento de la segunda a la primera. La suma de esos tiempos define la "ventana de colisión".

[38] Carrier Sense Multiple Access with Collission Detection.
[39] Llamado Carrier Sense

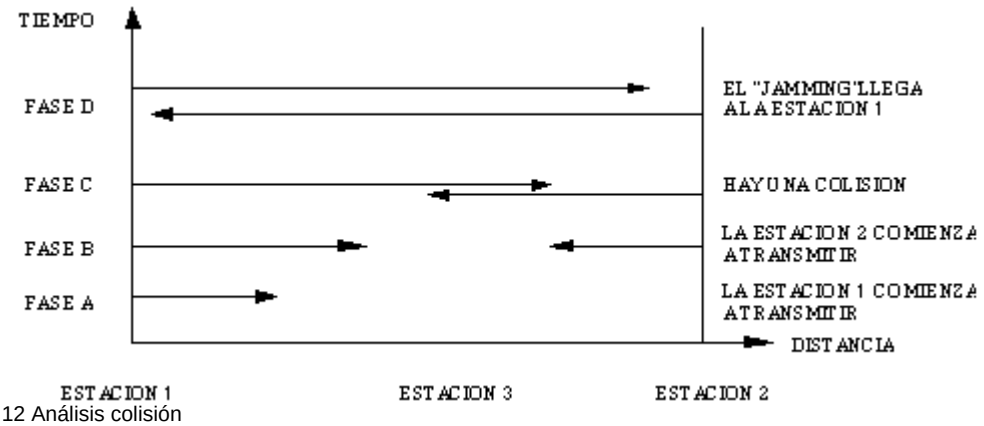


Fig. 1.12 Análisis colisión

Para asegurarse la ausencia de colisiones indetectadas, se deben cumplir dos condiciones:
1. La transmisión debe durar más que la ventana de colisiones. Por ejemplo en Ethernet el paquete mínimo es de 46 bytes y el máximo de 1500 bytes.
2. La estación transmisora debe chequear la ausencia de colisiones durante ese tiempo; después no es necesario.
Una vez detectada la colisión, ambas estaciones deben dejar pasar un tiempo determinado cuasialeatoriamente antes de intentar retransmitir. Si se produce otra colisión, se reintenta esperando un tiempo mayor. El tiempo promedio de demora se duplica con cada reintento. Puede haber colisiones múltiples. Es posible que una estación no pueda comunicarse durante mucho tiempo debido a una sucesión de colisiones.

1.8.2 Token passing:

Este sistema evita la colisión pues limita el derecho a transmitir a una máquina. Esa máquina se dice que tiene el Token (cospel). Este va pasando a intervalos fijos de una máquina a otra. La circulación de este a la siguiente máquina hace que, desde el punto de vista lógico, toda red basada en ella sea de un anillo. Debe notarse que un anillo lógico no implica un anillo físico. En efecto, si bien IEEE 802.5 emplea un anillo físico, IEEE 802.4 especifica un bus y ARCnet usa una estrella.
Por la red circulan dos tipos de mensajes: los "tokens" y los "frames". Un Token indica que la red está disponible. Además incluye información de prioridad, de forma tal que el control de la red lo pueda tomar sólo una estación con igual o mayor prioridad. Hay un timer que asegura que ninguna estación lo retenga demasiado tiempo. Un frame (trama) es un mensaje que contiene, entre otras cosas, la información que se quiere transmitir; las direcciones de las estaciones transmisora y receptora, aparte de un CRC para manejo de errores.

1.8.3 Comparación entre CSMA/CD y Token passing:

CSMA/CD	TOKEN PASSING
◆ PERMITE MAYOR PERFOMANCE CUANDO HAY POCAS COLISIONES.	◆ ASEGURA QUE UNA MÁQUINA PODRÁ TRANSMITIR.
◆ EN LA MAYORÍA DE LAS TRANSMISIONES SE ORIGINAN EN LA MISMA MÁQUINA	◆ TENIENDO UN TIEMPO DETERMINADO.
◆ DE IGUAL MANERA CUANDO EXISTE POCO TRÁFICO EN LA RED.	◆ NO IMPORTA CUANTO TRÁFICO HAYA EN LA RED.
	◆ NO SE VE AFECTADO EL PERFOMANCE DE LA RED POR EL TRÁFICO
	◆ CUENTA CON UN SISTEMA DE CONTROL QUE CONSTA DE ASEGURAR QUE UN MENSAJE LLEGUE A SU DESTINO ANTES DE QUE PASE CIERTO TIEMPO.
	◆ SOPORTA UN ESQUEMA DE PRIORIDADES PARA EL USO DE LA RED.

Tabla 1.2 diferencia entre csma/cd y token passing

1.9 El modelo OSI.

Una de las necesidades más apremiantes de un sistema de comunicaciones es el establecimiento de estándares, sin ellos sólo podrían comunicarse entre si equipos del mismo fabricante y que usaran la misma tecnología.
La conexión entre equipos electrónicos se ha ido estandarizando paulatinamente siendo las redes telefónicas las pioneras en este campo. Por ejemplo la histórica CCITT definió los estándares de telefonía: PSTN, PSDN e ISDN.
Otros organismos internacionales que generan normas relativas a las telecomunicaciones son: ITU-TSS (antes CCITT), ANSI, IEEE e ISO
La ISO[40] ha generado una gran variedad de estándares, siendo uno de ellos la norma ISO-7494 que define el modelo OSI, este modelo nos ayudará a comprender mejor el funcionamiento de las redes de computadoras.
El modelo OSI no garantiza la comunicación entre equipos pero pone las bases para una mejor estructuración de los protocolos de comunicación. Tampoco existe ningún sistema de comunicaciones que los siga estrictamente, siendo la familia de protocolos TCP/IP la que más se acerca.
El modelo OSI describe siete niveles para facilitar las interfaces de conexión entre sistemas abiertos.

[40] International Organisation for Standarisation

Fig. 1.13 Capas del modelo OSI

7.-Capa de Aplicación: Define el protocolo de las aplicaciones que trabajan sobre la red, además en este el sistema operativo de red y sus aplicaciones se hacen disponibles a los usuarios. Los usuarios emiten órdenes para requerir los servicios de la red.

6.- Capa de Presentación: Proporciona un mecanismo de negociación de los formatos de representación (conocidos como sintaxis de transferencia) para un determinado contenido del mensaje.

Este nivel elimina los problemas que puedan surgir al comunicar distintas arquitecturas, pues cada arquitectura estructura los datos de una forma específica, que no tienen por que ser compatibles. En el nivel de transporte se traducen los datos a un formato común, que se define en este mismo nivel.

En caso de ser necesario, también se encarga de la compresión y del cifrado (mal llamado encriptado).

Dispositivo utilizado: Pasarela.

5.- Capa de Sesión: Establece la comunicación entre las aplicaciones, la mantiene y la finaliza en el momento adecuado. Proporciona los pasos necesarios para entrar en un sistema utilizando otro. Permite a un mismo usuario, realizar y mantener diferentes conexiones a la vez (sesiones).

Dispositivo usado: Pasarela.

4.- Capa de Transporte: La función de este nivel es asegurar que el receptor reciba exactamente la misma información que ha querido enviar el emisor, y a veces asegura al emisor que el receptor ha recibido la información que le ha sido enviada. Envía de nuevo lo que no haya llegado correctamente.

Ejemplos de protocolos ISO son: TP0, TP1, TP2, TP3 y TP4. Y ejemplos de protocolos para Internet son: TCP y UDP.

Dispositivos usados: Pasarela (gateway).

3.- Capa de Red.- Establece las comunicaciones y determina el camino que tomarán los datos en la red.

Dispositivos usados: Encaminador (router).

Ejemplos de protocolos: IP, IPX.

2.- Capa de Enlace: Describe la forma de transportar de manera fiable los bits desde un nodo a otro en una red conmutada. Define conceptos tales como tramas, detección y corrección de errores y control de flujo.

En redes de difusión, además de lo antes mencionado, se encarga del control de acceso al medio compartido.

En redes de conmutación, además del control de flujo, controla el establecimiento mantenimiento y liberación de la conexión en cada uno de los enlaces. Asegura que el bit transmitido llegue de un nodo a otro, o del nodo al terminal (o viceversa). Es decir, garantiza un salto sin errores.

Ejemplos de protocolos son: HDLC, LAPB, LLC, LAPD, ALOHA[41] , CSMA, CSMA/CD y Paso testigo.
Dispositivos usados: Puentes (bridges).
1.- Capa Física: Define las características físicas del sistema de cableado, abarca también los métodos de red disponibles, incluyendo Token Ring, Ethernet y ArcNet. Este nivel especifica lo siguiente:
Conexiones eléctricas y físicas.
Como se convierte en un flujo de bits la información que ha sido paquetizada.
Como consigue el acceso al cable la tarjeta de red.
Un ejemplo de protocolo es el EIA RS-232[42], que define la utilización de los puertos serie de las computadoras.

Dispositivos usados: Cables, tarjetas y repetidores (hub).

En el modelo OSI el propósito de cada capa es proveer los servicios para la siguiente capa superior, resguardando la capa de los detalles de como los servicios son implementados realmente. Las capas son abstraídas de tal manera que cada capa cree que se está comunicando con la capa asociada en la otra computadora, cuando realmente cada capa se comunica sólo con las capas adyacentes de la misma computadora.

[41] detecta colisiones mediante técnica detección de error
[42] La Electronic Industries Association (EIA) ha formulado una norma para la conexión entre un equipo terminal de tratamiento de datos y un equipo de finalización de circuito de datos Esta norma está definida por el documento EIA-232-D sucesor del famoso EIA RS-232-C y se le denomina mas comúnmente RS-232.

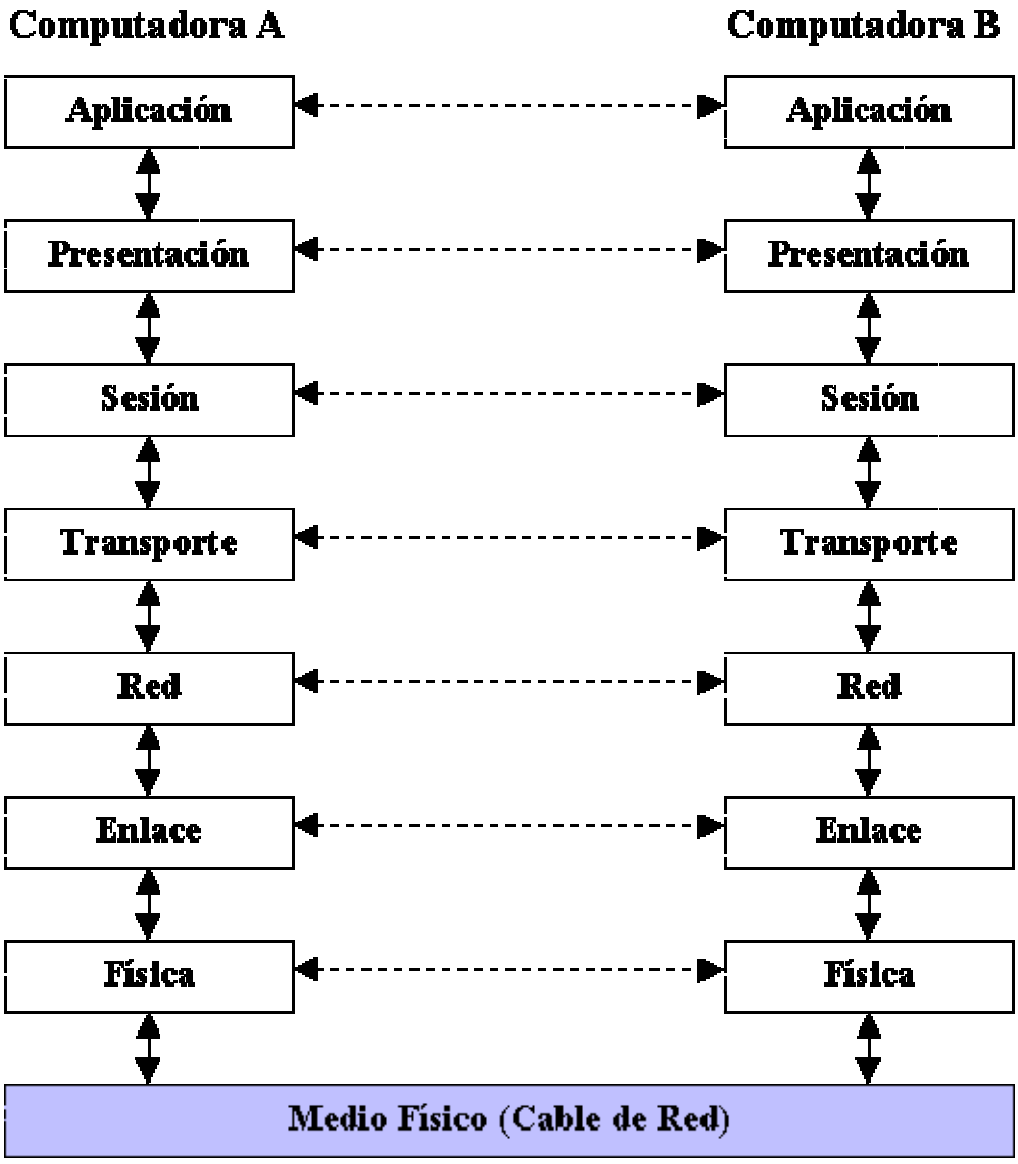


Fig. 1.14 Funcionamiento capas modelo OSI

Con esta última figura se puede apreciar que a excepción de la capa más baja del modelo OSI, ninguna capa puede pasar información directamente a su contraparte en la otra computadora. La información que envía una computadora debe de pasar por todas las capas inferiores, La información entonces se mueve a través del cable de red hacia la computadora que recibe y hacia arriba a través de las capas de esta misma computadora hasta que llega al mismo nivel de la capa que envió la información. Por ejemplo, si la capa de red envía información desde la computadora A, esta información se mueve hacia abajo a través de las capas de Enlace y Física del lado que envía, pasa por el cable de red, y sube por las capas de Física y Enlace del lado de el receptor hasta llegar a la capa de red de la computadora B.

La interacción entre las diferentes capas adyacentes se llama interface. La interface define que servicios la capa inferior ofrece a su capa superior y como esos servicios son accesados. Además, cada capa en una computadora actúa

como si estuviera comunicándose directamente con la misma capa de la otra computadora. La serie de las reglas que se usan para la comunicación entre las capas se llama *protocolo*

1.10 ¿Qué es TCP/IP?

Cuando se habla de TCP/IP, se relaciona automáticamente como el protocolo sobre el que funciona la red Internet. Esto, en cierta forma es cierto, ya que se le llama TCP/IP, a la familia de protocolos que nos permite estar conectados a la red Internet. Este nombre viene dado por los dos protocolos principales de esta familia:

- El protocolo TCP, funciona en el nivel de transporte del modelo de referencia OSI, proporcionando un transporte fiable de datos.

- El protocolo IP, funciona en el nivel de red del modelo OSI, que nos permite encaminar nuestros datos hacia otras maquinas.

Pero un protocolo de comunicaciones debe solucionar una serie de problemas relacionados con la comunicación entre computadoras, además de los que proporciona los protocolos TCP e IP.

1.10.1 Arquitectura de protocolos TCP/IP

Para poder solucionar los problemas que van ligados a la comunicación de computadoras dentro de la red Internet, se tienen que tener en cuenta una serie de particularidades sobre las que ha sido diseñada TCP/IP:

- Los programas de aplicación no tienen conocimiento del hardware que se utilizara para realizar la comunicación (módem, tarjeta de red...)

- La comunicación no esta orientada a la conexión de dos maquinas, eso quiere decir que cada paquete de información es independiente, y puede viajar por caminos diferentes entre dos maquinas.

- La interfaz de usuario debe ser independiente del sistema, así los programas no necesitan saber sobre que tipo de red trabajan.

- El uso de la red no impone ninguna topología en especial (distribución de los distintos ordenadores).

De esta forma, podremos decir, que dos redes están interconectadas, si hay una maquina común que pase información de una red a otra. Además, también podremos decir que una red Internet virtual realizara conexiones entre redes, que ha cambio de pertenecer a la gran red, colaboraran en el trafico de información procedente de una red cualquiera, que necesite de ella para acceder a una red remota. Todo esto independiente de las maquinas que implementen estas funciones, y de los sistemas operativos que estas utilicen.

1.10.2 Descomposición en niveles de TCP/IP.

Toda arquitectura de protocolos se descompone en una serie de niveles, usando como referencia el modelo OSI. Esto se hace para poder dividir el problema global en subproblemas de más fácil solución.
Al diferencia de OSI, formado por una torre de siete niveles, TCP/IP se descompone en cinco niveles, cuatro niveles software y un nivel hardware. A continuación describiremos los niveles, los cuales tienen cierto paralelismo con el modelo OSI.

Nivel de aplicación

Constituye el nivel mas alto de la torre TCP/IP. A diferencia del modelo OSI, se trata de un nivel simple en el que se encuentran las aplicaciones que acceden a servicios disponibles a través de Internet; corresponde con los niveles OSI al de aplicación, presentación y sesión.
Estos servicios están sustentados por una serie de protocolos que los proporcionan. Por ejemplo, tenemos el protocolo FTP[43], que proporciona los servicios necesarios para la transferencia de ficheros entre dos computadoras. Otro servicio, sin el cual no se concibe Internet, es el de correo electrónico, sustentado por el protocolo SMTP[44], el de conexión remota (TELNET) y otros más recientes como el protocolo HTTP[45].

Nivel de transporte

Este nivel coincide con el nivel de transporte del modelo OSI. Proporcionando una comunicación extremo a extremo entre programas de aplicación. La maquina remota recibe exactamente lo mismo que le envió la maquina origen. En este nivel el emisor divide la información que recibe del nivel de aplicación en paquetes, le añade los datos necesarios para el control de flujo y control de errores, y se los pasa al nivel de red junto con la dirección de destino.
En el receptor este nivel se encarga de ordenar y unir las tramas para generar de nuevo la información original.
Para implementar el nivel de transporte se utilizan dos protocolos:

- **UDP:** proporciona un nivel de transporte no fiable de datagramas, ya que apenas añade información al paquete que envía al nivel inferior, solo la necesaria para la comunicación extremo a extremo. Lo utilizan aplicaciones como NFS[46] y RPC[47], pero sobre todo se emplea en tareas de control.

- **TCP (Transport Control Protocolo):** es el protocolo que proporciona un transporte fiable de flujo de bits entre aplicaciones. Esta pensado para poder enviar grandes cantidades de información de forma fiable,

[43] File Transfer Protocol
[44] Simple Mail Transfer Protocol
[45] *Hypertext Transfer Protocol.*
[46] Network File System.
[47] Remote Procedure Call Llamada a Procedimiento Remoto.

liberando al programador de aplicaciones de la dificultad de gestionar la fiabilidad de la conexión (retransmisiones, perdidas de paquete, orden en que llegan los paquetes, duplicados de paquetes,...) que gestiona el propio protocolo. Pero la complejidad de la gestión de la fiabilidad tiene un coste en eficiencia, ya que para llevar a cabo las gestiones anteriores se tiene que añadir bastante información a los paquetes a enviar. Debido a que los paquetes a enviar tienen un tamaño máximo, como mas información añada el protocolo para su gestión, menos información que proviene de la aplicación podrá contener ese paquete. Por eso, cuando es mas importante la velocidad que la fiabilidad, se utiliza UDP, en cambio TCP asegura la recepción en destino de la información a transmitir.

Nivel de red

También precisamente coincide con el nivel de red del modelo OSI, y recibe el nombre de **nivel Internet**. Coloca la información que le pasa el nivel de transporte en datagramas IP, le añade cabeceras necesaria para su nivel y lo envía al nivel inferior. Es en este nivel donde se emplea el algoritmo de encaminamiento, al recibir un datagrama del nivel inferior decide, en función de su dirección, si debe procesarlo y pasarlo al nivel superior, o bien encaminarlo hacia otra maquina. Para implementar este nivel se utilizan los siguientes protocolos:

- **IP (Internet Protocol):** es un protocolo no orientado a la conexión, con mensajes de un tamaño máximo. Cada datagrama se gestiona de forma independiente, por lo que dos datagramas pueden utilizar diferentes caminos para llegar al mismo destino, provocando que lleguen en diferente orden o bien duplicados. Es un protocolo no fiable, eso quiere decir que no corrige los anteriores problemas, ni tampoco informa de ellos. Este protocolo recibe información del nivel superior y le añade la información necesaria para su gestión (direcciones IP , checksum)

- **ICMP (Internet Control Message Protocol):** proporciona un mecanismo de comunicación de información de control y de errores entre maquinas intermedias por las que viajaran los paquetes de datos. Esto datagramas los suelen emplear las maquinas (gateways, host,...) para informarse de condiciones especiales en la red, como la existencia de una congestión, la existencia de errores y las posibles peticiones de cambios de ruta. Los mensajes de ICMP están encapsulados en datagramas IP.

- **IGMP (Internet Group Management Protocol):** este protocolo esta íntimamente ligado a IP. Se emplea en maquinas que emplean IP multicast. El IP multicast es una variante de IP que permite emplear datagramas con múltiples destinatarios.

También en este nivel tenemos una serie de protocolos que se encargan de la resolución de direcciones:

- **ARP (Address Resolution Protocol):** cuando una maquina desea ponerse en contacto con otra conoce su dirección IP, entonces necesita un mecanismo dinámico que permite conocer su dirección física. Entonces envía una petición ARP por broadcast (o sea a todas las maquinas). El protocolo establece que solo contestara a la petición, si esta lleva su dirección IP. Por lo tanto solo contestara la maquina que corresponde a la dirección IP buscada, con un mensaje que incluya la dirección física. El software de comunicaciones debe mantener una cache con los pares IP-dirección física. De este modo la siguiente vez que hay que hacer una transmisión a es dirección IP, ya conoceremos la dirección física.

- **RARP (Reverse Address Resolution Protocol):** a veces el problema es al revés, o sea, una máquina solo conoce su dirección física, y desea conocer su dirección lógica. Esto ocurre, por ejemplo, cuando se accede a Internet con una dirección diferente, en el caso de PC que acceden por módem a Internet, y se le asigna una dirección diferente de las que tiene el proveedor sin utilizar. Para solucionar esto se envía por broadcast una petición RARP con su dirección física, para que un servidor pueda darle su correspondencia IP.

- **BOOTP (Bootstrap Protocol):** el protocolo RARP resuelve el problema de la resolución inversa de direcciones, pero para que pueda ser más eficiente, enviando más información que meramente la dirección IP, se ha creado el protocolo BOOTP. Este además de la dirección IP del solicitante, proporciona información adicional, facilitando la movilidad y el mantenimiento de las maquinas.

Nivel de enlace

Equivalente a los niveles OSI de enlace y nivel físico, este nivel se limita a recibir datagramas del nivel superior (nivel de red) y transmitirlo al hardware de la red. Pueden usarse diversos protocolos: DLC[48](IEEE 802.2), Frame Relay, X.25, etc.

La interconexión de diferentes redes genera una red virtual en la que las maquinas se identifican mediante una dirección de red lógica. Sin embargo a la hora de transmitir información por un medio físico se envía y se recibe información de direcciones físicas. Un diseño eficiente implica que una dirección lógica sea independiente de una dirección física, por lo tanto es necesario un mecanismo que relacione las direcciones lógicas con las direcciones físicas. De esta forma podremos cambiar nuestra dirección lógica IP conservando el mismo hardware, del mismo modo podremos cambiar una tarjeta de red, la cual contiene una dirección física, sin tener que cambiar nuestra dirección lógica IP.

[48] Data link control, se usa para mainframes

1.10.3 IP (Internet Protocol) versión 4.

El IP es un protocolo que pertenece al nivel de red, por lo tanto, es utilizado por los protocolos del nivel de transporte como TCP para encaminar los datos hacia su destino. IP tiene únicamente la misión de encaminar el datagrama, sin comprobar la integridad de la información que contiene. Para ello se utiliza una nueva cabecera que se antepone al datagrama que se está tratando. Suponiendo que el protocolo TCP ha sido el encargado de manejar el datagrama antes de pasarlo al IP, la estructura del mensaje una vez tratado quedaría así:

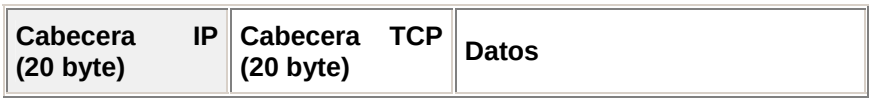


Fig. 1.15 funcionamiento de TCP/IP

La cabecera IP tiene un tamaño de 160 bit y está formada por varios campos de distinto significado. Estos campos son:

- **Versión:** Número de versión del protocolo IP utilizado. Tendrá que tener el valor 4. *Tamaño: 4 bit.*
- **Longitud de la cabecera:** (*Internet Header Length, IHL*) Especifica la longitud de la cabecera expresada en el número de grupos de 32 bit que contiene. *Tamaño: 4 bit.*
- **Tipo de servicio:** El tipo o calidad de servicio se utiliza para indicar la prioridad o importancia de los datos que se envían, lo que condicionará la forma en que éstos serán tratados durante la transmisión. *Tamaño: 8 bit.*
- **Longitud total:** Es la longitud en bytes del datagrama completo, incluyendo la cabecera y los datos. Como este campo utiliza 16 bit, el tamaño máximo del datagrama no podrá superar los 65.535 bytes, aunque en la práctica este valor será mucho más pequeño. *Tamaño: 16 bit.*
- **Identificación:** Valor de identificación que se utiliza para facilitar el ensamblaje de los fragmentos del datagrama. *Tamaño: 16 bit.*
- **Flags:** Indicadores utilizados en la fragmentación. *Tamaño: 3 bit.*
- **Fragmentación:** Contiene un valor (*offset*) para poder ensamblar los datagramas que se hayan fragmentado. Está expresado en número de grupos de 8 bytes (64 bit), comenzando con el valor cero para el primer fragmento. *Tamaño: 16 bit.*
- **Límite de existencia:** Contiene un número que disminuye cada vez que el paquete pasa por un sistema. Si este número llega a cero, el paquete será descartado. Esto es necesario por razones de seguridad para evitar un bucle infinito, ya que aunque es bastante improbable que esto suceda en una red correctamente diseñada, no debe descuidarse esta posibilidad. *Tamaño: 8 bit.*
- **Protocolo:** El número utilizado en este campo sirve para indicar a qué protocolo pertenece el datagrama que se encuentra a continuación de la cabecera IP, de manera que pueda ser tratado correctamente cuando llegue a su destino. *Tamaño: 8 bit.*

- **Comprobación:** El campo de comprobación (*checksum*) es necesario para verificar que los datos contenidos en la cabecera IP son correctos. Por razones de eficiencia este campo no puede utilizarse para comprobar los datos incluidos a continuación, sino que estos datos de usuario se comprobarán posteriormente a partir del campo de comprobación de la cabecera siguiente, y que corresponde al nivel de transporte. Este campo debe calcularse de nuevo cuando cambia alguna opción de la cabecera, como puede ser el límite de existencia. *Tamaño: 16 bit.*
- **Dirección de origen:** Contiene la dirección del *host* que envía el paquete. *Tamaño: 32 bit.*
- **Dirección de destino:** Esta dirección es la del *host* que recibirá la información. Los *routers* o *gateways* intermedios deben conocerla para dirigir correctamente el paquete. *Tamaño: 32 bits.*

Organización de la cabecera IP.

Versión	IHL	Tipo de servicio	Longitud total
Identificación		Flags	Fragmentación
Límite de existencia	Protocolo	Comprobación	
Dirección de origen			
Dirección de destino			

Fig. 1.16 datagrama

1.10.4 Direcciones IP y máscaras de red

En una red TCP/IP los ordenadores se identifican mediante un número que se denomina **dirección IP**. Esta dirección ha de estar dentro del rango de direcciones asignadas al organismo o empresa a la que pertenece, estos rangos son concedidos por un organismo central de Internet, el **NIC** (Network Information Center).
Una dirección IP está formada por 32 bits, que se agrupan en octetos:
01000001 00001010 00000010 00000011
Para entendernos mejor utilizamos las direcciones IP en formato decimal, representando el valor decimal de cada octeto y separando con puntos:
129.10.2.3
La dirección de una máquina se compone de dos partes cuya longitud puede variar:

- **Bits de red:** son los bits que definen la red a la que pertenece el equipo.
- **Bits de host:** son los bits que distinguen a un equipo de otro dentro de una red.

Los bits de red siempre están a la izquierda y los de host a la derecha, veamos un ejemplo sencillo:

Bits de Red	Bits de Host
10010110 11010110 10001101	11000101
150.214.141.	197

Para ir entrando en calor diremos también que esta máquina pertenece a la red 150.214.141.0 y que su máscara de red es 255.255.255.0. Si se quiere ir reflexionando sobre algo mostramos de nuevo en formato binario la máscara de red llevando acabo a la dirección de la máquina:

10010110	11010110	10001101	11000101
11111111	11111111	11111111	00000000

La máscara de red es un número con el formato de una dirección IP que nos sirve para distinguir cuando una máquina determinada pertenece a una subred dada, con lo que podemos averiguar si dos máquinas están o no en la misma subred IP. En formato binario todas las máscaras de red tienen los "1" agrupados a la izquierda y los "0" a la derecha.
Para llegar a comprender como funciona todo esto podríamos hacer un ejercicio práctico.

Ejercicio 1

Sea la dirección de una subred 150.214.141.0, con una máscara de red 255.255.255.0
Comprobar cuales de estas direcciones pertenecen a dicha red:
150.214.141.32
150.214.141.138
150.214.142.23

Paso 1: para ver si son o no direcciones validas de dicha subred clase C tenemos que descomponerlas a nivel binario:

150.214.141.32 10010110.1101010.10001101.10000000
150.214.141.138 10010110.1101010.10001101.10001010
150.214.142.23 10010110.1101010.10001110.00010111
255.255.255.0 11111111.1111111.11111111.00000000
150.214.141.0
10010110.1101010.10001101.00000000

Pasó 2: una vez tenemos todos los datos a binario pasamos a recordar el operador lógico AND o multiplicación:

Valor A	Valor B	Resultado
0	0	0
0	1	0
1	0	0
1	1	1

Vamos a explicar como hace la comprobación el equipo conectado a una red local.
Primero comprueba la dirección IP con su máscara de red, para ello hace un AND bit a bit de todos los dígitos:

150.214.141.32 10010110.1101010.10001101.10000000
255.255.255.0 11111111.1111111.11111111.00000000

_____

150.214.**141**.0 10010110.1101010.10001101.00000000

Luego hace la misma operación con la dirección IP destino.
150.214.141.138 10010110.1101010.10001101.10001010
255.255.255.0 11111111.1111111.11111111.00000000

_____

150.214.**141**.0 10010110.1101010.10001101.00000000
El resultado que obtenemos ambas veces es la dirección de red, esto no indica que los dos equipos están dentro de la misma red.

Paso3: vamos ha hacerlo con la otra dirección IP
150.214.142.23 10010110.1101010.10001110.00010111
255.255.255.0 11111111.1111111.11111111.00000000

_____

150.214.142.0 10010110.1101010.10001110.00000000
Como vemos este resultado nos indica que dicho equipo no pertenece a la red sino que es de otra red en este caso la red sería 150.214.142.0.

Ejercicio 2

Pasamos ahora a complicar un poco más la cosa. Como hemos leído antes la dirección IP se compone de dos partes la dirección de red y la dirección de host (máquina o PC). Imaginemos que en nuestra red solo hace falta 128 equipos y no 254 la solución sería dividir la red en dos partes iguales de 128 equipos cada una.
Primero cogemos la máscara de red.
Dirección de red Dirección de host.

_____._____._____._____
255.255.255.0 11111111.11111111.11111111.00000000

Si lo que queremos es crear dos subredes de 128 en este caso tenemos que coger un bit de la parte de identificativa del host.
Por lo que la máscara de re quedaría de esta manera.

Dirección de red Dirección de host.

_____.____.____.x.____
255.255.255.128 11111111.11111111.11111111.10000000

Donde X es el bit que hemos cogido para dicha construcción. Por lo que el último octeto tendría el valor 10000000 que es 128 en decimal.
Si la dirección de red que hemos utilizado es la 150.214.141.0 al poner esta máscara de red tendríamos dos subredes.
La 150.214.141.0 y la 150.214.141.128 que tendrían los siguientes rangos IP:
La 150.214.141.0 cogería desde la 150.214.141.1 hasta la 150.214.141.127
La 150.214.141.128 sería pues desde la 150.214.141.128 hasta la 150.214.141.254.
La máscara de red para las dos subredes sería la 255.255.255.128.

Comprobar.
Sea la máscara de red 255.255.255.128
La dirección de red 150.214.141.128
Comprobar si las siguientes direcciones pertenecen a dicha subred.

150.214.141.134
150.214.141.192
150.214.141.38
150.214.141.94

Si hemos realizado el ejercicio se tiene que comprobar que:
150.214.141.134 150.214.141.192 pertenecen a la subred 150.214.141.128
150.214.141.38 150.214.141.94 pertenecen a la subred 150.214.141.0

1.11 Clases de red

Para una mejor organización en el reparto de rangos las redes se han agrupado en cuatro clases, de manera que según el tamaño de la red se optará por un tipo u otro.

Las direcciones de clase A

Corresponden a redes que pueden direccionar hasta 16.777.214 máquinas cada una.
Las direcciones de red de clase A tienen siempre el primer bit a 0.
0 + Red (7 bits) + Máquina (24 bits)
Solo existen 124 direcciones de red de clase A.

Ejemplo:

	Red	Máquina		
Binario	0 0001010	00001111	00010000	00001011
Decimal	10	15	16	11

Rangos (notación decimal):
1. xxx.xxx.xxx - 126.xxx.xxx.xxx

Las direcciones de clase B

Las direcciones de red de clase B permiten direccionar 65.534 máquinas cada una.
Los dos primeros bits de una dirección de red de clase B son siempre 01.
01 + Red (14 bits) + Máquina (16 bits)
Existen 16.382 direcciones de red de clase B.
Ejemplo:

	Red		Máquina	
Binario	10 000001	00001010	00000010	00000011
Decimal	129	10	2	3

Rangos (notación decimal):
128.001. xxx.xxx - 191.254.xxx.xxx

Las direcciones de clase C

Las direcciones de clase C permiten direccionar 254 máquinas.
Las direcciones de clase C empiezan con los bits 110
110 + Red (21 bits) + Máquina (8 bits)
Existen 2.097.152 direcciones de red de clase C.

Ejemplo:

	Red			Máquina
Binario	110 01010	00001111	00010111	00001011
Decimal	202	15	23	11

Rangos (notación decimal):
192.000.001. xxx - 223.255.254.xxx

Las direcciones de clase D

Las direcciones de clase D son un grupo especial que se utiliza para dirigirse a grupos de máquinas. Estas direcciones son muy poco utilizadas. Los cuatro primeros bits de una dirección de clase D son 1110.

Direcciones de red reservadas

Existen una serie de direcciones IP con significados especiales.

- Direcciones de subredes reservadas:

000.xxx.xxx.xxx (1)
127. xxx.xxx.xxx (reservada como la propia máquina)
128.000. xxx.xxx (1)
191.255. xxx.xxx (2)
192.168. xxx.xxx (reservada para intranets)
223.255.255. xxx (2)

- Direcciones de máquinas reservadas:

xxx.000.000.000 (1)
xxx.255.255.255 (2)
xxx.xxx.000.000 (1)
xxx.xxx.255.255 (2)
xxx.xxx.xxx.000 (1)
xxx.xxx.xxx.255 (2)

1. Se utilizan para identificar a la red.
2. Se usa para enmascarar.

Internet Protocol (versión 6).

La nueva versión del protocolo IP recibe el nombre de IPv6, aunque es también conocido comúnmente como IPng (*Internet Protocol Next Generation*). El número de versión de este protocolo es el 6 frente a la versión 4 utilizada hasta entonces, puesto que la versión 5 no pasó de la fase experimental. Los cambios que se introducen en esta nueva versión son muchos y de gran importancia, aunque la transición desde la versión 4 no debería ser problemática gracias a las características de compatibilidad que se han incluido en el protocolo. IPng se ha diseñado para solucionar todos los problemas que surgen con la versión anterior, y además ofrecer soporte a las nuevas redes de alto rendimiento (como ATM, Gigabit Ethernet, etc.)
Una de las características más llamativas es el nuevo sistema de direcciones, en el cual se pasa de los 32 a los 128 bit, eliminando todas las restricciones del sistema actual. Otro de los aspectos mejorados es la seguridad, que en la versión anterior constituía uno de los mayores problemas. Además, el nuevo

formato de la cabecera se ha organizado de una manera más efectiva, permitiendo que las opciones se sitúen en extensiones separadas de la cabecera principal.

Formato de la cabecera.

El tamaño de la cabecera que el protocolo IPv6 añade a los datos es de 320 bit, el doble que en la versión 4. Sin embargo, esta nueva cabecera se ha simplificado con respecto a la anterior. Algunos campos se han retirado de la misma, mientras que otros se han convertido en opcionales por medio de las extensiones. De esta manera los *routers* no tienen que procesar parte de la información de la cabecera, lo que permite aumentar de rendimiento en la transmisión. El formato completo de la cabecera sin las extensiones es el siguiente:

- **Versión:** Número de versión del protocolo IP, que en este caso contendrá el valor 6. *Tamaño: 4 bit.*
- **Prioridad:** Contiene el valor de la prioridad o importancia del paquete que se está enviando con respecto a otros paquetes provenientes de la misma fuente. *Tamaño: 4 bit.*
- **Etiqueta de flujo:** Campo que se utiliza para indicar que el paquete requiere un tratamiento especial por parte de los *routers* que lo soporten. *Tamaño: 24 bit.*
- **Longitud:** Es la longitud en bytes de los datos que se encuentran a continuación de la cabecera. *Tamaño: 16 bit.*
- **Siguiente cabecera:** Se utiliza para indicar el protocolo al que corresponde la cabecera que se sitúa a continuación de la actual. El valor de este campo es el mismo que el de protocolo en la versión 4 de IP. *Tamaño: 8 bit.*
- **Límite de existencia:** Tiene el mismo propósito que el campo de la versión 4, y es un valor que disminuye en una unidad cada vez que el paquete pasa por un nodo. *Tamaño: 8 bit.*
- **Dirección de origen:** El número de dirección del *host* que envía el paquete. Su longitud es cuatro veces mayor que en la versión 4. *Tamaño: 128 bit.*
- **Dirección de destino:** Número de dirección de destino, aunque puede no coincidir con la dirección del *host* final en algunos casos. Su longitud es cuatro veces mayor que en la versión 4 del protocolo IP. *Tamaño: 128 bit.*

Organización de la cabecera IPv6.

Versión	Prioridad	Etiqueta de flujo	
Longitud		Siguiente cabecera	Límite de existencia
Dirección de origen			
Dirección de destino			

Fig. 1.17 datgrama IP v6

Las extensiones que permite añadir esta versión del protocolo se sitúan inmediatamente después de la cabecera normal, y antes de la cabecera que incluye el protocolo de nivel de transporte. Los datos situados en cabeceras opcionales se procesan sólo cuando el mensaje llega a su destino final, lo que supone una mejora en el rendimiento. Otra ventaja adicional es que el tamaño de la cabecera no está limitado a un valor fijo de bytes como ocurría en la versión 4.
Por razones de eficiencia, las extensiones de la cabecera siempre tienen un tamaño múltiplo de 8 bytes. Actualmente se encuentran definidas extensiones para *routing* extendido, fragmentación y ensamblaje, seguridad, confidencialidad de datos, etc.

Direcciones en la versión 6.

El sistema de direcciones es uno de los cambios más importantes que afectan a la versión 6 del protocolo IP, donde se han pasado de los 32 a los 128 bit (cuatro veces mayor). Estas nuevas direcciones identifican a un interfaz o conjunto de interfaces y no a un nodo, aunque como cada interfaz pertenece a un nodo, es posible referirse a éstos a través de su interfaz.
El número de direcciones diferentes que pueden utilizarse con 128 bits es enorme. Teóricamente serían 2^{128} direcciones posibles, siempre que no apliquemos algún formato u organización a estas direcciones. Este número es extremadamente alto, pudiendo llegar a soportar más de 665.000 **trillones** de direcciones distintas por cada **metro cuadrado** de la superficie del planeta Tierra. Según diversas fuentes consultadas, estos números una vez organizados de forma práctica y jerárquica quedarían reducidos en el peor de los casos a 1.564 direcciones por cada metro cuadrado, y siendo optimistas se podrían alcanzar entre los tres y cuatro trillones.
Existen tres tipos básicos de direcciones IPng según se utilicen para identificar a un interfaz en concreto o a un grupo de interfaces. Los bits de mayor peso de los que componen la dirección IPng son los que permiten distinguir el tipo de dirección, empleándose un número variable de bits para cada caso. Estos tres tipos de direcciones son:

- **Direcciones *unicast:*** Son las direcciones dirigidas a un único interfaz de la red. Las direcciones *unicast* que se encuentran definidas actualmente están divididas en varios grupos. Dentro de este tipo de direcciones se encuentra también un formato especial que facilita la compatibilidad con las direcciones de la versión 4 del protocolo IP.
- **Direcciones *anycast:*** Identifican a un conjunto de interfaces de la red. El paquete se enviará a un interfaz cualquiera de las que forman parte del conjunto. Estas direcciones son en realidad direcciones *unicast* que se encuentran asignadas a varios interfaces, los cuales necesitan ser configurados de manera especial. El formato es el mismo que el de las direcciones *unicast*.
- **Direcciones *multicast:*** Este tipo de direcciones identifica a un conjunto de interfaces de la red, de manera que el paquete es enviado a cada una de ellos individualmente.

Las direcciones de *broadcast* no están implementadas en esta versión del protocolo, debido a que esta misma función puede realizarse ahora mediante el uso de las direcciones *multicast*.

Sistema de nombres por dominio.

El sistema de nombres por dominio (DNS, *Domain Name System*) es una forma alternativa de identificar a una máquina conectada a Internet. La dirección IP resulta difícil de memorizar, siendo su uso más adecuado para los ordenadores. El sistema de nombres por dominio es el utilizado normalmente por las personas para referirse a un ordenador en la red, ya que además puede proporcionar una idea del propósito o la localización del mismo.

El nombre por dominio de un ordenador se representa de forma jerárquica con varios nombres separados por puntos (generalmente 3 ó 4, aunque no hay límite). Típicamente el nombre situado a la izquierda identifica al *host*, el siguiente es el subdominio al que pertenece este *host*, y a la derecha estará el dominio de mayor nivel que contiene a los otros subdominios:

nombre_ordenador.subdominio.dominio_principal

Aunque esta situación es la más común, el nombre por dominio es bastante flexible, permitiendo no sólo la identificación de *hosts* sino que también puede utilizarse para referirse a determinados servicios proporcionados por un ordenador o para identificar a un usuario dentro del mismo sistema. Es el caso de la dirección de correo electrónico, donde el nombre por dominio adquiere gran importancia puesto que el número IP no es suficiente para identificar al usuario dentro de un ordenador.

Para que una máquina pueda establecer conexión con otra es necesario que conozca su número IP, por lo tanto, el nombre por dominio debe ser convertido a su correspondiente dirección a través de la correspondiente base de datos. En los inicios de Internet esta base de datos era pequeña de manera que cada sistema podía tener su propia lista con los nombres y las direcciones de los otros ordenadores de la red, pero actualmente esto sería

impensable. Con esta finalidad se utilizan los servidores de nombres por dominio (DNS *servers*).

Los servidores de nombres por dominio son sistemas que contienen bases de datos con el nombre y la dirección de otros sistemas en la red de una forma encadenada o jerárquica.

Para comprender mejor el proceso supongamos que un usuario suministra el nombre por dominio de un sistema en la red a su ordenador local, realizándose el siguiente proceso:

- El ordenador local entra en contacto con el servidor de nombres que tiene asignado, esperando obtener la dirección que corresponde al nombre que ha suministrado el usuario.
- El servidor de nombres local puede conocer la dirección que se está solicitando, entregándosela al ordenador que realizó la petición.
- Si el servidor de nombres local no conoce la dirección, ésta se solicitará al servidor de nombres que esté en el dominio más apropiado. Si éste tampoco tiene la dirección, llamará al siguiente servidor DNS, y así sucesivamente.
- Cuando el servidor DNS local ha conseguido la dirección, ésta se entrega al ordenador que realizó la petición.

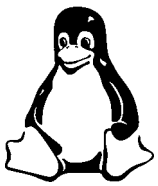
Si el nombre por dominio no se ha podido obtener, se enviará de regreso el correspondiente mensaje de error.

En el presente capitulo nos dimos a la tarea de estudiar los conceptos de una red ya que hasta las pequeñas empresas posenn una y es importante que sepamos como es que funciona como lo que seria los estandares y modelos con los que se rige y ademas tener un conocimiento mas solido sobre los elementos que conformar una red.

Asi como tambien conocer las distintas topologias, y tener un conocimiento de los protocolos usados enana red y tener un conocimiento mas firma sobre lo que inplica y que hay dentro de una red.

En el siguiente capitulo nos concentraremos en que es Linux y como este “sistema operativo” no compite con Windows porque es superior, porque en este mismo sitema podemos administrar nuestra red desde agregar usuarios, grupos y como un usuario llamado root se encarga de esta administración.

II. LINUX



A mediados de 1991 un estudiante finlandés llamado Linus Torvalds trabajaba en el diseño de un sistema operativo similar a Minix, que pudiera ejecutarse sobre plataformas Intel y compatibles, y sobre todo que fuera pequeño y barato; a raíz de un mensaje de este estudiante en comp.os.minix, algunas personas comenzaron a interesarse por el proyecto, y finalmente el 5 de octubre de ese año Linus Torvals hizo pública la versión 0.02 la primera funcional de lo que ya se denominaba Linux (Linus Unix). En esa versión, que aproximadamente utilizaron un centenar de usuarios, apenas se ofrecía soporte a hardware (excepto el que Linus tenía en su ordenador), no disponía de subsistema de red ni de sistema de ficheros propio, y las utilidades de espacio de usuario se podían contar con los dedos de las manos (un shell, un compilador, y poco más). Sin embargo, y a pesar de las duras críticas de pesos pesados en el mundo de los sistemas operativos como Andrew Tanenbaum, el proyecto era muy interesante, y poco a poco programadores de todo el mundo fueron aportando mejoras a este nuevo sistema.

A principios de 1994 apareció Linux 1.0, considerada la primera versión del operativo utilizable no sólo por hackers y programadores, sino por usuarios “normales”; de las aproximadamente 10000 líneas de la versión inicial se había pasado a unas 170000, y el centenar de usuarios se había multiplicado por mil. Linux 1.0 incorporaba subsistema de red (sin duda uno de los cambios que más ha contribuido a la expansión del operativo), entorno gráfico (arrastrado de versiones anteriores) y soporte a una gama de hardware relativamente amplia. La popularidad del operativo crecía mes a mes especialmente en entornos universitarios y de investigación gracias sobre todo a su filosofía: cualquiera podía (y puede) modificar una parte del núcleo, adaptarla, mejorarla, o incorporar nuevas líneas, con la única obligación de compartir el nuevo código fuente con el resto del mundo.

Sin embargo, no fue hasta 1996, con la aparición de Linux 2.0 (que incorporaba casi medio millón de líneas de código), cuando se produjo el gran estallido de Linux que perdura hasta la actualidad.

Esta nueva versión convertía a Linux en un sistema libre que, en algunos aspectos, no tenía nada que envidiar a entornos Unix comerciales; más de un millón de usuarios contribuía sin descanso a mejorar el sistema, y quizás por primera vez la arquitectura PC no era un mercado reservado casi en exclusiva a Microsoft. Muchas personas vieron que Linux podía llegar a ser rentable (a pesar de su filosofía de ser ”gratis”), y se comenzó a trabajar mucho en la facilidad de instalación y manejo para usuarios sin elevados conocimientos de informática; incluso llegaba a desbancar en muchas ocasiones al inamovible Minix a la hora de estudiar diseño de sistemas operativos en las universidades (algo poco comprensible, por otra parte, ya que cualquiera que hubiera tenido contacto con el código del kernel de Linux podría comprobar que a diferencia de Minix no está diseñado para ser legible y didáctico, sino para ser rápido).

En la actualidad Linux cuenta con varios millones de usuarios, y se ha convertido en el Unix más amistosota el usuario de todos los existentes, ya que no hacen falta conocimientos avanzados para instalarlo y manejarlo mínimamente; reconoce multitud de hardware (algo que siempre ayuda en el mercado de los ordenadores de sobremesa), y se puede utilizar para funciones tan diversas como servidores Web, de bases de datos, de correo electrónico, o como una sencilla Workstation. En muchas empresas medianas y pequeñas ha desplazado por completo a los sistemas Unix comerciales (caros y que generalmente corren sobre hardware que tampoco es barato), e incluso en grandes servidores se utiliza Linux como sistema operativo; aunque -esto es una crítica, por si no queda claro - en algunas ocasiones se echan de menos mecanismos de seguridad que sí están disponibles en otros Unix, podemos decir que Linux proporciona un nivel de seguridad, fiabilidad y estabilidad adecuado a la mayor parte de aplicaciones genéricas que nos podamos imaginar (es decir, no es un operativo apto para controlar una central nuclear, pero sí para cualquier aplicación de criticidad baja o media que podamos utilizar día a día).

2.1 ¿Qué es Linux?

Para poder definir lo que es Linux, es necesario explicar primero *¿Que es un sistema operativo?* Es un conjunto de programas principales y secundarios que administran las operaciones del hardware y del software de la computadora y que permiten interactuar con al usuario con el; entre las funciones básicas de cualquier sistema operativo se encuentran la administración de los trabajos, de las tareas, de los datos, de los dispositivos conectados a la computadora y la seguridad del propio sistema. Se puede afirmar que Linux es un sistema operativo. Que es algo así como el alma de la computadora, sin la cual los programas simplemente no podrían funcionar, y nosotros tampoco podríamos ordenarle operación alguna.
Sin embargo se distingue de los demás sistemas operativos por su precio y su potencia. Ya que Linux no se vende, no es un producto con el cual se comercie; esta disponible sin costo en Internet desde sus primeras versiones de forma completa (es decir con código fuente). En cuanto a la potencia esta va más allá de lo que ofrecen muchos otros sistemas operativos, y cuenta con el apoyo técnico de varios miles de programadores y usuarios en todo el mundo.

Podemos concluir que Linux es un sistema operativo gratuito de 32 o 64 bits para redes, similar a Unix, con código abierto, optimizado para Internet (utilizado por los piratas con mucha frecuencia) que puede funcionar en distintos tipos de hardware, incluyendo los procesadores Intel (x86) o Risc[49].

[49] (Reduced Instruction Set Computer). Se trata de un tipo de procesador especialmente rápido que utiliza una tecnología del tipo pipeline muy desarrollada, lo que le faculta para operar con un alto nivel de simultaneidad.

2.2 ¿Linux es gratuito y libre?

Es gratuito[50] porque puede obtenerse sin coste algúno. Por ejemplo, no es necesario adquirir un libro de Linux que contenga CD-ROM para obtenerlo. Ya que si se dispone de un acceso telefónico rápido, es posible descargarlo de Internet e instalarlo.
Si se compara con otros sistemas operativos, se aprecia claramente que la mayoría de los distribuidores exigen que se abone cada instalación, lo que significa que cada vez que se instala un sistema operativo, es necesario pagar un precio adicional. Por consiguiente, si se cuenta con diversas estaciones de trabajo hay que pagar varias licencias. Al contrario Linux puede instalarse en otras estaciones de trabajo y sin pagar ni un solo peso.

Es libre en el sentido de que ofrece una impresionante libertad técnica. Cuando se descarga Linux, se obtiene algo más que el simple sistema operativo. Se obtiene el código fuente, por lo que si no les gusta cómo funciona, tiene la posibilidad de modificarlo (y no solo a pequeña escala, si no que es posible hacerlo en general para adecuarlo a las necesidades personales).
Además Linux cuenta con muchos lenguajes de programación, compiladores y herramientas de desarrollo asociadas como son:

✓	**ADA**	✓	**Python[51]**
✓	**BASIC**	✓	**Lenguajes de shell (csh, bash).**
✓	**C**	✓	**TCL/TK[52]**
✓	**C++**	✓	**Perl**
✓	**Expect[53]**	✓	**GTK[54]**
✓	**FORTRAN**	✓	**PASCAL**

Tabla 2.1 lenguajes y herramientas de LINUX

Con la licencia general publica de GNU[55], puede utilizar estas herramientas para desarrollar y vender aplicaciones de Linux sin pagar derechos de comercialización. Sin embargo, si realiza algún cambio en las bibliotecas GPL[56], también debe realizarlos de forma gratuita en la GPL en turno.

[50] algunas aplicaciones para Linux de terceros no son gratuitas y sus creadores imponen restricciones de licencia.
[51] Un lenguaje scripts orientado a objetos.
[52] Un lenguaje scripts y un conjunto de herramientas con interfaz grafica de usuarios respectivamente
[53] Un lenguaje de script para automatizar sesiones de red.
[54] El cual es un conjunto de herramientas para crear aplicaciones GUI en Linux
[55] (GNU no se trata de Unix) Se crea en 1984 sin mucho éxito, bajo la filosofía del software libre, muy al estilo de UNIX.

[56] La licencia GPL o General Public License. Puedes instalar y usar un programa GPL en un computadora o en tantas como te apetezca, sin limitación. También puedes modificar el programa para adaptarlo a lo que tú quieras que haga. Además, podrás distribuir el programa GPL tal cual o después de haberlo modificado. Puedes hacer esto, regalando el programa o vendiéndolo, tu única obligación, es facilitar siempre con el programa binario el código fuente, es decir, el programa de forma que pueda ser leído por un programador.

La mayor libertad sigue siendo su código abierto, que proporciona importantes ventajas en lo que a seguridad se refiere. Cuando se utilizan sistemas operativos comerciales, el destino del usuario esta en manos del creador. Si el código tiene defectos, nunca se sabrá (y si se averigua es posible que ya sea tarde, porque el sistema ya estará en peligro).
En el caso de Linux cualquiera puede examinar el código para ver la implementación del sistema de seguridad con lo que surge un tema impetuosamente debatido: las personas que critican Linux insisten en que para beneficiarse de todas las ventajas de libertad técnica de Linux, es necesario tener unos conocimientos técnicos orientados mucho mayores que los que se necesitan en cualquier sistema operativo orientados al consumidor, lo que es cierto.

Existen algunas herramientas de seguridad de Linux que son realmente conjuntos de herramientas con un gran número de módulos de seguridad independientes. Si dichos conjuntos de herramientas se utilizan correctamente de forma combinada, proporcionan una gran flexibilidad para idear e implantar soluciones de seguridad personalizadas. A cambio de esta eficacia, hay que olvidarse de la sencillez de la computación de señalar y hacer clic. Por consiguiente, hay que reconocer que para establecer un host Linux seguro, es necesario invertir tiempo y esfuerzo.

2.3 Características de Linux

Sistema operativo multitarea: Linux fue diseñado completamente como un sistema operativo multitarea por lo que puede administrar y realizar dos o mas procesos o tareas de forma simultánea; estas tareas o procesos los controla con base a su propia eficiencia (versión del kernel), en la cantidad de memoria RAM disponible en la computadora, en la velocidad de la CPU, en la capacidad y velocidad del disco duro.
Sistema operativo multiusuario: Linux es un sistema operativo capaz de responder a las solicitudes de varios usuarios que emplean una misma computadora, pero que tienen necesidades distintas. El sistema operativo lleva el control de sus actividades, les asigna espacio en disco duro a cada uno, les permite entrar a sus cuantas o permisos y les restringe el acceso a los diferentes programas, utilerías, documentos, espacios etc.
Sistema operativo multiplataforma: Linux es soportado por computadoras personales con procesadores 386, 486, Pentium, Pentium pro, Pentium II, Pentium III, Pentium 4, Amiga o Atary, pero además existen versiones para plataformas como alpha y PowerPC.
Sistema de archivos: tiene la capacidad de operar con diversos sistemas de archivos como la FAT[57] de DOS, la VFAT[58] de Windows 9X, la OS2/FS o la ISO9660.

[57] (File Allocation Table). Tabla de asignación de archivos. Parte del sistema de archivos del DOS y OS/2 que lleva un seguimiento de la ubicación de los datos almacenados en un disco. Cuando el disco se formatea a alto nivel, el FAT se registra dos veces y contiene una tabla con una entrada para cada cluster (conglomerado) en disco. La lista de directorios, que contiene el nombre del archivo, extensión, fecha, etc.,
[58] (Virtual File Allocation Table) Tabla virtual de asignación de archivos. Sistema de archivos utilizado en Windows para Workgroups y Windows 95. Provee acceso de alta velocidad en Modo Protegido de 32 bits para manipulación de archivos.

Red: Ha sido desarrollado como un sistema operativo para trabajo en red, cuyo protocolo principal es TCP/IP; actualmente soporta también los protocolos SLIP/PP, PLIP, NFS[59], Telnet, TNP, SMT, IPX, Apple Talk, etc., y puede trabajar con casi todas las tarjetas de red existentes en el mercado.
Sistema operativo de 32 bits reales: Linux corre a 32 bits reales en una computadora personal y a 64 en una alpha; su kernel opera en el modo protegido del procesador y sus librerías emplean el enlace dinámico, con lo que varios programas o utilerías pueden ocupar la misma librería sin esta deba ser cargada en memoria repetidamente sino un sola vez.
Entorno: otra característica de Linux es que puede trabajar sin conflicto tanto en modo texto como en entornos gráficos que emplean sistemas de ventanas estilo Windows. Estos gráficos lo constituyen FWVM, GNOME, KDE, CDE, Enlightenment, Nextlevel, etcétera. La variedad de entornos soportados depende del tipo de tarjeta de video y del propio monitor configurado durante la instalación.

2.4 Distribuciones de Linux

Linux es un sistema de libre distribución por lo que podemos encontrar todos los ficheros y programas necesarios para su funcionamiento en multitud de servidores conectados a Internet. La tarea de reunir todos los ficheros y programas necesarios, asi como instalarlos en tu sistema y configurarlo, puede ser una tarea bastante complicada y no apta para muchos. Por esto mismo, nacieron las llamadas distribuciones de Linux, empresas y organizaciones que se dedican a hacer el trabajo "mas sencillo" para nuestro beneficio y comodidad.

Una distribución no es otra cosa, que una recopilación de programas y ficheros, organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet, o comprando los CDs de las mismas, los cuales contendrán todo lo necesario para instalar un sistema Linux bastante completo y en la mayoría de los casos un programa de instalación que nos ayudara en la tarea de una primera instalación. Casi todos los principales distribuidores de Linux, ofrecen la posibilidad de bajarse sus distribuciones, via FTP (sin cargo alguno).

Existen muchas y variadas distribuciones creadas por diferentes empresas y organizaciones a unos precios bastantes accesibles (si se compran los CDs, en vez de bajársela via FTP), las cuales deberíamos poder encontrar en tiendas de informática, librerías. En el peor de los casos siempre podemos encargarlas directamente por Internet a las empresas y organizaciones que las crean. A veces, las revistas de informática sacan una edición bastante aceptable de alguna distribución.

A continuación tenemos una grafica con las distribuciones más comunes en la línea del tiempo.

[59] *(Network File System). Sistema de ficheros distribuido para un entorno de red de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales. Originalmente desarrollado por Sun Microsystems Inc.*

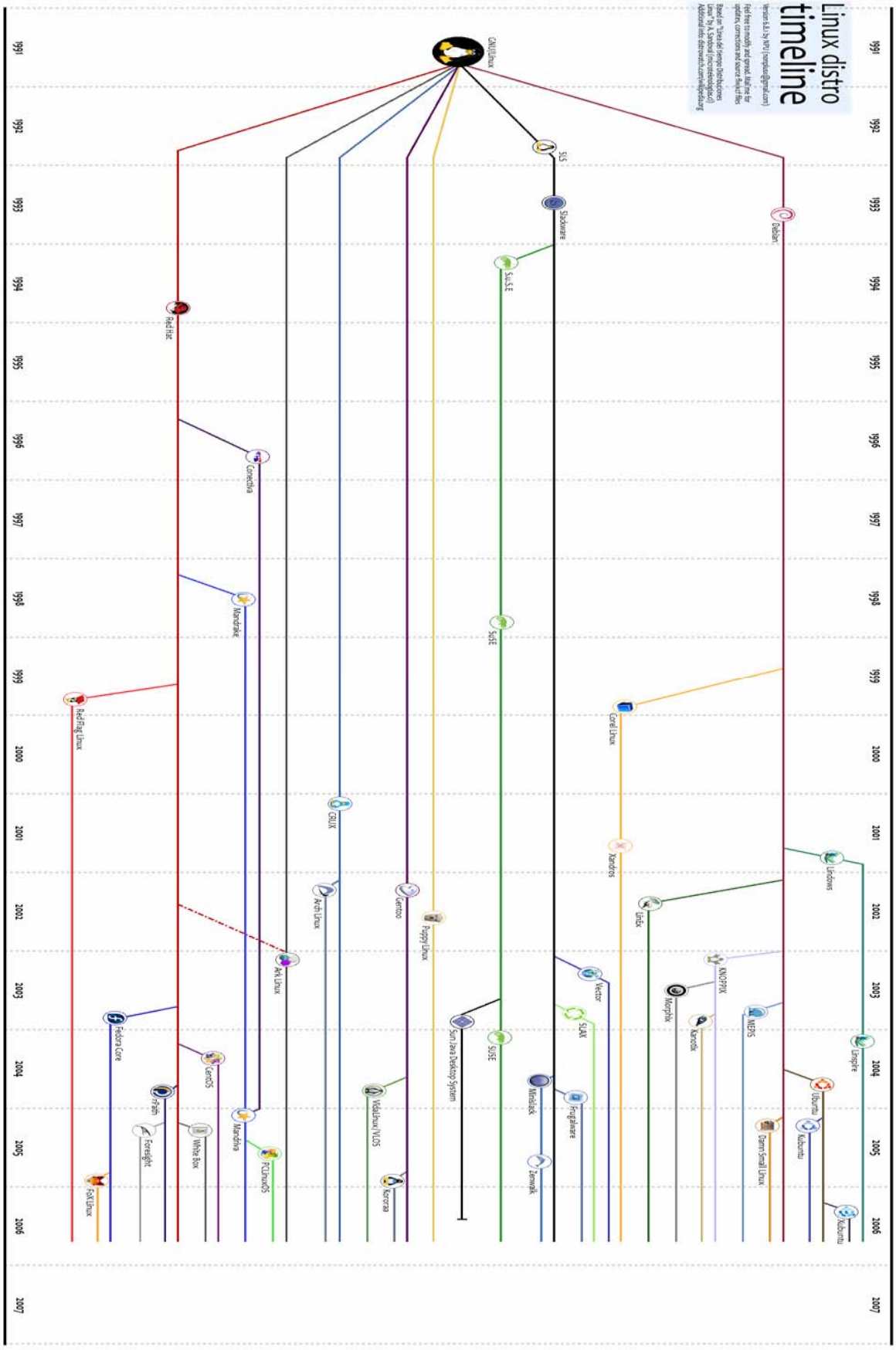


Fig 2.1 Evolucion de las distribuciones de Linux en el tiempo

Si vamos a instalar el sistema por primera vez, recomiendo hacerlo con una de estas distribuciones y en un futuro cuando se requiera actualizar el sistema con las últimas versiones y actualizaciones del núcleo y programas que se utilicen.

Si lo que queremos es probar una distribución Linux sin necesidad de instalarla, podemos probar una distribución LiveCD .

Un "LiveCD" no es otra cosa que una distribucion de Linux que funciona al 100%, sin necesidad de instalarla en el ordenador donde la probamos. Utiliza la memoria RAM del ordenador para 'instalar' y arrancar la distribucion en cuestion. En la memoria tambien se instala un "disco virtual" que emula al disco dure de un ordenador.

De esta forma solamente hace falta introducir el CD o DVD en el ordenador en cuestion y arrancarlo, al cabo de unos minutos tendremos un sistema Linux funcionando en el mismo. Este tipo de distribuciones solamente sirve para demostraciones y pruebas, ya que una vez que apagamos el ordenador, todo lo que hemos hecho desaparece.

Algunas distribuciones del tipo "LiveCD" vienen tambien con la opcion de instalacion una vez que la hemos probado. Existen muchas distribuciones de este tipo, algunas solamente en version "LiveCD", otras como demostraciones de distribuciones que se pueden instalar de la manera tradicional.

A continuación podemos encontrar información sobre las distribuciones más importantes de Linux (aunque no son las únicas).

DISTRIBUCIÓN REDHAT ENTERPRISE

Esta es una distribución que tiene muy buena calidad, contenidos y soporte a los usuarios por parte de la empresa que la distribuye. Es necesario el pago de una licencia de soporte. Enfocada a empresas.

- Pagina FTP principal: **ftp.redhat.com/pub/**
- **Pagina Web** de Red Hat http://www.redhat.com/

DISTRIBUCIÓN FEDORA

Esta es una distribución patrocinada por RedHat y soportada por la comunidad. Fácil de instalar.

- Pagina de descarga: **fedora.redhat.com/download/**
- **Pagina Web** de Fedora http://fedora.redhat.com/

DISTRIBUCIÓN DEBIAN

Otra distribución con muy buena calidad. El proceso de instalación es quizás un poco mas complicado, pero sin mayores problemas. Gran estabilidad antes que últimos avances.

- Pagina FTP principal: **ftp.debian.org/**
- **Pagina Web** de Debian http://www.es.debian.org/

DISTRIBUCIÓN S.u.S.E

Otra de las grandes. Calidad germana. Fácil de instalar.

- Pagina FTP principal: **ftp.suse.com**
- **Pagina Web** de S.u.S.E http://www.suse.de/es/

DISTRIBUCIÓN SLACKWARE

Esta distribución es de las primeras que existió. Tuvo un periodo en el cual no se actualizo muy a menudo, pero eso es historia. Es raro encontrar usuarios de los que empezaron en el mundo Linux hace tiempo, que no hayan tenido esta distribución instalada en su ordenador en algún momento.

- **Pagina Web** de Slackware http://www.slackware.com/

DISTRIBUCIÓN GENTOO

Esta distribución es una de las únicas que últimamente han incorporado un concepto totalmente nuevo en Linux. Es un sistema inspirado en BSD-ports. Podemos compilar/optimizar nuestro sistema completamente desde cero. No es recomendable adentrarse en esta distribución sin una buena conexión a Internet, una computadora medianamente potente (si queremos terminar de compilar en un tiempo prudencial) y cierta experiencia en sistemas Unix.

- **Pagina Web** de Gentoo http://www.gentoo.org/

DISTRIBUCIÓN UBUNTU

Distribución basada en Debian, con lo que esto conlleva y centrada en el usuario final y facilidad de uso.

- **Pagina Web** de Ubuntu http://www.ubuntu.com/

DISTRIBUCION UBUNTU - Version DESKTOP LIVECD

Distribucion basada en Debian, con lo que esto conlleva y centrada en el usuario final y facilidad de uso. La imagen ISO version DESKTOP de esta distribucion, es del tipo 'LiveCD' con posibilidades tambien de instalacion si queremos.

- **Pagina Web** de Ubuntu http://www.ubuntu.com/

DISTRIBUCION GNOPPIX - LIVECD

Esta distribucion esta basada en Ubuntu y usa Gnome como gestor de ventanas.

- **Pagina Web** de Gnoppix http://www.gnoppix.org/

DISTRIBUCION SuSE LIVE - Version LIVECD

Version LiveCD de la distribucion SuSE.

- **Pagina Web** de SuSE http://www.suse.com/

DISTRIBUCION KNOPPIX - LIVECD

Distribucion LiveCD basada en Debian.

- **Pagina Web** de Knoppix http://www.knopper.net/knoppix/index-en.html

DISTRIBUCION CENTOS - Version LIVECD

Version LiveCD de la distribucion Centos. Basada en Redhat Enterprise.

- **Pagina Web** de Centos http://www.centos.org

DISTRIBUCION GENTOO - Version LIVECD

Version LiveCD de la distribucion Gentoo.

- **Pagina Web** de Gentoo http://www.gentoo.org

DISTRIBUCION SLAX - LIVECD

Distribucion LiveCD basada en Slackware

- **Pagina Web** de Slax http://www.slax.org

2.5 El núcleo de Linux.

El kernel es Linux en si mismo. Todo lo que se encuentra en su entorno no son más que aparatos extras, adornos, con el fin de que nos resulte atractivo. Mantener el núcleo sano, conserva estable el sistema completo. En el área de la gestión del núcleo encontraremos una serie de cuestiones que normalmente se deben tener en cuenta. Algunas de ellas justifican un interés inmediato, mientras que otras constituyen consideraciones a largo plazo.

2.5.1 Partes del núcleo.

Básicamente conformada por dos componentes principalmente. El primer componente es la parte compilada, que constantemente se mantiene en la memoria. Esta parte debería consistir solo en aquellos elementos que son necesarios constantemente. El resto debería ir en el segundo componente del núcleo: los módulos del núcleo.
Los elementos que se utilizan de manera intermitente (algunos controladores de dispositivos) se colocan en los módulos del núcleo. Estas piezas pueden ser cargadas o descargadas durante la operación, asegurando que el propio núcleo no va a ocupar una gran parte de su RAM y sin embargo dispondrá de la funcionalidad completa del sistema.
Los módulos que se utilizan rara vez se pueden cargar manualmente cuando los necesitan y luego quitarlos cuando haya acabado.

2.5.2 Cuando personalizar el núcleo.

A veces se presentan situaciones concretas que hacen deseable o necesario, construir un núcleo de cliente. Entre estas situaciones podemos incluir:

- Instalar un programa requiere una característica de núcleo que no tiene activada.
- Agregar un nuevo dispositivo que utiliza continuamente y no quiere tener que cargar el modulo manualmente.

- Comprar un dispositivo nuevo que no esta disponible en su actual núcleo.
- Su actual núcleo es bastante antiguo y sabe que en un futuro próximo necesitara características que en el no estén disponibles, así que quiere instalar la versión mas reciente para ponerse al día.

2.5.3 Elegir un nuevo núcleo.

Hay varias cosas a tener en consideración cuando se elige un nuevo núcleo:

- Si se necesita soporte para los dispositivos nuevos ¿la nueva versión también lo soporta?
- ¿Soporta la nueva versión todos los dispositivos antiguos? A veces a una versión nueva le faltan unos cuantos elementos que están preparados en la siguiente.
- ¿El núcleo es estable?
- Si el nuevo dispositivo es soportado solamente por un núcleo experimental, ¿merece la pena arriesgar la estabilidad?

La preparación del núcleo funciona en dos rutas de desarrollo diferentes. El núcleo experimental es exactamente lo que parece, código que esta actualmente en desarrollo. Primero se comprueban nuevos controladores de dispositivos y nuevas características. Cuando se hayan añadido suficientes elementos nuevos el núcleo sólido experimental se convertirá en un núcleo de producción y comienza un nuevo proceso de núcleo experimental. El núcleo de producción es el que utilizan en sus máquinas la mayoría de los administradores de sistemas.

El nombre de un núcleo termina con una cadena de números con un formato X.Y.Z. cada numero se incrementa a medida que se añaden nuevas características y correcciones de fallos. X es el que cambia mas lentamente podrían pasar años antes de que haya núcleos que empiecen con 3, por ejemplo. Decidir cambiar el número de una versión principal es un acto subjetivo. Pasar de un núcleo de base 2 a tres, representaría un avance enorme. Por ejemplo, cuando los números de la versión saltaban del 1 al 2 se añadían nuevos elementos.

- Soporte para cuatro plataformas nuevas de arquitectura hardware (incluyendo Apple Power PC y Atari).
- Controladores de dispositivos IDE adicionales, mejor manejo de un error IDE, auto detección de algunas características IDE y la habilidad para manejar más controladores.
- Controladores de dispositivos SCSI adicionales.
- Lista de tarjeta Ethernet ampliamente expandida.
- La moderna estructura modular.
- Advanced Power Management (APM) para dispositivos como los ordenadores portátiles plegables.
- Construcción, enmascarado y capacidades adicionales de routing del IP.
- Soporte ISDN[60].

[60] (Red Digital de Servicios Integrados --RDSI) La gran ventaja de este tipo de tecnología es que permite que los datos, conexiones de voz, etc. viajen a través de un solo cable. Actualmente este servicio es ofrecido por las principales compañías telefónicas a nivel mundial y goza de gran aceptación.

- Soporte de sistemas de archivos VFAT y SMB[61], entre otros.
- Soporte de cuota del sistema de archivos.

El componente y cambia en base semiregular, quizás cada seis más o menos. Este es el dígito crucial a la hora de señalar si el núcleo es experimental o de producción. Los núcleos experimentales tienen un componente “Y” impar, por ejemplo, 2.3.12, 3.7.16, y 4.19.92. Por otra parte, los núcleos de producción tienen un componente “Y” par, como 2.2.12, 3.8.16 ó 4.20.92. De hecho, incluso podemos ver una versión con el siguiente formato X.Y.Z-A. Cuando veamos esto significa que esta versión fue especialmente alterada para su distribución Linux.

2.5.4 Documentación del núcleo.

El archivo 00-INDEX, contiene una lista con descripciones de una línea de todos los archivos de documentación. Esto es útil si descarga una versión más nueva de la que tiene. Otro archivo es el BUG-HUNTING, y contiene instrucciones de cómo buscar, encontrar e informar sobre un fallo en el núcleo. El archivo oops-tracking.txt, contiene información adicional sobre este punto.
El archivo Coding style. Es primordialmente, una guía de estilo para formatear el código del núcleo. Otro archivo importante que tiene que ver con esto es magic-number.txt que contiene una lista de los números mágicos de la estructura de datos que ya están en uso.
Para a los que les interesa como es que el núcleo maneja la organización de I/O, seria importante que vieran los archivos IO-mapping.txt. e ioctl-number.text. Si se quiere ejecutar una amplia gama de archivos binarios y hacerlo desde la línea de comandos, tendría que leer el archivo binfmt_misc.txt. para cuestiones especificas sobre Java, hay que leer java.txt también podemos descubrir cómo trata el núcleo los errores si leemos el archivo exception.txt, y como maneja el kernel las dependencias durante la configuración en smart-config.txt.
El archivo kernel-parameters.txt contiene los parámetros que acepta el núcleo al arrancar. Hay un documento que explica el funcionamiento del programa que gestiona los módulos del núcleo. Este archivo es kmod.txt. Otro que se ocupa de la información previa acerca de la habilidad del núcleo de Linux para cargar y descargar módulos de una forma dinámica es modules.txt. Para ver cómo el núcleo maneja el cierre de los archivos, lea el archivo locks.txt y el archivo de información previa relacionada, mandatory.txt. Los problemas de memoria se tratan con memory.txt.
En el archivo kernel-docs.txt, están disponibles los índices de una amplia diversidad de documentación del núcleo.

Arquitectura (Archivos específicos.)

Hay algunos archivos del núcleo que tratan específicamente con una forma u otra de arquitectura de hardware.

[61] Bloque de mensajes de servidor. Un protocolo de red usado por las redes de Microsoft® Windows® para acceder a sistemas de archivos de otras máquinas.

- ARM-README: Notas del equipo de desarrollo del código ARM, relacionados con cuestiones específicas que los usuarios ARM necesiten tener en cuenta.
- IO-APIC.txt: Útil para aquellos que tengan placas madre de competencia SMP[62] de arquitectura Intel o basada en Intel. No todas las placas madre son completamente adaptables al IO–APIC. Este archivo se centra en cómo decir si su placa madre es manejable y cómo tratar con los problemas.
- Arm: Este subdirectorio contiene archivos adicionales relacionados con la versión ARM de Linux.
- i386: Este subdirectorio contiene archivos adicionales relacionados con la versión i386 (Intel) de Linux.
- m68k: Este subdirectorio contiene archivos adicionales relacionados con la versión Macintosh 68k de Linux.
- Mca.txt: Discusión de la arquitectura microchannel i386 que se maneja bajo Linux.
- Mtrr.txt: Discusión del Memory Type Range Control en los sistemas Intel Pentium Pro y Pentium II.
- Powerpc: Este subdirectorio contiene archivos adicionales que se refieren a la versión Power PC de Macintosh de Linux.
- Sgi-visws.txt: Discusión del manejo de Linux de las estaciones de trabajo visuales de SGI.

Hardware. (Archivos específicos)

Hay una diferencia entre los archivos específicos de arquitectura y los archivos específicos de hardware: la arquitectura se refiere al tipo global de diseño del sistema; el hardware es mas como una etiqueta de dispositivos que una etiqueta de diseño. Los archivos entre las fuentes del núcleo que están relacionadas con dispositivos específicos son:

- README.DAC960: Información especifica sobre el controlador Miles DAC960/DAC110m PCI RAID.
- Cdrom: Este subdirectorio contiene archivos de información en un número de modelos y fabricantes de CD-ROM.
- computone.txt: Información especifica del controlador de serie computone Intelliport II/Plus Multiport.
- cpqarray.txt: información especifica acerca de los controladores Compaq´s SMART2 Intelligent Disk Array.
- Digiboard.txt: Información especifica del multimodem de hardware digiboard PC/Xi, PC/Xe y PC/Xeve.
- digiepca.txt: Más información perteneciente a la tecnología digiboard.
- fb: Este subdirectorio contiene archivos de información sobre los controladores de memoria intermedia y dispositivos. La memoria intermedia del marco esta relacionada con la manera en la que la maquina maneja los gráficos.

[62] abreviatura de *Symetric Multi-Processors*, designa la capacidad de los núcleos Linux 2.0 y versiones siguientes de funcionar sobre máquinas con varios procesadores.

- filesystems: Este subdirectorio contiene archivos de información sobre los diferentes tipos de sistemas de archivos que entienden el Linux.
- ftape.txt: Discusión referente al controlador cinta ftape.
- hayes-esp.txt: Información especifica para modems Hayes ESP (Enhanced Serial Port).
- ide.txt: Información relacionada con el manejo de los dispositivos IDE del núcleo de Linux.
- Isdn: Este subdirectorio contiene información sobre el manejo ISDN del núcleo y los diversos controladores disponibles, que incluyen información acerca del PPP sobre el ISDN.
- joystick-api.txt: Información sobre el código de escritura que utiliza las capacidades de joystick de Linux.
- joystick.txt: Información sobre los controladores de joystick de Linux.
- joystick-parport.txt: Información sobre el Linux y los joysticks de puerto paralelo.
- md.txt: Discusión sobre el manejo del dispositivo Multi Disk de Linux.
- mkdev.ida: Este es, en realidad un shell script que le ayuda a configurar información de dispositivos para controladores SMART RAID IDA.
- networking: Este subdirectorio contiene información sobre los diferentes controladores del dispositivo de tarjetas de red disponibles en Linux.
- paride.txt: Discusión de dispositivos de puerto paralelo de Linux e IDE.
- parport.txt: Discusión de los módulos y del soporte del puerto paralelo de Linux.
- pci.txt: Discusión sobre lo que se tiene que evitar al escribir controladores PCI.
- pcwd-watchdog.txt: discusión sobre la forma en que maneja Linux la tarjeta watchdog de ISA Berkshire PC.
- ramdisk.txt: este archivo no es exactamente especifico de hardware, pero si incluye un controlador: el controlador RAMdisk utilizado en Linux.
- risccom8.txt: Notas sobre el controlador serial de multipuerto RISCCom/8.
- rtc.txt: Discusión sobre el controlador de reloj en tiempo real utilizada en Linux para continuar con las arquitecturas PC y Alpha.
- scsi-generic.txt: Discusión sobre los controladores de Linux usados para manejar dispositivos SCSI[63] que no sean discos, cintas o CD-ROM.
- scsi.txt: Breve discusión sobre el soporte SCSI bajo Linux.

[63] Small Computer System Interface. Es un interface hardware de tipo serie para periféricos muy común.

- serial-console.txt: Deshacer el output de consola en distintos dispositivos.
- smp.tex,smp.txt: Discusión sobre como configurar el manejo del multiprocesador durante la configuración del núcleo.
- Sound: Este subdirectorio contiene información de cada uno de los controladores de sonido disponibles en Linux.
- specialix.txt: Alcance con el que cuenta el controlador de serie multipuerto Specialix 108+.
- stallion.txt: Alcance del controlador de serie multipuerto stallion.
- sx.txt: Alcance del controlador de serie de multipuerto Specialix SX/SI.
- video4linux: Este subdirectorio contiene información y Application Program Interfaces (API) por lo que se refiere a unidades de video para Linux.
- watchdog.txt: discusión de los controladores de interfaz del reloj automático Watchdog bajo Linux.

Archivos útiles extra.

- Changes: Este archivo contiene una lista de los sitios Web de los recursos del núcleo. Sin embargo, no es este el motivo por el que el archivo está en esta sección. La maravilla de este archivo changes, es que incluye una lista de la versión que necesita de todo el software que requiere el núcleo o la construcción del núcleo, y como determinar qué versión tiene en ese momento.
- Configure.help: Este archivo contiene una lista de todos y cada uno de los controladores y opciones que puede activar o desactivar durante el proceso de configuración del núcleo, junto con algún párrafo comentando lo que hace esta opción y quien podría, o no querer utilizarla. Si no estamos seguros de si se quiere o no implementar algo, este es el lugar preciso para obtener información.
- devices.txt, devices.text: Estos dos elementos son el mismo archivo. Uno esta en formato text que es el lenguaje de marcado para explicar documentos. El otro es un archivo de texto ASCII. El archivo de dispositivos contiene una lista de todos los archivos y números de dispositivos manejados por el núcleo.
- initrd.txt: A veces hay confusión acerca del disco RAM de arranque que puede crear cuando compila un núcleo nuevo. Este archivo se centra en el tipo de usos que tiene este disco RAM y en como llevarlos a la practica.
- kbuild: Este subdirectorio contiene archivos que tratan del proceso de construcción del núcleo.
- proc[64].txt: Este archivo contiene una discusión de la sección /proc del sistema de archivos y de lo que contiene.

[64]Es un mecanismo adicional para que el núcleo y los módulos del núcleo envíen información a los procesos el sistema de ficheros /proc. Originalmente diseñado para permitir un fácil acceso a la información sobre los procesos, es ahora usado por cada parte del núcleo que tiene algo interesante que informar.

- Sysctl: Este subdirectorio contiene archivos que hablan de los cambios que se hacen en los archivos /proc/sys/.
- sysrq.txt: en este archivo podremos aprender mas sobre las ligaduras clave y como enviar comandos directamente al núcleo por medio de ellas.

Widgets[65] diversos.

- VGA-softcursor.txt: Cómo modificar su cursor en el modo línea de comandos.
- logo.gif: El logo de Linux no oficial.
- logo.txt: Explicación del logo.gif.
- nbd.txt: Discusión de los controladores de dispositivos de bloque de red experimentales.
- nfsroot.txt: como montar el sistema de archivos root a tráves de un montaje NFS en un sistema sin disco.
- spinlocks.txt: discusión sobre como crear spinlocks eficaces en maquinas de multi y uniprocesador.
- svga.txt: Discusión de selección de modo de video del nivel del núcleo en maquinas 80X60.
- unicode.txt: Discusión acerca de como el núcleo utiliza unicode[66] para referirse a los caracteres.

Obtener la versión del núcleo y otra información del sistema.

El comando unname le permite obtener información del sistema eligiendo los parámetros apropiados.

Parámetro	Propósito
-a	Proporciona la información para todos los parámetros de una sola vez.
-m	Hace una lista de la arquitectura correspondiente a la máquina.
-n	Hace una lista del nombre hosts de la máquina.
-p	Proporciona el tipo de procesador con el que cuenta la máquina.
-r	Proporciona la versión del núcleo.
-s	Hace una lista del sistema operativo que esta siendo utilizado.
-v	Dice cuando fue instalado este sistema operativo

Tabla 2.2 comandos de unname

[65] Un *widget* es un elemento gráfico con el que el usuario puede interactuar

[66] *Superconjunto del conjunto de caracteres ASCII que utiliza dos bytes en lugar de uno para cada carácter. Unicode es capaz de manejar 65,536 combinaciones de caracteres en lugar de 256, y puede contener los alfabetos de la mayor parte de los lenguajes a nivel mundial. ISO define un conjunto de caracteres de cuatro bytes para alfabetos mundiales, pero también utiliza el Unicode como un subconjunto.*

2.5.5 Instalar un núcleo nuevo.

1. Adquiera la fuente del núcleo, bien ya sea desde Internet o desde un CD.
2. Instale la fuente. Si no estamos utilizando un RPM, desempaquetamos el tarball en usr/src.
3. Necesitamos asegurarnos de que también hay otros paquetes instalados. Esta es una lista de ellos, con los números de la versión de Red Hat 6.2 (solo para ejemplificar) incluidos para el nombre del archivo:
   - kernel-headers-2.2.14.50.i386.rpm: Contiene los archivos de cabecera para la fuente del núcleo.
   - cpp-1.1.2-30.i386.rpm: El paquete preprocesador C, utilizado para compilación.
   - egcs-1.1.2-30.i386.rpm: El Compilador C.
   - glibc-devel-2.1.3.-15.i386.rpm: Es el conjunto de librería de desarrollo de C.
   - make-3.77-6.i386.rpm: El gestor de construcción que utilizara para compilar la fuente.
   - ncurse-5.0-11.i386.rpm: necesario para controlar el cursor si escoge la opción menuconfig, para configurar los parámetros del núcleo.
   - ncurses-devel-5.0-11.i386.rpm: necesario para controlar el cursor si elige la opción menuconfig, parámetros del núcleo.
4. Cambiemos el directorio /usr/scr/Linux-version para el paquete. Para el Red Hat 6.2 es usr/scr/Linux-2.2.14.
5. Escribamos “make old config” para construir un archivo de configuración basado en las configuraciones de su núcleo actual. Este es un buen punto si no quiere arriesgarse a olvidarse de incluir cosas que ya había agregado.
6. Escoja la herramienta de configuración del núcleo que desea utilizar.
7. Una vez que haya elegido una herramienta de configuración del núcleo siga las instrucciones de cada uno.
8. Escriba “make[67] dep” para construir una lista de las dependencias de la fuente.
9. Escriba “make clean”, para eliminar los archivos temporales usados hasta el momento.
10. Escriba “make bzImage”, para compilar el núcleo base.
11. escriba “make modules”, para compilar los módulos que cargará el núcleo durante la operación.
12. El compilador coloca la fuente del núcleo en el directorio usr/scr/Linux/arch/i386/boot/bzImage. Copie este archivo en el directorio /boot.
13. Cambie el nombre /boot/bzImage por el de /boot/vmlinuz-version, con el comando vm, donde la versión es la versión del núcleo nuevo que esta a punto de instalar.
14. Copie el archivo /boot/System. map al /boot/System. map-old.
15. Copie el archivo/usr/src/linux/System.map, al directorio /boot.

[67] Recuerde que aunque cada uno de estos pasos make sea corto de leer, puede llevar varios minutos o más, dependiendo de la velocidad del CPU.

16. Escriba “make modules_install”, para instalar los módulos en /lib/modules/versión.
17. Aunque no es necesario, un buen consejo es crear una versión de arranque especial del núcleo, que contenga los módulos que necesita el hardware en el momento del arranque, por ejemplo si una unidad de disco primaria es SCSI[68]. Para hacerlo, utilice mkinitrd/boot/initrd-versión-versión.
18. No podríamos arrancar con el núcleo nuevo hasta que le diga al LILO donde encontrarlo.

Configurar el núcleo

Configurar el núcleo con Config.

La herramienta de configuración del núcleo config, esta completamente basada en texto. Lleva algún tiempo efectuar el proceso completo de configuración, ya que config obtiene la configuración del núcleo nuevo haciendo preguntas de una en una. Más que proporcionar una prueba preliminar de muestra, que es un proceso muy largo, advertimos aquí algunos elementos a los que se les debe prestar especial atención. Se dan en orden que podría encontrárselos, siendo la opción por defecto la que la encabeza, que es una letra mayúscula o una palabra escrita en mayúsculas:

1. Para iniciar esta herramienta de configuración, escriba “make config”. El código de compilación se desplaza y luego aparece la siguiente pregunta:

```
*Code maturity level options*[69]
  Prompt for development and/or incomplete code/drivers
  (CONFIG_EXPERIMENTAL) [Y/n/?]
```

2. Para un servidor de producción importante, recomendamos elegir “n”. Es mejor utilizar un código sólido y de confianza para este tipo de maquina. Así que, escriba “n” y pulse intro. Si desea responder “Y”, simplemente pulse intro, ya que la “Y” mayúscula denota la opción por defecto. Después, verá lo siguiente:

```
  *
  *Processor type and features
  *
  Processor                         family
(386,486/CX486, 586/K5/5x86/6x86, Pentium/K6/TSC, Ppro/6x86MX) [386]
```

[68] Small Computer System Interface. Es un interface hardware de tipo serie para periféricos muy común.

[69] Las líneas que comienzan con asteriscos son encabezamientos que muestra el programa config para etiquetar a qué pertenecen las preguntas. También las opciones de respuesta a veces cambian el orden. Se organizan en orden, de la mas probable a la menos probable.

3. Las respuestas validas son cada una de las entradas individuales, pero para las que están separadas por barras oblicuas, solamente necesitan escribir una de las partes entre las barras oblicuas. Por ejemplo, si la maquina tiene un procesador Pentium, Escribiríamos “Pentium” y pulse Intro. Si tiene un procesador 386, simplemente pulse intro porque este elemento se anota por defecto. Después, verá lo siguiente:

```
Maximum Physical Memory (1GB, 2GB) [1GB]
```

4. Como se puede apreciar aquí solo hay dos opciones disponibles. Si por alguna razón su maquina tiene entre 1 y 2 GB de RAM escriba “2GB” y pulse intro. Si no, acepte la opción por defecto pulsando intro. Después verá :

```
Math Emulation (CONFIG_MATH_EMULATION) [Y/n?]
```

5. Si tiene un procesador 486DX o más rápido, entonces escriba “n” y pulse Intro, porque la emulación matemática esta construida en procesadores de estas velocidades. Si no, pulse Intro para aceptar la opción por defecto. Después aparecerá:

```
MTRR (Memory Type Range Register) support (CONFIG_MTRR) [Y/n?]
```[70]

6. Esta opción es principalmente para uso de maquinas Intel P6 así como otras clasificadas en la línea 6 de Pentium. Escriba “n” y pulse intro si no tiene una maquina como esta, o simplemente pulse intro si la tiene. El siguiente elemento es:

```
 Symmetric multi-processing support (CONFIG_SMP) [Y/n/?]
```

7. Si dispone de multiprocesadores en la placa madre, entonces escriba “y” y pulse Intro. Si no, no necesita esta opción, así que pulse Intro para aceptar la opción por defecto. Lo siguiente es:

```
*
*Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n?]
```

8. Hay alguna situación en la que no desea esta opción. Sin ella, tiene que compilar todo directamente en el núcleo. Pulse Intro para aceptar la opción por defecto.
9. Si está instalando Linux en un ordenador portátil con características Advanced Power Management, la siguiente es una característica altamente útil. Si no, no la necesita:

```
Advanced Power Management BIOS Support (CONFIG_APM) [Y/n/?]
```[71]

[70] Cuando no este seguro de que contestar, seleccione el signo de interrogación. Esta respuesta le da una descripción de la opción del núcleo que está considerando, de modo que puede hacer una elección mas sensata

[71] hay muchos elementos que hacen referencia a marcas especificas del hardware o correcciones para ellos

10. Hay un número de opciones que se refiere al soporte Integrated Development Integrated (IDE). Si no hay dispositivos IDE[72] en su sistema, puede que no necesite este soporte.
11. Resulta útil tener los siguientes elementos activados o al menos modulares si se piensa utilizar RAID[73].

```
Multiple devices driver support (CONFIG_BLK_DEV_MD) [Y/n/?]
Autodetect RAID partitions (CONFIG_AUTODETECT_RAID) [Y/n/?]
Linear more (append) (CONFIG_MD_LINEAR) [M/y/n/?]
RAID-0 (striping) mode (CONFIG_MD_STRIPED) [M/y/n/?]
RAID-1 (mirroring) mode (CONFIG_MD_MIRRORING) [M/y/n/?]
RAID-4/RAID-5 mode (CONFIG_MD_RAID5) [M/y/n/?]
Mylex     DAC960/DAC1100     PCI     RAID     Controller     support
(CONFIG_BLK_DEV_DAC960) [M/y/n/?]
```

12. Características útiles para máquinas que van a actuar como cortafuegos de filtrado de paquetes son:

```
Network firewalls (CONFIG_FIREWALL) [Y/n/?]
UNIX domain sockets (CONFIG_UNIX) [Y/m/n/?]
IP: policy routing (CONFIG_IP_MULTIPLE_TABLES) [Y/n/?]
(New)
IP: firewalling (CONFIG_IP_FIREWALL) [Y/n/?]
IP:  firewall  packet  netlink  device  (CONFIG_IP_FIREWALL_NETLINK)
[Y/n/?]
IP: use FWMARK value as routing key (CONFIG_IP_ROUTE_FWMARK) [Y/n/?]
(nuevo)
```

13. Una máquina que vaya a utilizar como router debería tener seleccionados los siguientes elementos:

```
IP: avanced router (CONFIG_IP_ADVANCED_ROUTER) [Y/n/?]
IP: equal cost multipatch (CONFIG_IPROUTE_MULTIPATH) [Y/n/?] (New)
IP :use TOS value as routing key (CONFIG_IP_ROUTE_VERBOSE) [Y/n/?]
(New)
IP:  large  routing  tables  (CONFIG_  IP_ROUTE_LARGE_TABLES)  [Y/n/?]
(New)
IP:  fast  network  addres  traslation  (CONFIG_IP_ROUTE_NAT)  [Y/n/?]
(New)
IP: optimize as router not host (CONFIG_IP_ROUTER) [Y/n/?]
```

14. Las siguientes son algunas de las características útiles para un cortafuegos proxy:

```
IP: transparent proxy support (CONFIG_IP_TRANSPARENT_PROXY) [Y/n/?]
IP: masquerading (CONFIG_IP_MASQUERADE_ICMP) [Y/n/?]
IP: ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP) [Y/n/?]
```

15. En el supuesto de que quiera realizar un hosting virtual y asignar direcciones IP múltiples a la misma interfaz, entonces incluya lo que aparece a continuación:

```
IP: aliasing support (CONFIG_IP_ALIAS) [Y/n/?]
```

16. Lo que viene a continuación nos ayudara a protegernos de los ataques de negación de servicio:

[72] *Integrated Drive Electronics*
[73] *Matriz Redundante de Discos Independientes.*

```
SYN flood protection (CONFIG_SYN_ COOKIES) [Y/n/?]
```

Configurar el núcleo con el menuconfig.

Es un programa guiado por menú no-gráfico que le permite retroceder y avanzar por los diferentes tipos de opciones del núcleo y cambiar cosas hasta que el resultado final le resulte satisfactorio.

1. Escriba “make menuconfig” para ejecutar la herramienta. Parte del código del compilador se desplaza, y después se abre la herramienta en el terminal como se ve enseguida.

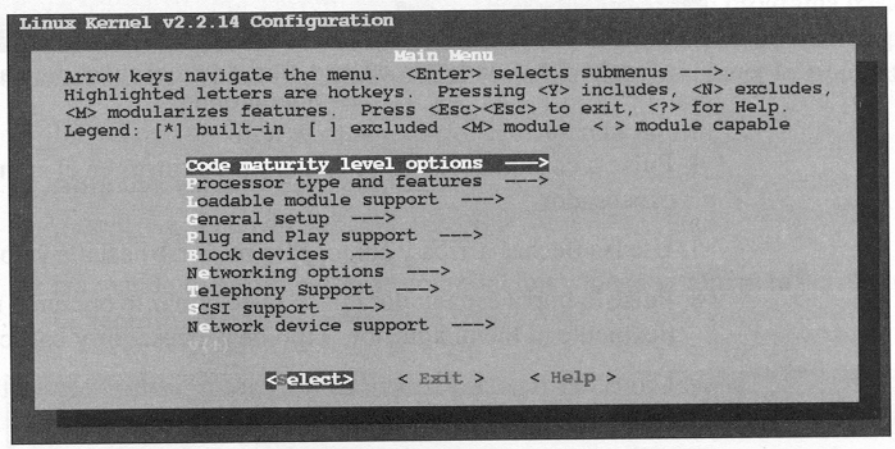


Fig. 2.2 configuración del núcleo con menú config

2. Para seleccionar el menú Tipo de procesador y características, utilice la flecha hacia abajo con el fin de descender a través del menú.
3. Para abrir el menú pulse Intro. Esto le traslada al cuadro de diálogo Tipo de procesador y características, que podemos ver en la siguiente figura. Algunos de los elementos son opciones del núcleo y algunos son submenús.

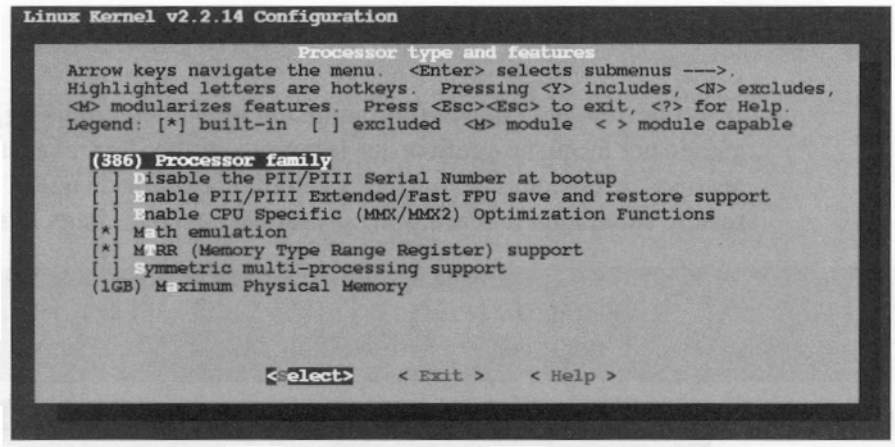


Fig. 2.3 submenu de menú config

4. Pulse a continuación Intro con el fin de entrar en el submenú Familia del procesador.
5. Use las flechas arriba y abajo, para moverse hasta la velocidad de su CPU.
6. Pulse la barra espaciadora para seleccionar la opción y regresar automáticamente al menú anterior, tipo de procesador y características.

7. Utilice las flechas arriba y abajo, para ir hasta el elemento de emulación matemática.
8. si la casilla esta seleccionada, lo que significa que contiene un asterisco, use la barra espaciadora para anular la selección, si la maquina es 486DX o más rápida.
9. Use las flechas arriba y abajo para ir a la opción de soporte MTRH (Memory Type Range Register).
10. Escriba un signo de interrogación, para lanzar la ayuda. Se abre un cuadro de dialogo que contiene una entrada que explica para qué sirve esta opción del núcleo. Esta información se extrae del archivo Configure.help.
11. Lea la entrada. Utilice las teclas de flecha para desplazarse por el texto.
12. Pulse Intro cuando termine de leer.
13. Utilice las flechas derecha e izquierda para trasladarse por los menús de la parte inferior, hasta la opción salir.
14. Pulse Intro, para regresar al menú principal menuconfig.
15. Utilice las flechas derecha o izquierda, para elegir la opción Salir.
16. Pulse salir, para salir de la herramienta.
17. Si desea guardar la configuración, lleve el cursor hasta Sí. Si no seleccione No.
18. Pulse Intro para finalizar.

Configurar el núcleo con Xconfig.

Es un programa basado en la GUI, que le permite utilizar el ratón para moverse por los diferentes grupos de opciones del núcleo. Para emplearla haga lo siguiente:

1. Introduzca la GUI, si es necesario, escribiendo “startx”.
2. Abra una ventana del terminal.
3. Escriba “/usr/src/Linux/makexocnfig”.
4. Vemos la información de compilación y luego se abre la herramienta xconfig.
5. Si es novato en la configuración del núcleo, es una buena opción recorres todas las opciones. Haga clic en la opción Controladores de CD-ROM antiguos (no SCSI, no IDE), que se muestran a continuación.

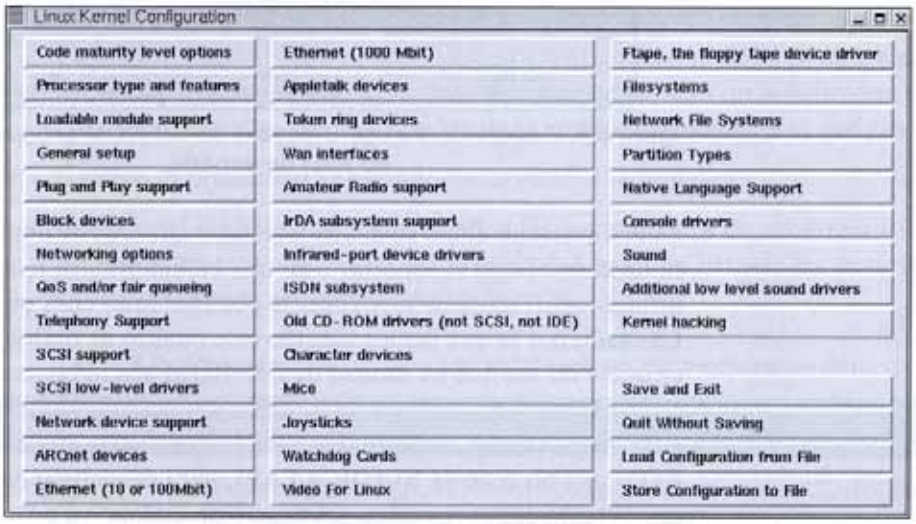


Fig. 2.4 configuración del núcleo con Xconfig

6. Es útil entender cómo se deben leer estos cuadros de diálogo. Si la primera línea menciona si soportar o no una característica, téngalo en cuenta en primer lugar, antes de preocuparse por cualquier cosa que venga después. La razón por la que hemos elegido este cuadro de diálogo como ejemplo, es que a no ser que su Unidad de CD-ROM sea lo bastante antigua como para no ser IDE, SCSI o cualquier otra relacionado con éstas, no lo necesita. Haga clic en la “n” que esta al lado de soporte de unidades de CD-ROM que no sean SCSI /IDE/ATAPI, para desactivar esta opción, fíjese que el resto de las opciones del cuadro de diálogo ya no están disponibles.
7. Haga clic en el botón menú principal, para cerrar el cuadro de diálogo.
8. Haga clic en el botón soporte de idioma nativo, para abrir el cuadro de diálogo Soporte de idioma nativo.
9. Haga clic en “n”, para cada uno de los idiomas para los que no necesita soporte.
10. Cuando termine, haga clic en el botón Menú principal.
11. Cuando haya terminado la configuración, haga clic en el botón Guardar y Salir, para guardar los cambios. Se abrirá un cuadro de dialogo.
12. lea el contenido del cuadro de diálogo, que le explica que acaba de completar la configuración de su núcleo y de cual es el siguiente comando en el proceso de compilación del núcleo.

Insertar módulos manualmente.

Para insertar un módulo de núcleo, haga lo siguiente:
1. Utilice el comando cd, para cambiar al directorio /lib/modules.
2. Observe el contenido del directorio con el comando ls. Si nunca antes ha agregado un núcleo nuevo, debería haber solo un subdirectorio. Este subdirectorio, lleva el nombre de la versión del núcleo a la que sirve.
3. Cambie al directorio apropiado de la versión del núcleo.
4. Localice el modulo que desea insertar. Hay un número de subdirectorios presentes en esta ubicación. En tabla siguiente hay una lista con una breve descripción de cada subdirectorio.

Directorio	Contenido
Block	Unidades para dispositivos de bloque (que no sean CD-ROM y SCSI, tienen su propia sección), que son medios de acceso aleatorio donde le mirar cualquier porción en cualquier momento, mejor que se forzado a secuencia. Un modo rápido de decir si está tratando con un dispositivo bloques es que el volumen de los datos es en bytes.
Cdrom	Controladores para una variedad de unidades de CD-ROM populares.
Fs	Módulos para tipos de sistemas de archivos soportados.
Ipv4	Módulos para servicios de la versión 4 de IP, como enmascarado de IP.

misc	Diversos módulos que no pertenecen expresamente a ninguno de los demás directorios.
modules.dep	Archivo que contiene una lista de los módulos que dependen de otros módulos. Este archivo lo genera el comando depmod.
Net	Módulos de unida de dispositivo de red, como controladores Ethernet.
pcmcia	Controladores de dispositivo PCMCIA.
Scsci	Controladores de dispositivos SCSI.
Video	Controladores de tarjeta de monitor y vídeo.

Tabla 2.3 descripción de subdirectorios

5. Cambie al subdirectorio apropiado.
6. Escriba luego “insmod módulo”, donde módulo es el nombre del modulo que quiere cargar.

Quitar módulos manualmente.

Para quitar un módulo del núcleo cargado, haga lo siguiente:

1. Hay que obtener una lista de los módulos que están en funcionamiento, para ver que módulos están cargados lo podemos averiguar con el comando Ismod o con /more/proc/modules.
2. Señale qué módulo desea cargar.
3. Escriba “rmmod módulo” para descargar el módulo. Si hay módulos activos que dependen de aquel que va a quitar, entonces aparecerá un error de dependencia.
4. Si aparece un error de dependencia, señale si necesita, o no, el módulo dependiente. Puede que necesite mantenerlo, lo que significa que debería conservar el módulo inicial que quería descargar. Sin embargo, puede que tampoco necesite el modulo dependiente.
5. Para quitar el modulo inicial además de sus módulos dependientes, escriba “rmmod-r módulo”.

2.5.6 Obtener información del sistema.

El directorio /proc y su contenido, se crea en RAM por el núcleo de Linux, para almacenar información actual del sistema. Para obtener información acerca de su caja Linux, sus procesos y muchos más, es bueno darle un vistazo a este archivo. En la tabla que a continuación se muestra nos da un panorama general de los subdirectorios que podemos encontrar en el /proc[74].

Directorio	Contenido
number	Cada directorio que tiene un número en su nombre corresponde al proceso que se está ejecutando en ese momento. En el se ofrecen los detalles sobre cómo se invoco ese proceso y de cuál es su estado al.
cpuinfo	Archivo que contiene información del CPU y de la arquitectura del ma.

[74] Si necesita información mas profunda, escribe “man proc”.

devices	Archivo con una lista de los grupos de dispositivos y de sus números Asignados.
dma	Lista de los canales directos de acceso (DMA) de memoria de la stry Estándar Architecture (ISA) que están en uso y por parte de que ento.
filesystems	Lista de los tipos de sistema de archivos que el núcleo debe entender.
interrupts	Número de interrupciones disponibles en cada configuración de petición de interrupción (IRQ), que incluye el dispositivo que utiliza cada IRQ.
ioports	Direcciones de puerto Hexadecimales en uso, por diferentes sitivos input/output (I/O).
Kcore	Archivo binario que contiene la información almacenada en RAM.
Kmsg	Archivo binario que contiene mensajes del núcleo, accesible con el comando dmesg.
ksyms	Información creada y utilizada para el núcleo para gestionar módulos.
loadavg	Media de carga bajo la que se encuentra el servidor, que se toma lo cuántos trabajos se estaban ejecutando hace 1 minuto, 5 minutos y 15 minutos.
meminfo	Cantidad de RAM libre y utilizada, espacio de intercambio y mucho
modules	Lista de texto completa de los módulos cargados por el núcleo.
Net	Directorio conteniendo un grupo de archivos que posee información acerca de varios aspectos de la configuración en red de la máquina.Normalmente, es más fácil seguir esta información utilizando las herramientas netstat e ifconfig que mirando los archivos en bruto.
Pci	Todos los dispositivos PCI encontrados en el arranque y las configuraciones asignadas a ellos.
SCSI	Directorio que contiene información sobre los dispositivos SCSI de la máquina, así como datos más detallados de los dispositivos duales.
Stat	Estadísticas detalladas de lo que has estado haciendo en el núcleo y el sistema.
sys	Directorio que contiene información relacionada con las variables del núcleo.
uptime	Cuantos segundos ha estado funcionando el sistema sin reiniciarlo y cuanto tiempo ha estado parado.
version	Versión actual del núcleo.

Tabla 2.4 subdirectorios

2.5.7 Actualizar LILO.

Una vez que haya creado un núcleo nuevo con una versión nueva, necesita decirle a Linux Loader (LILO), el gestor de arranque por defecto utilizado por Linux, dónde puede encontrar el archivo de imagen de manera que la maquina pueda arrancar con el nuevo núcleo. Para ello hay que hacer lo siguiente:

1. Abra primero el archivo /etc/lilo.conf, con el editor vi o con su editor de texto preferido.
2. Busque la sección que define los parámetros de arranque del núcleo actual. Debería buscar algo así:

```
image=/boot/vmlinuz-2.2.14-5.0
label=linux
initrd=/boot/initrd-2.2.14-5.0.img
read-only
root=/dev/hdal
```

3. Coloque el cursor en la línea *image.*
4. Copie la sección completa en el buffer del editor. En vi, para una sección de 5 líneas como el ejemplo anterior, escribiría “5yy”
5. Traslade el cursor a una línea en blanco por encima de la línea de imagen, después escriba “p” para pegar el texto. Ahora tiene dos definiciones de imagen idénticas.
6. Cambie ahora los números de versión de la instancia superior, para que coincidan con la nueva versión del núcleo, y después hay que cambiar la etiqueta de modo que pueda decir la diferencia entre los dos. Por poner un ejemplo podría terminar con

```
image=/boot/vmlinuz-2.2.25
label=linux-new
initrd=/boot/initrd-2.2.25.img
read-only
root=/dev/hdal
image=/boot/vmlinuz-2.2.14-5.0
label=linux
initrd=/boot/initrd-2.2.14-5.0.img
read-only
root=/dev/hdal[75]
```

7. Guarde y salga del archivo. Escriba “ZZ” para hacer esto en vi.
8. Escriba “sbin/lilo-v” para ejecutar LILO en modo de descripción detallada. Esta acción asegura que todos los archivos necesarios estén colocados de forma adecuada con objeto del arranque.
9. Cuando reinicie, pulse la tecla Tab para ver las opciones. Arranque con el núcleo nuevo. Si todo esta tranquilo, vuelva atrás y retire de la lista el núcleo antiguo y retírelo también del directorio /boot. Asegúrese de ejecutar LILO otra vez, para llevar a efectos sus cambios.

2.5.8 Shell Scripting.

Este se refiere al entorno en que va a trabajar, no al entorno grafico sino a los comandos y variables a los que tiene acceso en la línea de comandos. El predeterminado en Linux es bash shell, también conocido como bourne again shell, que se basa en el shell sh de unix (el shell Bourne). Otros shell disponibles para los usuarios de Linux son los que se mencionan a continuación.

[75] La razón por la que primero se debe poner el nuevo núcleo, es que la primera imagen arranca por defecto.

Los shells disponibles.

Debido a que hay una gran variedad de shells de Linux para los usuarios del mismo solo me voy a concentrar en los más populares que se enumeran en una tabla a continuación.

N	DESCRIPCIÓN
ash	Basado en el shell predeterminado del sistema V, una ligera variación de sh. Más pequeño que otros shells.
bash	Shell de Linux predeterminado. Una versión actualizada del shell de Unix original, el shell bourne (sh). Nombrado sh bajo Linux. Contiene características de ksh (el shell de korn original) y csh.
bsh	Basado en el shell predeterminado de AIX, otra de las variaciones de sh.
pdksh	La versión del dominio publico del shell de Korn, que se bas en el sh original. Este shell tiene características adicionales de depuración de script que lo hacen atractivo.
csh	El padre del árbol de la familia del segundo shell de Unix. Su lenguaje se basa en el lenguaje de programación C.
tcsh	Una versión actualizada del shell de C original

Tabla 2.5 shells de LINUX

Variables de entorno

Las variables de las que cada shell se ocupa para personalizar una experiencia de usuario son las *variables de entorno.* Estas variables tienden a ser escritas en letras mayúsculas para distinguirlas de las *variables de shell, que* se utilizan para pasar información dentro de un programa. La distinción entre una variable de entorno y una variable de shell es que una variable de entorno mantiene su configuración hasta que se abandona el shell, mientras que una variable de shell solamente dura mientras perdure el programa en el que está configurada. La tabla siguiente muestra una lista de las variables de entorno utilizadas comúnmente.
En cualquier momento, puede ver el contenido de una variable de entorno en el bash shell escribiendo "echo $VARIABLE".

Var	
HOME	Almacena su dirección inicial.
MAIL	Almacena la ruta a su cola de correo, que es el archivo que contiene de su bandeja de entrada.
PATH	Almacena la lista de directorios en la que busca su sistema cuando ejecuta programas.
PS1	Almacena el prompt de registro.
PWD	Almacena el directorio actual en el que está.
SHELL	Almacena el shell que esta utilizando actualmente.
USER	Almacena su nombre de registro.

Tabla 2.6 variable de entorno

Existen dos modos de configurar el valor de una variable de entorno. Una de las formas es "Bash Shell Scripting". La otra manera es configurar el valor de forma permanente editando el archivo ~/bash_profile. Después de hacer esto, esta variable se activa siempre que se registre en la cuenta que utiliza un bash shell.

Bash shell scripting

Puesto que el bash shell es el shell de raíz predeterminado y el más frecuentemente usado entre los usuarios de Linux, nos ocupamos del scripting del bash shell. Casi todos los script del bash shell comienzan con la siguiente línea.

```
#!/bin/bash
```

Esta línea le dice al shell qué intérprete los comandos a utilizar a la hora de ejecutar el script. Permitiéndole ejecutar los shell script escritos para un tipo de shell dentro de otro shell.

Shell scripting básico.

Un shell script consiste en comandos que se pueden utilizar directamente en la línea de comandos. Un comando en un script se ejecuta escribiéndolo igual que lo haría en la línea de comandos, aunque es aconsejable ser más cuidadoso e incluir también la ruta al comando. Por ejemplo, podría ejecutar ls -la simple- mente con la siguiente línea en el script:

```
ls-la
```

No se requieren comandos adicionales o caracteres especiales. También puede que quiera establecer variables de entorno u otras variables. En el bash shell, las variables de shell y de entorno se configuran en este formato:

Variable=valor

No puede haber ningún espacio entre la variable y el signo igual. Esto ayuda al shell a saber que se está refiriendo a una variable y no a un comando. En realidad se utilizan los valores de estas variables colocando el signo del dólar delante de ellas, como *$variable.* Por ejemplo, podríamos tener el siguiente par de líneas en diferentes partes del script:

```
count=10
nextcount=$count + 1
```

Para asignar una cadena de texto, en su lugar, utilizaría lo siguiente:

string="Esta es mi cadena de texto"

También puede configurar una variable para que sea el resultado de un comando colocando el comando en comillas invertidas:

Variable=’*comando’*

Si quisiéramos mostrar el valor de una variable. Igual que hace en la línea de comandos, podemos utilizar el comando **echo** para este fin. Sin embargo, necesita comprender que lo que ocurre depende de la sintaxis que utilice. Podría simplemente tener lo siguiente:

```
echo $Count
```

Este comando mostraría simplemente " 10" si continúa con el ejemplo. También podría tener esta línea:

```
echo “ la cuenta es $cuenta\ .”
```

Utilizar comillas dobles le dice al comando **echo** que calcule el valor de las variables y caracteres que aparecen dentro de las comillas dobles. Si quiere tener esta sentencia impresa literalmente, tiene que utilizar comillas simples corno estas:

```
echo 'La cuenta es $cuenta.'
```

O podría combinar los dos enfoques y utilizar:

```
echo ‘La cuenta es ‘ $cuenta. ’. ‘
```

También puede agregar comentarios con un signo (#). Los comentarios son útiles para asegurar que más tarde cuando reexamine o intente alterar el código, pueda comprender lo que estaba pensando cuando creó el mismo**.** No tiene que tener el signo ($) al principio de la línea. Si desea agregar un comentario en la misma línea como código, puede hacerlo en este formato:

código # todo lo que hay en la línea después de # está comentado

Input y output

Algunos script necesitan interactuar con el usuario o con otros programas. Esto requiere que comprenda las diferentes opciones que tiene a su disposición para aceptar información entrante. Lo primero de lo que debe darse cuenta es que puede incluir entrada directamente en la línea de comandos cuando llame al script. Si simplemente escribe "./script" entonces no hay datos que recordar. Sin embargo, si escribe algo como:

```
.Iscript jadams "John Adams” psmith Paulette Smith
```

Entonces cada uno de los elementos de datos es asignado a variables $. El primer elemento de la línea de comandos, después del nombre del script es $1, es decir jadams. Desde este punto, "John Adams" es $2, psmith es $3, Paulette es $4 y Smith *es $5.* Otras entradas relacionadas con la línea de comandos son las listadas a continuación.

Variable	Contiene
$@	Una lista completa de todos los argumentos de la línea de comandos, entre comillas.
$#	El número de argumentos de la línea de comandos.
$*	Una lista completa de todos los argumentos de la línea de comandos.

Tabla 2.7 variables de input y output

También puede utilizar el comando **read** para aceptar datos desde el usuario. Esta herramienta proporciona un prompt al usuario que le permite entrar en una línea de material. Para poner toda la entrada en una variable, utilice para ello este formato:

read variable

También es posible tener en la línea el elemento de cada individuo, con espacios separando los valores, colocados en variables diferentes, utilizando este formato:

read *variable variable1 variable2 ... variableN*

Otra herramienta valiosa es **getopts**[76]**.** Esta herramienta se utiliza dentro de los bucles con el siguiente formato:

getopts *letras variable*

La porción ***letras*** de la sentencia contiene letras individuales que corresponden a la entrada, con cada letra representando un elemento concreto. Si la letra es simplemente un comando sin argumentos, entonces escriba la letra, por ejemplo "a". Sin embargo, si se trata de un comando con su correspondiente argumento, coloque dos puntos (:) detrás de él, por ejemplo, "a:". El elemento *variable* representa el nombre de la variable que quiere utilizar para caminar por las letras en un bucle.
La siguiente sentencia espera tres comandos diferentes:
getops a variables

Sin embargo, en realidad esta sentencia lo que espera son tres comandos y un argumento:

getops **a:** variable

La entrada para esta sentencia podría ser:

```
mc>re
```

mientras que la entrada para la segunda sería:

```
more/etc/profile
```

Si quiere acceder al argumento, entonces puede utilizar para ello la variable **OPTARG.**
No toda la entrada viene del usuario. Parte de la entrada viene de otros programas. Hay fundamentalmente dos métodos para compartir datos entre programas. El primero se denomina redireccionar.
Antes de aprender a utilizar la redirección, necesitamos entender algunos otros conceptos básicos del shell:

[76] **getopts** en realidad no se usa para permitir a un usuario entrar a la información. Se utiliza como método para analizar la información proporcionada en la línea de comandos.

- *STDOUT (standard output):* Cuando escribe un comando, quizá ls, ese comando a menudo envía su salida a STDOUT. Normalmente, su STDOUT es su pantalla.
- *STDIN (standard input):* Siempre que escriba información en la línea de comandos, está utilizando STDIN. Los programas también pueden utilizar esta interfaz.
- STDERR *(standard error):* Muchos programas envían sus mensajes de error a STDERR.

Cuando trata con la redirección, en realidad está tomando datos que iban dirigidos a STDOUT, STDIN o STDERR y alterando su curso con el fin de que vayan donde los quiera llevar. Los operadores utilizados para ello se encuentran en la tabla siguiente.

Símbolo	Resultado	Ejemplo
>	Intercepta datos que van a STDOUT y los envía al archivo Especificado.	ls-la > lista
2>	Intercepta datos que van a STDERR y los envía al archivo especificado.	ls estás ahí 2> errores
<	Abre el archivo especificado y envía el contenido a STDIN.	gato>lista
>>	Intercepta datos que van a STDOUT y los envía al archivo especificado, adjuntando la información actual ya existente si el archivo ya contiene texto.	ps aux>> listadeprocesos

Tabla 2.8 operadores de STDOUT, STDIN Y STDERR

El segundo método para controlar lo que le ocurre a la salida de un programa es el tubo. Esta característica está representada por una línea vertical (|). Más que tratar con archivos, utilizar un tubo le permite colocar la salida del comando a su izquierda y la entrada a su derecha, como representamos aquí:

Comando1 | comando2

Puede tener más de dos comandos cuando trata con tubos; podría incluso tener esta estructura:

Comando1 | comando2 | comando3 | comando4 | comando5

Condicionales y bucles.

Condicionales (sentencias que funcionan solamente si un elemento es verdadero), y bucles (sentencias que se repiten de acuerdo con un criterio especificado) que están disponibles en los script de bash shell. Los tipos de condicional y bucles, así como sus usos, son los que aparecen abajo.

Inicio	Fin	Intermedio	Propósito
for	done	do	Traspasa el bucle el número de veces prescrito.
If	fi	then, else,elif	Comprueba si las condiciones dadas son verdaderas o falsas; continúa si son verdaderas.
until	done	do	Comprueba si las condiciones dadas son verdaderas o falsas continúa si son verdaderas.
while	done	none	Traspasa el bucle si la condición dada es verdadera.

Tabla 2.9 condicionales y bucles en el bash shell

La sentencia if condicional es, la más compleja. Su formato más básico, es :

```
if condición
    then resultado
fi
```

Este, código tiene como resultado de que si se cumple la condición, todas las líneas del código, que puede ser una o muchas, de then a fi se ejecutan. Puede tener este otro formato:

```
if condición
    then resultado
    else resultado
fi
```

Con esta estructura, su condicional ahora tiene opciones. Comprueba si la condición es verdadera, y si lo es, continua con la línea de código asociada con la oración then. Pero si la condición es falsa, en vez de salirse de la condicional completamente, el código asociado con la oración else se ejecuta.
Finalmente, ésta es la condicional más compleja:

```
if condición
then resultado
elif condición
      resultado
fi
   else resultado
fi
```

La oración elif significa "else if". La condición if inicial comprueba los valores que se proporcionan en el método que se especifica. Un resultado verdadero envía el script al código asociado con la oración then y, después, continúa hasta pasar la sentencia condicional completa. Si se comprueba que la condición es falsa, entonces se comprueba una segunda condición. Un resultado verdadero para la segunda condición ejecuta el código asociado con la oración elif. Si las dos condiciones son falsas, entonces el script salta al código asociado con la oración else.

Los tres tipos de bucle (for, until y while) utilizan el mismo formato, como el siguiente:

Tipo de bucle condición do

resultados

done

Lo importante es determinar qué tipo de bucle necesita utilizar.

Probar condiciones.

Tanto las condiciones como los bucles requieren que compruebe los datos para ciertos valores o con otros tipos de datos, para que sepan cómo continuar. Estas comparaciones se realizan con el comando test. Existen varias maneras de utilizar test, dependiendo de si está comparando elementos o bien buscando características.
Un tipo de archivo es una característica que puede comprobar. Esta información puede ser importante cuando intenta asegurarse de que su programa tiene alguna discriminación sutil a la hora de manejar cualquier problema en el que se pueda meter, como, por ejemplo, necesitar escribir en archivos que no existen. La tabla siguiente contiene una lista de los operadores que se encuentran disponibles para utilizar el comando test de esta forma:
Test operador /path/File

Operador	Verdadero cuando
-b	El dispositivo especial de bloques, lo que significa que es medio de almacenamiento de algún tipo, existe.
-c	El dispositivo especial de carácter, lo que significa que es un dispositivo de salida, como un MODEM o un monitor.
-d	El directorio existe.
-f	El archivo existe.
-L	El enlace simbólico existe.
-p	El FIFO, cauce designado, existe.
S	El socket existe.

Tabla 2.10 operadores del comando test

Otro tipo de comprobación disponible es la de examinar los permisos de un archivo. Esta comprobación es útil para asegurarse de que los archivos creados por su script tienen finalmente los permisos configurados de la manera adecuada, incluso si ha cambiado los permisos predeterminados en su sistema desde que escribió el script. La tabla siguiente contiene una lista de los operadores disponibles para llevar a cabo la comprobación de los permisos. Estos operadores se utilizan con el siguiente formato:

test operador / path/ file

Operador	Verdadero cuando
-g	El archivo es SGID[77].
-G	El archivo es propiedad del GID[78] que ejecuta el script.
-k	El archivo tiene su conjunto de sticky bit[79].
-O	El archivo es propiedad del GID.
-r	El script tiene permisos de lectura para este archivo.
-u	El archivo es SUID[80].
-w	El script tiene permisos de escritura para este archivo.
-x	El script tiene permisos de ejecución para este archivo.

Tabla 2.11 operadores para comprobar permisos

También puede comprobar información adicional sobre los archivos, principalmente a través de comparaciones. Esto es útil si quiere asegurarse de que está utilizando la última versión de un archivo o el original en lugar de un enlace. La tabla siguiente proporciona la lista completa. En este caso, el formato en el que utiliza los operadores varía.

Operador	Verdadero cuando	Formato
-e	El archivo existe.	-e archivo
-ef	Los dos archivos son enlaces fijos el uno para el otro.	Archivo –ef archivo 2
-nt	El primer archivo fue modificado más recientemente.	Archivo –nt archivo2
-ot	El segundo archivo fue modificado más recientemente que el primero.	Archivo –ot archivo2
-s	El archivo existe y es mayor de 0 bytes.	-s archivo

Tabla 2.12 Comparaciones para operadores

No todas las operaciones de comprobaciones se llevan a cabo en archivos. También puede utilizar **test** para examinar cadenas de texto. Estos operadores son útiles para comprobar la entrada de usuario, contenidos de archivo y mucho **más.** La tabla siguiente proporciona una lista de los operadores **test** de cadena de texto.

[77] Significa 'Set Group ID', es decir, 'Fijar la Identidad del grupo'. Es análogo
[78] Group IDentification
[79] (bit de permanencia): sólo lo puede activar root. Provoca que el programa al que se le aplica quede residente en memoria de forma que la próxima vez que sea llamado su carga sea más rápida.
[80] significa 'Set Users ID', o sea, fija la identificación del dueño y el programa se ejecuta con la identificación y los permisos del dueño. Un caso típico es el del programa *passwd*, que cambia la contraseña de entrada de los usuarios modificando el archivo */etc/passwd*.

Operador	Verdadero cuando	Formato
=	Las cadenas son las mismas.	cadena1 = cadena2
!=	Las cadenas no son las mismas.	cadena1!= cadena2
-n	La cadena no está vacía.	–n cadena
-z	La cadena esta vacía.	–z cadena

Tabla 2.13 Operadores de test en cadenas de texto

Más familiarizado. La tabla siguiente contiene una lista de los operadores con los que puede comparar valores numéricos. Todos estos elementos se utilizan en el siguiente formato:

```
test valor1 operador valor2
```

Operador	Verdadero cuando
-eq	Los dos valores son iguales.
-ge	El primer valor es mayor o igual que el segundo.
-gt	El primer valor es mayor que el segundo.
-le	El primer valor es menor o igual que el segundo.
-lt	El primer valor es menor que el segundo.
-nt	Los dos valores no son iguales.

Tabla 2.14 Operadores ara comparar valores numéricos

Finalmente, algunos operadores funcionan para las condiciones generales de test. Si coloca un signo de admiración delante de la condición le dice a test que invierta el resultado verdadero o falso. Por ejemplo, esto resulta en falso:

```
! 1 -eq 1
```

Si quiere solicitar que dos condiciones diferentes de test han de ser verdaderas, entonces utilice este formato:

```
Condición1 -a condición2
```

Sustituya la -a (y) por -o (o) en caso de que solamente una de ellas tenga que ser verdadera.

Cambiar el shell activo.

Si quiere cambiar el shell activo en curso para su cuenta, debe hacer lo siguiente:

1. Entre en la cuenta en la que quiere configurar el shell.

2. Escriba a continuación "chsh -1 " con objeto de ver los shell que tiene disponibles, incluidos también sus nombres de ruta completos. Con la instalación por defecto de Red Hat, debería tener a su disposición todos los siguientes:

```
/bin/bash
/bin/sh
/bin/ash
```

```
/bin/bsh
/bin/tcsh
/bin/csh
```

3. Escriba "chsh –s/ruta/shell" para cambiar el shell predeterminado para su cuenta.

Instalar el shell korn de dominio público.

Para instalar el shell korn de dominio público en Red Hat de Linux, haga lo siguiente:

1. Entre en el sistema como root.
2. Si no tiene ya instalado el CD-ROM de Red Hat en la unidad de CD-ROM, insértelo.
3. Escriba después `"mount/mnt/cdrom"` para montar el CD-ROM en el sistema de archivos.
4. Escriba `"cd/mnt/cdrom/RedHat/RPMS"` para cambiar al directorio de paquetes en el CD-ROM
5. Escriba `"rpm -ivh pdksh"` y pulse Tab para completar el nombre del archivo e Intro para instalar el shell. Se mostrará al usar el comando chsh.

Estructura de archivos y directorios.

Linux proporciona estructuras y soporte para poder gestionar sus propios archivos. Esta estructura consiste en una seria jerárquica de niveles de directorios, llamado raíz, el cual se divide en varios subdirectorios, que a su vez, se puede ramificar en más subdirectorios.
A cada usuario se le asigna un subdirectorio propio, de tal manera que cuando usted accede a ese directorio, se convierte en su directorio de trabajo actual.

Directorio (/)	Contenido.
/bin	Ficheros binarios o ejecutables.
/dev	Dispositivos especiales.
/etc	Información y programas.
/sbin	Ordenes ejecutables solo por el administrador.
/home	Directorio de usuario.
/lib	Librerías.
/proc	Estructura virtual de archivos.
/tmp	Archivos temporales.
/usr	Archivos de configuración y programas usados por el sistema.
/var	Históricos del sistema.
/boot	Información necesaria para el sistema de arranque
/dev/console	Sistema de consola.

/dev/ttyS	Acceso a puertos.
/dev/cua	Acceso a puertos.
/dev/hda	Primer disco duro.
/dev/sda	Primer disco duro SCSI.
/dev/lpo	Primer puerto paralelo.
/dev/tty	Consolas virtuales.
/dev/pty	Seudoterminales.
/usr/x386	Sistema x window.
/usr/bin	Archivos binarios o ejecutables.
/usr/etc	Información y programas.
/usr/include	Archivos para compilar C.
/usr/man	Página man.
/usr/src	Código fuente.
/u sr/scr/linux	Código fuente del núcleo.
/var/adm	Archivos de administración.
/var/spool	Archivos spool.
/usr/x11/bin	Ejecutables X window.

Tabla 2.15 Estructura de archivos y directorios

En el presente capitulo estudiamos temas basico en el tema de Linux como comenzando por su historia que es Linux y tambien nos centramos en el núcleo del sistema como sus partes del mismo aprendimos a actualizar lilo como el shell del mismo.

En este capitulo ya vimos lo mas basico de Linux pero es importante hablar de la segurida que nos proporciona a nivel sistema, a nivel usuario y a nivel red, tocaremos este tema en el siguiente capitulo la segurida y las herramientas del sistema que nos apoyan en este y analizaremos varias opciones de las mismas.

III. SEGURIDAD EN LINUX

Cuentas de usuarios

En Linux, toda la potencia administrativa se confiere a una sola cuenta llamada root, que es el equivalente al Administrador de Windows NT. Con esta cuenta se controla:

Cuentas de usuario, archivos, directorios y Recursos de red.

La cuenta root permite realizar cambios masivos en todos los recursos o cambios específicos solamente en unos pocos. Por ejemplo, cada cuenta es una entidad independiente con un nombre de usuario, una contraseña y unos derechos de acceso independientes, lo que otorga o deniega accesos a cualquier usuario, combinación de usuarios o a todos los usuarios.

Se puede apreciar que el Usuario A tiene una autorización superior a la que le corresponde debido a que ha roto (hecho crack) las contraseñas del sistema, lo que no se debe hacer. Mientras se investiga el problema, es posible congelar la cuenta del usuario sin que ello afecte a los Usuarios B y C. Linux mantiene aislados a los usuarios de esta forma, en parte por motivos de seguridad y en parte para imponer orden en su entorno, la siguiente grafico nos muestra el potencial que tiene la cuenta root.

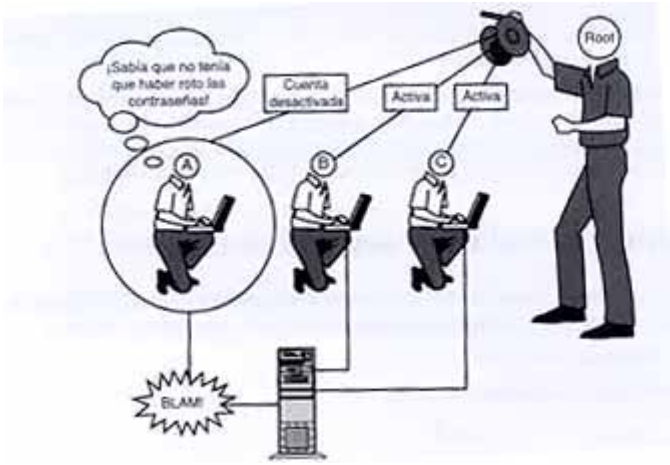


Fig. 3.1 la cuenta root administra todas las cuentas de usuarios

Los usuarios generan sus propios archivos, instalan sus propios programas, etc. Para mantener el orden, Linux mantiene aislados los directorios de los usuarios. Cada usuario recibe un directorio principal y un espacio en el disco duro. Esta ubicación es independiente de las áreas del sistema y de las que ocupan los restantes usuarios.

Con ello, se evita que la actividad normal de los usuarios afecte al sistema de archivos. Además, proporciona a los usuarios una cierta privacidad, cada uno de los usuarios posee sus propios archivos y, a menos que especifique lo contrario, los restantes usuarios no pueden acceder a ellos

Como root, se controla a los usuarios que tienen acceso y el lugar en que almacenan sus archivos. Esto es sólo el principio. También puede controlar a qué recursos pueden acceder los usuarios y cómo se manifiesta dicho acceso.

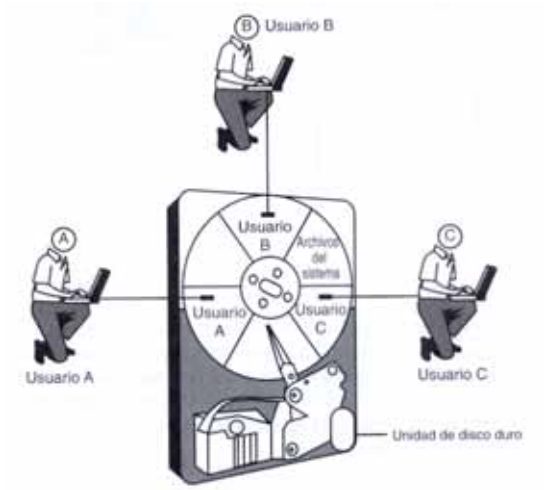


Fig. 3.2 los directorios de los usuarios están aislados de los del sistema

3.1 Control de acceso discrecional (DAC)

Permite controlar el grado hasta el que pueden acceder a los archivos y directorios los distintos usuarios. Es posible especificar con total exactitud la forma en que los Usuarios A, B y C acceden a los mismos archivos. El Usuario A puede leer, escribir y ejecutar los tres archivos. Por contra, el Usuario B sólo puede leerlos y escribir en ellos, y, finalmente, el Usuario C no puede siquiera acceder a ellos. Dichas limitaciones se implementan a través de los grupos, como se muestra en el siguiente grafico.

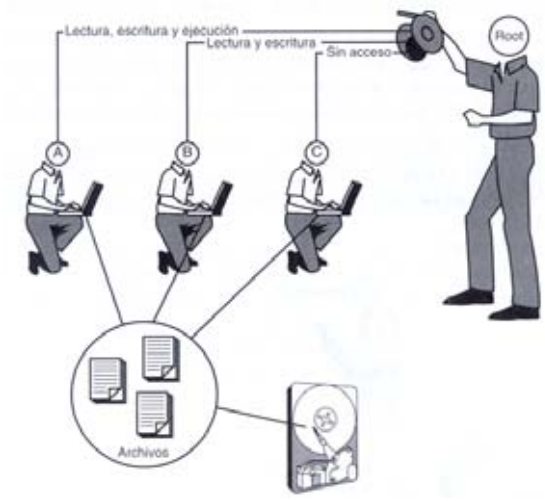


Fig. 3.3 el root controla el acceso de los usuarios a los archivos

Dado que, a menudo, las organizaciones se dividen en departamentos y que es posible que varios usuarios de dichos departamentos tengan que acceder a los mismos archivos, Linux permite agrupar a los usuarios. De esta forma,

cuando se definen permisos para determinados archivos y directorios, no es necesario hacerlo para todos y cada uno de los usuarios.

Como se muestra en la siguiente figura, el Grupo A tiene acceso de sólo escritura, mientras que el Grupo B tiene acceso de lectura y escritura. Dicha gestión a nivel de grupo resulta muy útil cuando hay muchos usuarios y varios subconjuntos de usuarios necesitan privilegios idénticos o muy parecidos.

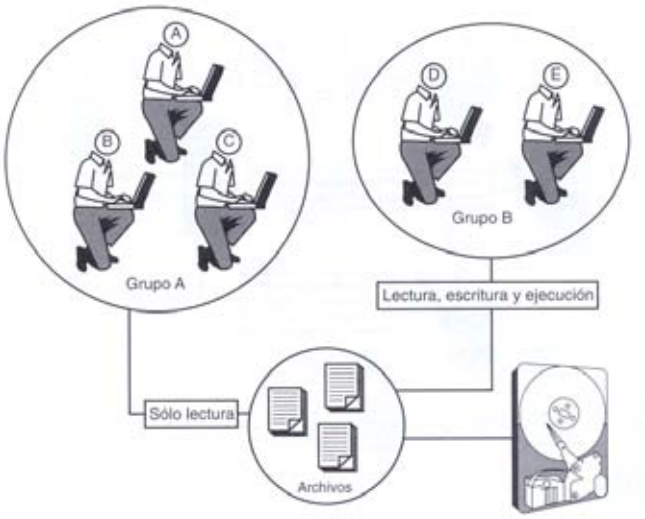


Fig. 3.4 los grupos son conjuntos de usuarios con accesos similares

3.2 Control de acceso a la red

Es posible implantar reglas de acceso a la red extremadamente refinadas. El Usuario A no se puede conectar, el Usuario B debe utilizar una máquina determinada para poder conectarse y el Usuario C puede conectarse libremente desde el lugar que desee.

Esta funcionalidad viene muy bien en los entornos de red o cuando el sistema Linux es un servidor de Internet. Por ejemplo, permite mantener un servidor Web solamente para los clientes de pago. Si se quiere dar un paso más, quizás se desee no permitir que hosts no autorizados intenten conectarse. En Linux, muchos servicios de red ofrecen esta función.

Fig. 3.5 El root controla quien puede acceder al servidor.

3.3 Cifrado

Es el proceso de mezclar los datos para que no puedan leerlos los que no tengan autorización para ello, en la mayoría de los esquemas de cifrado, es necesario tener una contraseña para reorganizar los datos de forma que puedan leerse. El cifrado se utiliza principalmente para mejorar la privacidad o para proteger información importante.

Por ejemplo, Linux ofrece varias opciones de cifrado punto a punto para proteger los datos que circulan.

Habitualmente, cuando se transmiten datos a través de Internet atraviesan muchas *gateways* (pasarelas). En su camino, dichos datos son vulnerables a escuchas electrónicas. Linux cuenta con varias utilidades complementarias que permiten cifrar o codificar los datos para que si alguien los captura, sólo vea un tremendo desorden.

Por ejemplo, los datos de la tarjeta de crédito del Usuario A se cifran antes de salir de su red interna y permanecen así hasta que el servidor de comercio los descifra. Este proceso protege a los datos de ataques y posibilita un comercio electrónico seguro.

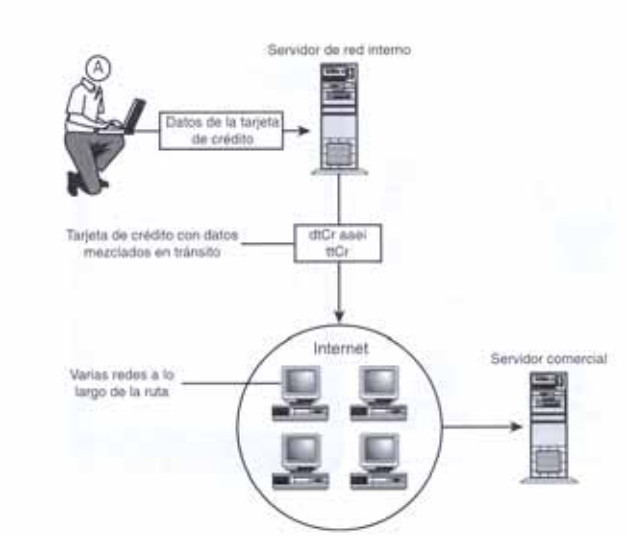


Fig. 3.6 Linux cifra datos durante su circulación y así los protege de personas de fuera

3.4 Registro, auditoria y control de red integrados

Los intrusos rápidamente sacan partido de estas oportunidades mediante el ataque al mayor número de máquinas posible antes de que se arregle el agujero. Linux no puede predecir cuándo va a sufrir algún ataque un *host,* pero puede registrar los movimientos de la persona que realiza dicho ataque

Linux detectara, marcara la hora y grabara las conexiones de red. Esta información se redirige a los registros para su posterior examen.

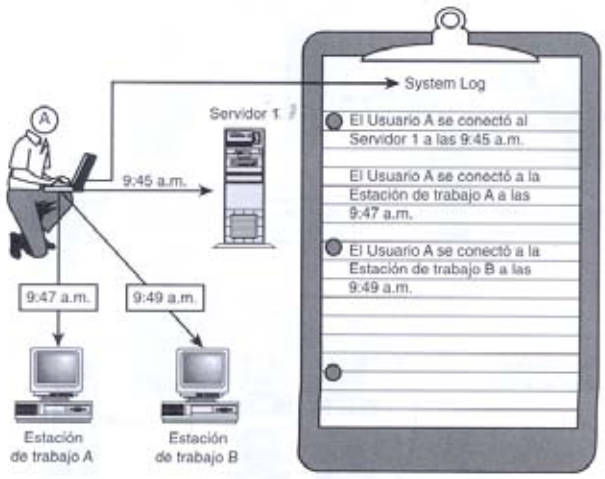


Fig. 3.7 Linux registra todas las conexiones entrantes.

La capacidad de registro es un componente vital de la arquitectura de seguridad de Linux y proporciona la única evidencia real de que se ha producido un ataque. Teniendo en cuenta que hay un gran número de metodologías de ataque distintas, Linux graba registros a nivel de red, de *host* y de usuario. Por ejemplo, Linux realiza las siguientes funciones:

- Registra todos los mensajes del sistema y del núcleo.
- Registra todas las conexiones de la red, la dirección IP de la que parte cada una de ellas, su longitud y, en algunos casos, el nombre de usuario y sistema operativo de la persona que realiza el ataque.
- Registra los archivos que solicitan los usuarios remotos.
- Puede registrar qué procesos se encuentran bajo el control de cualquier usuario.
- Puede registrar todos y cada uno de los comandos que ha emitido un usuario determinado.

Los registros son indispensables cuando se investigan las intrusiones en la red, aún cuando dichas investigaciones se realicen a posteriori. Sin embargo, dado que Linux graba los registros en tiempo real, se podría pensar que debe de haber alguna forma en la que Linux responda a los ataques.

3.5 Detección de intrusiones.

Hay muy pocos sistemas operativos que incluyan herramientas de detección de intrusiones. De hecho, hace muy poco tiempo que dichas herramientas se han introducido en las distribuciones estándar de Linux.

Entre las herramientas con que cuenta Linux y los complementos que pueden descargarse de Internet, es posible establecer una avanzada capacidad de detección de intrusiones

- Es posible hacer que Linux registre los intentos de intrusión y que avise cuando se produzcan dichos ataques.

- Es posible hacer que Linux acometa acciones predefinidas cuando los ataques cumplan unos criterios específicos.
- Es posible hacer que Linux distribuya desinformación, como por ejemplo, que imite a un sistema operativo que no sea Linux. La persona que lleva a cabo el ataque pensará que está desprotegiendo un sistema Windows NT o Solaris.

De hecho, la mayoría de las distribuciones de detección de intrusos y de engaños son conjuntos de herramientas. Todos estos mecanismos forman los componentes individuales de la compleja arquitectura de seguridad de Linux. Uno a uno, es posible que no parezcan tan extraordinarios, pero cuando se utilizan de forma conjunta, constituyen un exhaustivo método global en lo relativo a la seguridad de redes.

3.6 Administración básica del sistema Linux.

El root controla a los usuarios individuales, a los grupos y los archivos, y dicho control se ejerce, habitualmente, en una secuencia lógica. Como lo demuestra la siguiente figura.

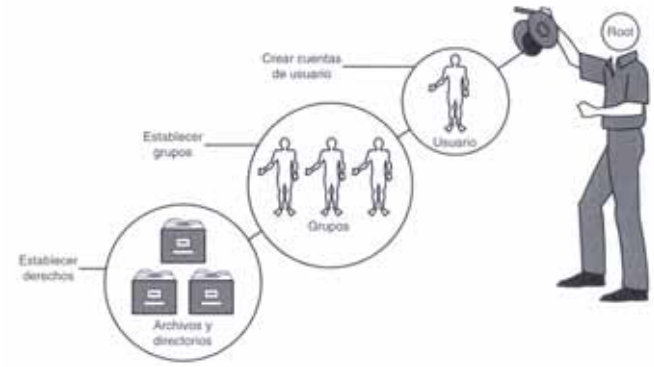


Fig. 3.8 El root crea las cuentas y define los grupos y los derechos de acceso.

Se empieza con usuarios individuales o un conjunto de ellos. Al crear sus cuentas, dichos usuarios se organizan en grupos dependiendo de sus tareas respectivas y de sus necesidades de acceso. Finalmente, se definen con mayor precisión los derechos de acceso individuales de cada usuario en aquellos lugares en los que difieran de los de su grupo.

El *root* es una cuenta. Es una cuenta con autoridad suficiente. La cual no se debe utilizar nunca para fines personales, salvo que sea absolutamente necesario. Esta restricción se debe a varios motivos. En primer lugar, como raíz tiene poder absoluto. Tiene todos los permisos posibles sobre los archivos y ninguna restricción de acceso puede cambiar cualquier cosa en cualquier momento. Este poder es muy útil, pero si se utiliza indiscriminadamente, se pueden provocar de forma involuntaria daños irreparables. Se puede abrir el sistema a incalculables amenazas a la seguridad.

3.6.1 Crear y administrar cuentas.

La autorización para iniciar la sesión es un privilegio que no hay que conceder nunca a la ligera. Si es posible proporcionar a los usuarios servicios críticos sin concederles acceso al *shell*. El acceso al *shell* se produce cuando los usuarios tienen acceso remoto por telnet a una *shell* local del servidor. Ya que cuantos más usuarios tengan acceso a la *shell,* más probable es que aparezca una brecha en la seguridad.
Los usuarios de la shell pueden aprovecharse de archivos y servicios a los que no pueden acceder los atacantes remotos. Éstos deben obtener acceso a la *shelí* antes de aprovechar los agujeros internos; un usuario válido de la *shell* ya ha recorrido la mitad del camino. Pero, aunque no tengan intenciones malvadas, los usuarios de la shell también pueden causar problemas, por ejemplo si los usuarios crean archivos *rhosts.* Si es imprescindible otorgar a los usuarios acceso a la *shell* durante la creación de una red Linux, con estas medidas al menos se reducen los riesgos:

- Dedique una máquina exclusivamente para el acceso a la *shell.*
- Restrinja dicha máquina solamente para el uso de la shell.
- Elimine de ella todos los servicios de red que no sean esenciales.
- Instale un conjunto genérico de aplicaciones y al crear las particiones tenga en cuenta el restablecimiento tras algún desastre. En otras palabras, es de esperar que haya que reinstalar Linux con frecuencia.
- Prohíba las relaciones de confianza entre la *shell* y otras máquinas. Considere la posibilidad de separar los sistemas de archivos importantes (/tmp, /home, /var) en otras particiones y mueva los binarios *suid* a una partición que linux monte no setsuid.
- Redirija los registros a un servidor de registros o, si el presupuesto lo permite, a algún medio en el que sólo se pueda escribir una vez y registre todo.

Si va a configurar una sola máquina con Linux, aplique las mismas reglas básicas: conceda acceso a la *shell* exclusivamente a aquellos que realmente lo necesiten.

3.6.2 Estructura de las cuentas

passwd

El archivo passwd se encuentra en el directorio /etc. Si ha estado utilizando Linux en un entorno puramente gráfico y aún no domina la línea de comandos.

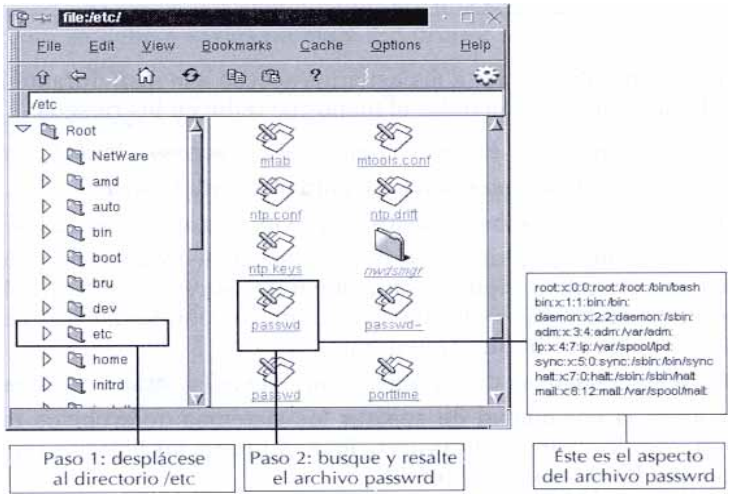


Fig. 3.9 etc/passwd con el administrador grafico de archivos

Cada línea almacena el registro de una cuenta y cada registro consta de siete campos (los campos de las cuentas están delimitados por columnas). Vamos a examinar cada uno de los campos utilizando la cuenta asignada al usuario matt.

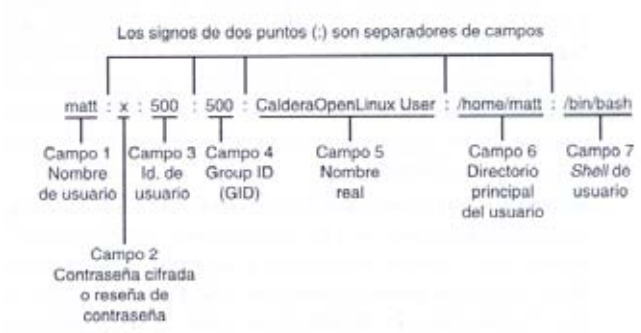


Fig. 3.10 /etc/passwd se divide en siete campos delimitados por (:), username, password, userid, realname, user home y user shell

Username: Almacena el nombre de usuario del interesado. Es aconsejable crear nombres de usuario que se parezcan a los nombres reales de los usuarios. Por ejemplo, si el nombre real de un usuario es María del Val Hernández, su nombre de usuario podría ser mvhdez. Esta práctica no es imprescindible, pero facilita el reconocimiento de los usuarios mediante sus nombres de usuario, lo que resulta de especial importancia en los entornos empresariales. La longitud de los nombres de usuario no debe superar los ocho caracteres y debe escribirse en minúsculas.

Password: Almacena la contraseña de acceso del usuario. Cada una de las versiones de Linux almacena la contraseña de los usuarios de forma distinta. Las anteriores distribuciones de Linux almacenaban la contraseña del usuario en forma cifrada (como, por ejemplo, x1mmmFtgA8), mientras que las nuevas que emplean *shadowing* sólo almacenan una reseña de la contraseña (x) y ocultan la contraseña cifrada en otra parte.

Userid: Almacena el número de identificación de usuario (UID) del usuario. Este número se adjunta a los procesos del usuario. Cuando se elige el UID de

un usuario nuevo se puede asignar cualquier número único y arbitrario entre el 0 y el 65534 (no utilice el 0, ya que es *root).* Sin embargo, no es aconsejable que los UID sean demasiado arbitrarios. En su lugar, reserve un bloque de números específicamente para los usuarios y asígnelos secuencialmente. Por ejemplo, puede restringir los UID a números entre 500 y 700. El primer usuario es el 501, el segundo el 502 y así sucesivamente. De esta forma, con un solo vistazo a la lista de procesos puede saber quién está realizando cada tarea. Si la lista de procesos informa de que hay varios UID en el rango 500-700, sabrá qué usuarios poseen cada uno de los procesos (en la mayoría de los casos no será necesario que se moleste en elegir el UID, ya que muchas de las últimas herramientas de administración de Linux lo hacen automáticamente).

groupID: Almacena el número de identificación de grupo del usuario, que refleja el grupo nativo del usuario. Los usuarios pueden pertenecer o no a otros grupos, pero siempre pertenecen a su grupo nativo. Cada versión de Linux asigna este campo de forma distinta. La mayoría de las distribuciones colocan a todos los usuarios en el mismo grupo predeterminado (por ejemplo, users). Caldera y Red Hat asignan a cada usuario su propio grupo, llamado grupo privado. En este mismo capítulo se explican los grupos con mayor detalle. De nuevo, no utilice el 0, ya que es r*oot.*

Real name: Suele recibir el nombre de campo General Electric Comprehensive operating System (GECOS) y almacena el nombre real del usuario, entre otras cosas. Si no se define, Linux lo ajustará automáticamente (como hacia OpenLinux en el caso de matt). Este campo se utiliza principalmente para temas relacionados con los informes, como por ejemplo en respuesta a las consultas *finger*. Tenga en cuenta que en este campo se puede definir otra información, entre la que se incluye el número de teléfono de casa o del trabajo del usuario.

user home: Almacena la ubicación del directorio de inicio del usuario (en este caso, /home/matt). Si durante la instalación se ha creado una partición y un directorio especiales para los usuarios (que no sea /home), ése es el que hay que seleccionar. Sin embargo, hay que asegurarse de que todos los directorios de los usuarios se mantienen en la misma partición y bajo la misma jerarquía de directorios. A menos que exista una buena razón para no hacerlo, es muy aconsejable almacenar los directorios de los usuarios en /home.

user shell: Almacena la shell predeterminada del usuario. Ésta es la *shell* en la que entra el usuario la primera vez que se conecta. Si se ha cargado toda la distribución de Linux, se puede elegir entre varias opciones: ash, csh, bash, ksh, tcsh, zsh, etc. Sin embargo, es recomendable restringir a todos los usuarios a una *shell* común. Cuantas más debilidades tengan las *shells* que se proporcionen, más oportunidades tendrán los crackers de encontrar un agujero en una de ellas.
Pero hay algo más además de las entradas de /etc/ passwd. Durante el proceso de creación de cuentas también hay que crear directorios, entre los que se incluye el directorio de inicio del nuevo usuario, habitualmente /home/user.

Además, si las cuentas se añaden manualmente, será necesario copiar los archivos de inicio predeterminados (que se encuentran en /etc/ skel) en el directorio de inicio del nuevo usuario (y definir los permisos apropiados).

Es muy probable que /etc/ skel contenga estos archivos:

```
-rw-r--r--   1 root     root          49 Nov 25  1997 .bash_logout
-rw-r--r--   1 root     root         913 Nov 24  1997 .bashrc
-rw-r--r--   1 root     root         650 Nov 24  1997 .cshrc
-rw-r--r--   1 root     root         111 Nov  3  1997 .inputrc
-rwxr-xr-x   1 root     root         186 Sep  1  1998 .kshrc
-rw-r--r--   1 root     root         392 Jan  7  1998 .login
-rw-r--r--   1 root     root          51 Nov 25  1997 .logout
-rw-r--r--   1 root     root         341 Oct 13  1997 .profile
-rwxr-xr-x   1 root     root         182 Sep  1  1998 .profile.ksh
drwxr-xr-x   2 root     root        1024 Jun  4 21:37 .seyon
```

En algunos sistemas, profile recibe el nombre de local.profile.
En su estado original, el propietario de estos archivos es *root*. Para prepararlos para que los utilice otro usuario, se puede hacer lo siguiente:

```
mkdir /home/newuser
cp /etc/skel/.* /home/newuser/
chown newuser /home/newuser
chown newuser /home/newuser/.*
chgrp newuser-userid /home/newuser
chgrp/home/newuser/.*
chmod 755 /home/newuser
chmod 644 /home/newuser/.*
```

3.6.3 Añadir usuarios

Añadir usuarios con herramientas gráficas

Dependen de su distribución de Linux. Una de ellas es usercfg, disponible en Red Hat y Caldera OpenLinux.

usercfg

Necesita: usercfg + python

Archivos de configuración: ***/usr/lib/rhs/control-panel/usercfg.init, /usr/lib/rhs/usercfg, /usr/lib/rhs/usercfg/usercfg.py, usr/lib/rhs/usercfg/usercfg.pyc***

usercfg es una herramienta independiente para la administración de cuentas, pero para utilizarla hay que tener el lenguaje Python y sus bibliotecas. Si se realiza una instalación completa, no debe de haber ningún problema. Sin embargo, si se seleccionan de forma selectiva las herramientas de desarrollo y se han excluido *usercfg y* Python, hay que instalarlas ahora. *usercfg* se encuentra en /usr/bin. Tenga en cuenta que la interfaz gráfica de *usercfg* puede variar. En algunos casos, se basa en X, mientras que en otros, se ejecuta a través de LISA con cuadros de diálogo a través de una *shell* o desde un indicativo de comandos.

Para iniciar *usercfg* desde X, haga clic en su icono en la ventana Admin Tools en Caldera o en el panel de control en Red Hat Si no puede encontrar *usercfg* ahí, abra un Xterm y escriba la siguiente línea de comandos:

```
$usercfg
```

LISA (la herramienta de instalación y administración de sistemas de Linux) cargará usercfg.

El botón Call ya estará resaltado. Desplácese a Add New User y pulse Intro. usercfg le llevará a los seis pasos necesarios para crear una cuenta:

1. Añadir el nombre de conexión del usuario.
2. Añadir el UID del usuario.
3. Añadir el grupo del usuario.
4. Añadir el directorio principal del usuario.
5. Añadir la *shell* predeterminada del usuario.
6. Añadir el nombre completo del usuario.

Para finalizar, *usercfg* abrirá una interfaz de texto y solicitará una contraseña:

```
Enter new Unix password:
```

Escriba la contraseña del nuevo usuario y pulse Intro. *usercfg* solicita confirmación de la contraseña:

```
Enter new UNIX password:
```

Tras verificar la contraseña, *usercfg* almacenará la información en /etc/passwd.

En las últimas versiones de linux, las herramientas gráficas de administración varían. Openlinux 2.2 incluye COAS *(Caldera Open Adrninistration System),* mientras que Red Hat incluye linuxconf.

Fig. 3.11 pantalla inicial de usercfg.

Añadir usuarios con adduser.

adduser

Aplicación: adduser

Necesita: adduser + /bin/sh

Archivos de configuración: ninguno

Utilidades parecidas: useradd

Para utilizar adduser, escriba el comando adduser añadiéndole un nombre de Usuario:

```
$ adduser Nicole
```

adduser hace todo menos definir la contraseña:

```
Looking for first available UID    ... 508
Looking for first available GID    ... 509
Adding login: Nicole ... done.
Creating home directory: /home/Nicole ... done.
Creating mailbox: /var/spool/mail/Nicole ... done.
Don't forget to set the password.
```

Para definir la contraseña, escriba el comando *passwd con* el nombre de usuario. Por ejemplo, en este caso, el comando sería:

```
$ passwd nicole
```

Linux pide la contraseña y la confirmación de la misma:

```
Enter new UNIX password:
Enter new UNIX password:
```

adduser asigna automáticamente los valores, incluyendo el UID y el GID. Si desea un mayor control de la línea de comandos y ha instalado Shadow Suite pruebe con useradd.

Shadow Suite es un conjunto de herramientas para ocultar la información de la contraseña. Habitualmente, Linux almacena toda la información del usuario, incluyendo las contraseñas cifradas, en /etc/passwd. Esta práctica es poco segura, ya que expone las contraseñas cifradas a todos los usuarios (passwd debe poder leerse). Con el método de *shadowíng,* Linux guarda las contraseñas de los usuarios y deja una marca en /etc/passwd.

Añadir usuarios editando manualmente /etc/passwd.

Otra forma de añadir usuarios es editando manualmente /etc/passwd, para lo que se utiliza una herramienta especial: *vipw.*

vipw

Si se va a editar /etc/passwd manualmente, utilice *vipw* (abreviatura de *vi passwd). vipw bloquea passwd* mientras se realizan modificaciones, con lo que garantiza que los cambios se realizan de forma segura.

El editor predeterminado de *vipw es vi.*

Comandos de vipw:

Comando	Resultado
a	Indica a vi que empiece a añadir texto detrás del cursor. Este comando se introduce la primera vez que se inicia vipw. Si no se inicia, no aparecerá texto hasta que se pulse la "s" minúscula.
ctrl+b	Desplazarse hacia arriba página a página.
ctrl.+f	Desplazarse hacia abajo una página.
d	Si se pulsa una vez, suprime un carácter o un operador. Al pulsarlo dos veces, suprime toda la línea.
D	Suprime toda una línea.
I	Inicializa el modo de inserción, muy parecido a lo que se hace en ed.
x	Notifica a vi que elimine el carácter actual.
X	Notifica a vi que elimine el carácter inmediatamente anterior al cursor.
w	Permite saltar de una palabra a otra.
w:	Escribe los cambios en el archivo actual.
Mayús+p	Pega texto.
Mayús+h	Sitúa el cursor al comienzo del archivo (como la tecla Inicio cuando se utiliza en procesadores de texto).
Mayús+1	Le desplaza a la última línea del archivo.
w: nombre de archivo	Guarda los cambios en un archivo nuevo.
:wq	El comando para guardar y salir. Cuando acabe de realizar modificaciones, pulse la tecla Esc e introduzca este comando, y *vi* guardará los cambios y le devolverá a la shell.

Tabla 3.1 comandos de teclado de vipw

La primera vez que se carga vi, comienza en modo comandos. Mientras esté en este modo, vi reconocerá una amplia gama de comandos que llevan a cabo funciones de búsqueda, corte, pegado, supresión e inserción. Para pasar del modo comandos al modo edición, y viceversa, pulse la tecla Esc.
Si cree que vi es difícil de utilizar, utilice el editor predeterminado de vipw. Por ejemplo, podría utilizar pico si quisiera, ya que es mucho más sencillo y se comporta como un editor de DOS. En ese caso, debe cambiar la variable de entorno EDITOR. Para hacerlo en la *shell* de C, escriba este comando:

```
$ setenv EDITOR pico
```

Con ello se define que el editor es pico. A partir de ese momento, cuando llame a vip utilizará pico en su lugar. Para definir pico como el editor predeterminado de bash escriba este comando:

```
$export EDITOR=pico
```

Dependiendo del tipo de instalación que se elija, pico puede instalarse o no. Es una parte del paquete de cliente de correo pine.

3.6.4 Suprimir usuarios

A menos que el sistema emplee el *shadowing* de contraseña, los usuarios pueden eliminarse en dos pasos:

1. Elimine sus entradas de /etc/passwd.
2. Elimine su directorio de inicio (/home/username).

Cuando vaya a suprimir la entrada de un usuario de /etc/passwd, no olvide utilizar vipw. En caso contrario, puede eliminar el directorio de un usuario de la siguiente forma:

```
rm -r /home/username
```

3.6.5 Realiza tareas administrativas con su.

su, el usuario sustituto

EI comando su permite ejecutar una *shell* con varios UID y GID que no sean los suyos (siempre que conozca la contraseña correcta).Por ejemplo, ésta es una forma de convertirse temporalmente en *root*

```
$su
```

Linux le pedirá una contraseña. Si escribe la correcta, su le integrará en una *shell* como raíz

su tiene algunas opciones importantes en la línea de comandos, que se resumen a continuación

opción	propósito
-c [comando]	La opción -c se utiliza para enviar un comando a la *shell.* Aquí, se ejecuta el comando bajo el usuario que se especifique sin que sea necesario iniciar ninguna shell interactiva, lo que es útil cuando se desea ejecutar un solo comando bajo el UID.
--help	La opción --help se utiliza para obtener un breve resumen de las Opciones válidas de su.

-l o -login	La opción -l se utiliza para obtener una *shell* de conexión de su. Es ligeramente diferente a un su estándar, lo que proporciona el nuevo UID, pero realmente no inicia la sesión como el usuario especificado (por ejemplo, no lleva al directorio de inicio, como haría una conexión real). Cuando se utiliza la opción –l, su inicia una *shell* de inicio de sesión y, a continuación, lee y ejecuta los archivos de inicio del usuario.
-p	La opción -p se usa para preservar las variables de entorno actuales.
-s	La opción -s se utiliza para especificar una *shell* determinada durante una sesión.

Tabla 3.2 opciones de líneas de comando de su

sudo

El comando sudo permite a los usuarios elegidos ejecutar determinados comandos como si fueran *root.*

Aplicación: sudo

Necesita: sudo + /etc/sudoers + /etc/netgroups + visudo

Archivos de configuración: */etc/sudoers*

Los usuarios entran en el modo sudo escribiendo este comando:

```
$sudo
```

A continuación, sudo solicita una contraseña. Si el usuario escribe la correcta, entra. En caso contrario, sudo registra el intento de acceso.

sudo permite limitar de forma estricta los usuarios que pueden invocarlo y los comandos que dichos usuarios pueden ejecutar. Estos parámetros se especifican en

`/etc/sudoers`

/etc/sudoers se estructura en secciones:

Comandos que pueden ejecutar los usuarios de sudo.

Alias de los *host, incluyendo hosts,* grupos, direcciones IP y redes (si hay alguna).

Alias de los usuarios (si hay).

Especificaciones de los usuarios, incluyendo los tipos de *host,* las IP de los *hosts,* la lista de usuarios autorizados y el usuario como el que se ejecuta (normalmente, root).

Las listas se delimitan mediante comas. Éste es un ejemplo desglosado con marcadores:

```
# Sample /etc/sudoers file.
```

```
# This file MUST be edited with the 'visudo' command as root.
# See the man page for the details on how to write a sudoers file.
# User alias specification
# six users
User_Alias FULLTIMERS=[comma-delimited list of users]
User_Alias PARTTIMERS= [comma-delimited list of users]
```

Dado que sudoers es un archivo orientado a la seguridad (de forma parecida a /etc/passwd), hay que tomar precauciones especiales al editarlo. La distribución de sudo incluye una herramienta especial diseñada expresamente para este fin: visudo. Control de acceso

Modificar /etc/sudoers con visudo

visudo se parece mucho a vipw. Su finalidad es proporcionar un medio limpio y seguro de edición de /etc/sudoers. visudo bloquea sudoers durante la realización de modificaciones y, lo que es más importante, busca errores de sintáxis y no permitirá enviarlos al disco.

Control de acceso.

Control de acceso es cualquier técnica que otorga o deniega a los usuarios acceso a los recursos del sistema, entre los que se incluyen archivos, directorios, volúmenes, unidades, servicios, *hosts,* redes, etc.

Permisos y propiedad

En Linux, el acceso de los usuarios a los distintos archivos y directorios se limita mediante la concesión de permisos. Hay tres tipos básicos de permisos:

- De lectura: permite a los usuarios leer el archivo especificado.
- De escritura: permite a los usuarios modificar el archivo especificado.
- De ejecución: permite a los usuarios ejecutar el archivo especificado.

Cuando se asignan estos permisos, Linux guarda un registro de los mismos que
posteriormente aparece reflejado en las listas de archivos. El estado de los permisos de cada uno de los archivos se expresa mediante marcas. Las marcas de permiso son:

r: acceso de lectura.
w: acceso de escritura.
x: acceso de ejecución.

```
drwxr-xr-x   3 Nicole   Nicole        1024 Jun 2 1998 1g
-rwxrwxr-x  1 Nicole   Nicole         45 Apr 18 13:07 parse_out.pl
```

La columna de permisos tiene 10 caracteres.

Fig. 3.12 propiedades de la tabla de permisos.

Como se muestra, el primer carácter especifica el tipo de recurso. En este campo:

- representa un archivo.

b representa un archivo de bloques especial.

c representa un archivo de caracteres especial.

d representa un directorio.

l representa un enlace simbólico.

Los nueve caracteres restantes se dividen en tres grupos de tres:

- Los permisos del propietario: estos permisos muestran el acceso del propietario al archivo.
- Permisos de grupo: estos permisos muestran el acceso del grupo al archivo.
- Permisos mundiales: estos permisos muestran los derechos que tiene el resto del mundo a acceder a este archivo (si tiene algúno).

Vamos a aplicar esto al *script* de Perl, es posible ver que este recurso es un archivo.

De igual modo, los usuarios del grupo también pueden leerlo, escribirlo y ejecutarlos

-rwxrwxr-x 1 Nicole Nicole 45 Apr 18 13:07 parse_out.pl

Finalmente, aquellos que no sean Nicole y que no pertenezcan a su grupo sólo pueden leer y ejecutar el archivo, no pueden escribir en él.

Por tanto

El primer carácter indica el tipo de archivo, normalmente un archivo normal (-) o un directorio (d).

El primer conjunto de tres caracteres indica los privilegios del usuario.

El siguiente conjunto de tres caracteres indica los privilegios del grupo.

El último conjunto de tres caracteres indica los privilegios del resto de usuarios.

Estos permisos se establecen con el comando chmod

chmod: cambiar los permisos de los archivos

Para definir los permisos de un usuario determinado sobre un archivo o un directorio se utiliza chmod.el cual acepta tres operadores:

El operador - quita los permisos.

El operador + agrega permisos.

El operador = asigna permisos.

permisos que pueden quitar, agregar o asignar estos operadores.

Permiso de chmod	Explicación
r	El carácter r añade o quita el permiso de lectura. Ejemplo chmod +r nombre de archivo añade permiso de lectura a nombre de archivo
w	El carácter w añade o quita el permiso de escritura. Ejemplo chmod-w nombre de archivo elimina el permiso de escritura de nombre de archivo.
x	El carácter x añade o quita permiso de ejecución. Ejemplo chmod + x nombre de archivo añade el permiso de ejecución a nombre de archivo.

Tabla 3.3 permisos de chmod.

Un método consiste en añadir letras (r, w, x) para asignar permisos a archivos individuales y directorios. Otro es utilizar el sistema octal, donde se pueden añadir valores octales para crear un conjunto de permisos final.

El sistema octal

En el sistema octal, los números representan permisos. En seguida se resume el esquema octal y lo que representa cada número.

Valor octal	Explicación
0000	Equivale a --- o no hay ningún permiso.
0001	Equivale a --x o permiso de ejecución para el propietario del archivo.
0002	Equivale a --w- o solamente permiso de escritura para el propietario del archivo.
0004	Equivale a r-- o solamente permiso de lectura para el propietario del archivo.
0010	Equivale al permiso de ejecución para el grupo, donde el segundo conjunto de tres es --x.
0020	Equivale al permiso de escritura para el grupo, donde el segundo conjunto de tres es -w-.
0040	Equivale al permiso de lectura para el grupo, donde el segundo conjunto de tres es r--.

0100	Equivale al permiso de ejecución para el mundo, donde el segundo conjunto de tres es --x.
0200	Equivale al permiso de escritura para el mundo, donde el segundo conjunto de tres es -w-.
0400	Equivale al permiso de lectura para el grupo, donde el segundo conjunto de tres es r--.
1000	El modo 1000 es para el "difícil"; se aplica a directorios importantes (como /tmp). El bit difícil restringe la eliminación de los archivos a los propietarios del directorio o de los archivos que éste contiene, lo que permite crear directorios en los que pueden escribir todos los usuarios, aunque se puede evitar que puedan eliminar los archivos del resto de usuarios (estas restricciones se imponen aun cuando los permisos del archivo se hayan definido de forma distinta). Los directorios definidos con el bit difícil se identifican mediante una t en una gran lista, en contraposición a la d habitual.
2000	El modo 2000 aplica el bit SETGID.
4000	El modo 4000 aplica el bit SETUI D.

Tabla 3.4 valores octales.

Si se utilizan valores octales puros, hay que añadirlos juntos, lo que deriva un número final que expresa todos los permisos concedidos, para facilitar las cosas, es posible reducir rápidamente los permisos del propietario, de los grupos y de otros usuarios a un número de tres dígitos utilizando estos valores:

0 = Sin permisos.

1 = Ejecución.

2 = Escritura.

3 = Escritura y ejecución (actualmente no se utiliza mucho).

4 = Lectura.

5 = Lectura y ejecución.

6 = Lectura y escritura.

7 = Todo el conjunto: lectura, escritura y ejecución.

Para que puedan utilizarlo todos los usuarios, tiene que aplicar los permisos adecuados.

Podría hacer algo parecido a esto: chmod 751 myscript.cgi

En este caso, myscript.cgi lleva las siguientes restricciones de acceso:

- El propietario puede leerlo, escribirlo y ejecutarlo (7).
- El grupo puede leerlo y ejecutarlo (5).
- El mundo (usuarios externos) sólo pueden ejecutarlo (1).

Archivos con permisos especiales.

Hay dos permisos especiales para los archivos:

SGID (define el ID de grupo, 2000 octal o S).
SUID (define el ID de usuario, 4000 octal o S).

Los programas con permisos de SGID o SUID son especiales, ya que los permisos de su propietario se respetan aun cuando los ejecuten otros usuarios. Esto es, si se define el valor *root* SUID en un programa, éste siempre se ejecutará como *root,* aunque lo utilice un usuario normal. Este es el motivo por el que los archivos de SGID y SUID pueden suponer un riesgo para la seguridad.

Cuando se define el SUID/SGID de un directorio, los usuarios que pertenezcan al grupo autorizado pueden modificar exclusivamente sus propios archivos de dicho directorio.

Si los atacantes pueden explotar las debilidades de los programas root de SUID, potencialmente pueden obtener privilegios de root.

Los archivos SUID se pueden buscar con el siguiente comando: find /-perm +4000

En una instalación completa de Caldera Openlinux 1.1, esta búsqueda da como resultado 81 archivos.
Algunos de estos archivos representan serios agujeros en la seguridad. Por ejemplo, fíjese en esta entrada: */usr/lib/games/abuse/abuse.console*

Esta entrada (que también se encuentra en Red Hat 2. 1) puede ofrecer a los atacantes acceso a la *shell de root.*

Protegerse contra ataques basados en SUID y SGID.

Es posible protegerse contra dichos ataques con un método de cuatro flancos o con un sistema de selección:

- Pocos programas deben ser de SUID. Aquellos que deban serlo obligatoriamente deben tener su propio grupo.
- Asegúrese de que no se puede escribir en los *scripts* de SUID.
- En el caso de los programas de SUID que no necesiten imperiosamente que se defina el SUID, cambie sus permisos (chmod -s [programa]).
- En el caso de los programas de SUID que sean en gran parte inútiles o no esenciales (como los juegos en un equipo de la empresa), elimínelos o desinstálelos.

Algunos puntos vulnerables conocidos relacionados con SUID.

Desgraciadamente, no existe ninguna Esta universal de programas relacionados. Sin embargo, se muestra algunos problemas de los que se tiene conocimiento.

Programa	Detalles
/usr/bin/convfont	En algunos sistemas, /usr/bin/convfont es el root de SUID. Puede llevar a una *shell* de *root.*

Croad	En SlackWare 3.4, crond es vulnerable a un ataque cuyo resultado es una *shell root* de SUID. La solución es actualizarlo.
Cxterm	cxterm (SlackWare 3.1, 3.2) es *root* de SUID y necesita serlo. Sin embargo, es vulnerable a un desbordamiento de *buffer* que, cuando se explota, da como resultado una *root* de SUID. La solución es actualizarlo.
Deliver	deliver es una herramienta que distribuye correo remoto a destinatarios locales. En las versiones 2.0.12 y anteriores, deliver es vulnerable a un desbordamiento de *buffer* tanto en Debian como en SlackWare. Este hecho es importante, ya que deliver es *root* de SUID. La solución es actualizarlo.
dip3.3.7i	En SlackWare 2.1.0, dip (una utilidad para gestionar sesiones de PPP[81]) era setuid y ejecutable en todo el mundo.Además,dip3.3.7 o en SlackWare 3.4 es *root* de SUID y vulnerable. Solución: actualizarlo.
Dos	En los primeros paquetes de Debian, en el paquete DOSEMU (0.64.0.2-9), /usr/sbin/dos es *root* de SUID. La solución es eliminar el permiso de SUID.
Dump	dump (en Red Hat 2.1) es *root* de SUID. Solución: anular SUID.
Gnuplot	Algunas distribuciones de Linux (como SuSE 5.2) incluyen el *root de* SUID gnuplot. Éste es un ejemplo típico en el que un programa es *root* de SUID sin ninguna buena razón para ello. La solución: chmod -s /usr/bin/gnuplot.
Ideafix	Ideafix es un conjunto de herramientas de desarrollo. Dentro de dicho conjunto se encuentra el programa wm, que tiene un punto vulnerable que conduce a una *shell root* de SUID.
Protector de Pantallas de KDE	Los protectores de pantalla de K Desktop (KDE) 1.0 en Caldera Open- Linux ejecutaban el *root* de SUID.
Killmouse	killmouse (de Doom) ejecuta varios *scripts de SUID.* Solución: eliminar SUID (véase startmouse).
Kppp	kppp se incluye con K Desktop. Es una utilidad para configurar las redes de acceso telefónico en KDE. Es vulnerable a los desbordamientos y ejecuta el *root* de SUID. Solución: no lo ejecute en *el root de* SUID.
Libxt	Los programas creados con las bibliotecas compartidas X11R6 de XFree86 anteriores a la versión 3.3 pueden ser vulnerables a desbordamientos de buffer que pueden conducir a *root* en los archivos de SUID o de SGID. Solución: actualizar.
linuxconf	linuxconf (en Red Hat 5.1) *es root* de SUID. Solución: eliminar el permiso de SUID (chmod -s /bin/linuxconf.
s-povray	povray es un programa de *ray-tracing* para gráficos. En

[81] (Protocolo Punto a Punto) Se define mediante el RFC 1661 y nos proporciona un método para transmitir paquetes a través de enlaces seriales punto a punto.

	la versión 3.02 s-povray es *root* de SUID y según se informa debe estar para llevar acabo funciones de visualización.
startmouse	En varios sistemas (sobre todo SlackWare 3), startmouse (parte de la distribución del juego Doom) es *root* de SUID. La solución es ajustar los permisos.
suidexec	suidexec en Debian 2.0 (en el paquete suidmanager, 0.18) puede proporcionar acceso a *root* a través de *scripts de shell* de SUID.
wsmbconf	wsmbconf (parte de samba-1.9.18p10-3) ejecutaba el SGID que poseía *el root.*

Tabla 3.5 Debilidades conocidas de Linux relacionadas con SUID

3.6.6 Los grupos al detalle

Linux establece automáticamente los privilegios en los archivos que posee *root,* con lo que los protege de los usuarios habituales. Sin embargo, de vez en cuando, es posible que se vea obligado a proteger algún grupo de usuarios (y sus posesiones) de otro.

De esta forma, todo el mundo obtiene lo que necesita, pero una parte de la información sale de los límites sin ninguna solicitud especial. Este es el concepto básico que subyace tras los grupos.

3.6.7 Crear grupos

En los sistemas sin ocultación, los datos de los grupos se almacenan en /etc/group

etc/group y añadir nuevos usuarios

La estructura de /etc/group es parecida a la de /etc/passwd.

```
root: : 0:
wheel: : 1 0:
bin::1:bin, daemon
daemon::2:bin,daemon
sys::3:bin,adm
adm::4:adm,daemon
tty::5:
disk::6:
lp::7:daemon,lp
mem::8:
kmem::9:
operator::11:
mail::12:mail
news::13:news
```

```
uucp::14:uucp
man::15:
games::20:
gopher::30:
dip::40:

ftp::50:
users::100:amd,marty,dnb,manny,moe,jack,jill,stacy,Nicole
nobody::65534:
amd::500:amd
marty::502:marty
dnb::503:dnb
manny::504:manny
moe::505:moe
jack::506:jack
jill::507:jill
stacy::508:stacy
Nicole::509:Nicole
```

El archivo se compone de registros de grupos. Cada línea almacena un registro y cada registro se divide en cuatro campos delimitados por dos puntos **(:):**

Group name.

Group password.

Group ID (GID).

Group users.

Observará que todos los usuarios, se han colocado de forma predeterminada en el último campo

users:: 100:amd,marty,dnb,manny,moe,jack,jill,stacy,Nicole

Para añadir un grupo, modifique manualmente /etc/group e inserte una línea nueva que defina a dicho grupos.

Asignar el nuevo GID de forma secuencias. Por tanto, si el último GID ha sido 509, el nuevo debe ser 510.

Una vez creados los grupos, debemos asignar los propietarios de los archivos y de los directorios. Aunque todos los usuarios de los grupos tendrán los mismos derechos de acceso.

3.6.8 Utilizar herramientas gráficas para definir los propietarios, los permisos y los grupos

Es posible que utilice Linux exclusivamente en modo gráfico y que no se sienta cómodo con las líneas de comandos. No pasa nada. La mayoría de las distribuciones dominantes de Linux, en particular OpenLinux y Red Hat, incluyen utilidades con interfaz gráfica de usuario para establecer los permisos y las propiedades

Por ejemplo, Caldera **Openlinux** incluye un editor de permisos en el sistema de escritorio Looking Glass. Que se utiliza para que se cumpla la configuración de seguridad

El editor de preferencias es sencillo. Sin embargo, obliga a realizar cambios importantes en un directorio elegido o en el método de creación de archivos predeterminado. Si es posible, debe habituarse a establecer los permisos de forma manual.

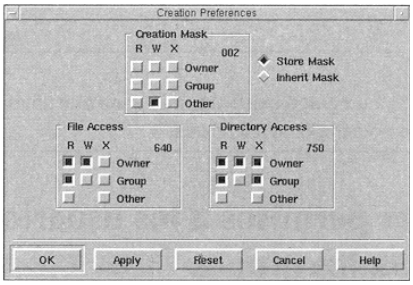


Fig. 3.13 El editor de preferencias de permisos looking Glass de Caldera Openlinux

3.6.9 Cómo se relacionan los usuarios con los grupos.

Es posible que se pregunte de qué forma los usuarios que ya tienen un grupo primario ejercen sus privilegios desde otro grupo; pues bien, lo hacen a través de del comando newgrp.

newgrp: cambiar el grupo actual

Los usuarios pueden pasar de un grupo a otro durante la misma sesión, con un nuevo inicio de sesión utilizando el comando newgrp. La sintaxis del comando es:

```
$newgrp [group]
```

Siempre que el usuario sea miembro de un grupo, funciona perfectamente.

3.6.10 Eliminar grupos

Para eliminar un grupo, suprima su entrada en /etc/group/
Cuando se suprime un grupo, también se elimina su GID. Habitualmente, esto no supone ningún problema, ya que los grupos que se crean suelen ser grupos especiales, independientes y bien diferenciados de los grupos principales y predeterminados del usuario, no siempre ocurre esto. A veces, el grupo que se elimina es también el grupo principal o predeterminado de uno o varios usuarios. Por consiguiente, antes de eliminar cualquier grupo, anote su GID. Posteriormente, compruebe /etc/passwd.

3.6.11 Desconectar el sistema

Linux necesita tiempo para cerrar los procesos abiertos, guardar los datos no guardados en el disco y realizar una limpieza.

shutdown: apagar el sistema Linux

Para apagar el sistema Linux, utilice el comando shutdown. Está especialmente diseñado para desconectar Linux de forma segura. Durante este proceso, shutdown realiza las siguientes acciones:
Notificar a los restantes procesos y usuarios que el apagado es inminente.
Apaga otros procesos que aún se están ejecutando.
Notifica a *root* a medida que se desconecta cada servicio.
reinicia el sistemas
shutdown admite varias opciones de la línea de comandos

Opción	Propósito
-c	La opción -c se utiliza para cancelar un apagado que ya estaba programado.
-h	La opción -h se utiliza para forzar una detención de todo el sistema tras apagarse el sistema.
-k	La opción -k se utiliza para simular un apagado y enviar mensajes de apagado a los usuarios sin que realmente se apague el sistema.
-r	La opción -r se utiliza para forzar un reinicio tras apagarse el sistema.
-t [segundos]	La opción -t se utiliza para establecer el tiempo, en segundos, antes de que shutdown realmente realice su tarea (enviar señales, apagar procesos, etc.).

Tabla 3.6 opciones para shutdown

Para apagar inmediatamente el sistema y reiniciarlo escriba el siguiente comando:

shutdown -r now
El valor de tiempo también se puede expresar de forma mas concreta en minutos (shutdown -r +minutos) o en horas (shutdown -r 12:24).

3.7 Seguridad de los usuarios de Linux.

3.7.1 Ataques a contraseñas

El término describe diversas actividades, entre las que se incluye cualquier acción dirigida a romper, descifrar o borrar contraseñas o a sortear de cualquier otra forma los mecanismos de seguridad de las contraseñas.

Actualmente, cualquiera puede romper contraseñas de Linux utilizando herramientas automatizadas.

Una deficiente seguridad de las contraseñas pone en peligro a todo el sistema. Los atacantes que inicialmente obtienen sólo acceso limitado pueden extender rápidamente dicho acceso mediante el ataque a una seguridad de contraseñas débil. A menudo, con meros ataques a contraseña, los atacantes obtienen acceso como *root* y arrebatan el control no sólo de un *host* sino de varios.

Existen varias técnicas de ataque a contraseña como los pasos necesarios para protegerlas entre los que se incluyen:

- Instalación del shadowing de contraseña.
- Refuerzo de contraseñas en aplicaciones de terceros.
- Refuerzo del sistema frente a ataques a contraseña.
- Desarrollo de políticas efectivas de contraseñas.

Cómo genera y almacena Linux las contraseñas

Muchas de las primeras distribuciones de Linux almacenaban las contraseñas de los usuarios en /etc/passwd, lo que no resultaba seguro, ya que /etc/passwd es (y debe ser) legible. De ahí que cualquier usuario pueda ver los contenidos de /etc/passwd simplemente concatenándolo

```
$cat /etc/passwd
root:80zrR2ac.IEGY:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
nobody:*:65534:65534:Nobody:/:/bin/false
 Las contraseñas ocupan el segundo campo:
bwagner:..CETo68esYsA:501:501:Bill      Wagner:/home/bwagner:/bin/bash
marty:jvXHHBGCK7nkg:502:502:Marty           Rush:/home/marty:/bin/bash
dnb:ilYD6CckS.JIA:500:503:Caldera  OpenLinux  User:/home/dnb:/bin/bash
manny:bj2NcvrnubUqU:503:504:Caldera                              OpenLinux
User:/home/manny:/bin/bash           moe:IK4OBb5NnkAHk:504:505:Caldera
Openlinux  User:/home/moe:/bin/bash  jack:FL.OtOVxVegL.:505:506:Caldera
OpenLinux                                    User:/home/jack:/bin/bash
jill:JMpkhgZrXePnM:506:507:Caldera                               Openlinux
User:/home/jill:/bin/bash          stacy:OOfE8weNKJUFw:507:508:Caldera
Openlinux  User:/home/stacy:/bin/bash  Alex:ylRWmr3zbhms6:509:100:Alex
Brittain:/home/Alex:/bin/bash     Nicole:zKQR.cqTgzkco:508:509:Caldera
OpenLinux User:/home/Nicole:/bin/bash
User:/home/Nicole:/bin/bash
```

Observe que las contraseñas están alteradas de modo que resulten incompresibles. Han sido sometidas a criptografía.

Ataques a diccionario

Las contraseñas de Linux codificadas con DES pueden romperse rápidamente, habitualmente en cuestión de minutos. Existen dos razones principales para ello:

a) El factor humano: los usuarios eligen invariablemente contraseñas débiles.

b) Longitud limitada: las contraseñas de Linux son cortas. El número de transformaciones necesarias para cifrarlas es relativamente pequeño.

En los ataques a diccionario, los atacantes toman diccionarios (grandes listas de palabras),y los codifican utilizando DES. Durante este proceso, envían palabras corrientes, nombres propios y otro texto precisamente a través de las mismas permutaciones y transformaciones a las que se exponen las contraseñas de Linux. Con el paso del tiempo, utilizando herramientas de

ruptura de alta velocidad, los atacantes pueden codificar cada palabra del diccionario de 4.096 formas diferentes. Cada vez que una herramienta de ruptura obtiene dicho texto codificado, lo compara con las contraseñas de /etc/passwd. Encuentra una coincidencia y, cuando esto ocurre, comunica al agresor que se ha roto una contraseña.

Crack

Aplicación: Crack

Necesita: C + *root* (y Perl si se lleva a cabo una ruptura en paralelo o multiproceso)

Archivos de configuración. *dictgrps.conf, dictrun.conf, network.conf*

Para ejecutar Crack, hay que ser *root.* Tenga en cuenta que si le descubren ejecutando Crack sobre archivos de contraseñas de otros compañeros se meterá en un lío, porque es ilegal. Aunque sea el administrador de un sistema, puede encontrarse con problemas, así que debe asegurarse de tener la autorización adecuada antes de poner a prueba o romper un sistema de contraseñas. Crack es la herramienta de auditoria de contraseñas más conocida en la comunidad UNIX. Crack se encuentra actualmente en su versión 5.0a.

Descomprimir Crack

Tras haber descargado Crack, colóquelo en un directorio apropiado para descomprimirlo. Como en /root. A continuación, descomprima el archivo Crack utilizando gunzip:

```
$ gunzip.crack5.0.tar.gz
```

De esta forma, lo descomprimirá a un archivo denominado crack5.0.tar, un archivo de tipo tar. Descomprima este archivo usando el comando tar, de la siguiente forma:

```
$ tar -xvf crack5.0.tar
```

A continuación, verá pasar por la pantalla muchos nombres de archivos y directorios. Crack está ocupado descomprimiendo un directorio denominado c5Oa/. Dependiendo de la carga y de los recursos del sistema, este proceso puede tardar más o menos tiempo. Cuando Crack haya acabado de descomprimirse, cambie al directorio c5Oa/.

Crear Crack

Para crear Crack, escriba la siguiente línea de comando:

```
$./Crack -makeonly
```

De nuevo, verá pasar muchos mensajes por la pantalla mientras se compila Crack. Este proceso puede durar unos 10 minutos. Si el sistema compila Crack correctamente, verá el siguiente mensaje:

```
all made in util
```

```
make[1]: Leaving director '/rot/c50a/src/util'
Crack: makeonly done
```

A continuación, debe hacer que Crack compile los diccionarios. Para ello, introduzca el siguiente comando:

```
$ Crack -makedict
```

Este proceso tardará algún tiempo. Cuando haya terminado, Crack mostrará el siguiente mensaje:

```
Crack: Created new dictionary...
Crack: makedict done.
```

Ejecutar Crack

Crack puede romper el archivo /etc/Passwd directamente, por lo que no es necesario copiar los registros de las contraseñas en otro archivo. Sin embargo, para tener todo junto, recomendamos copiar /etc/passwd en passwords.txt del directorio /cd50a:

$ cp /etc/passwd passworels.txt

Para ejecutar Crack, introduzca el comando Crack, más las opciones, más el nombre del archivo que contiene las contraseñas, como sigue:

$ Crack passwords.txt

Se iniciará Crack y mostrará un informe inicial.
Tras iniciarse, Crack se ejecuta como un proceso en segundo plano a menos que se especifique lo contrario. Se le Puede seguir la pista utilizando el comando ps. Estas son algúnas salidas típicas:

```
1175 2 S N 0:04 cracker -kill run/Ksamshacker.sams.net.1092

1178 2 Z N 0:00 (kickdict <zombie>)
```

A medida que funciona, Crack aplica muchas reglas a cada palabra. Las reglas son las distintas formas posibles en que puede haberse escrito una contraseña. Por ejemplo:

a. Alternar mayúsculas con minúsculas.
b. Escribir la palabra hacia delante y hacia atrás y concatenar ambos resultados.
c. Repetir una palabra una, dos o varias veces.
d. Añadir el número 1 al comienzo o al final de cada palabra

A continuación algúnas de las reglas que emplea Crack.

Regla	Resultado
append:$X	Se añade el carácter X al principio de la palabra actual.
capitalise: c	Pone la primera letra en mayúscula.
dfirst:[	Borra el primer carácter de la palabra actual.

dlast:[	Borra el último carácter de la palabra actual.
duplicate : d	Deletrea la palabra actual dos veces y las funde.
lowercase: l	Pone en minúscula la palabra actual.
ncapital: C	Pone en minúscula la primera letra y en mayúscula el resto.
pluralise: p	Pone en plural la palabra actual.
reflect: f	Escribe la palabra actual primero hacia delante, luego hacia atrás y las funde.
reverse: r	Escribe al revés la palabra actual.
togcase: t	Invierte las mayúsculas y minúsculas.
uppercase: u	Pone en mayúscula la palabra actual.

Tabla 3.7 reglas habituales del crack

También se pueden tener a la vista Crack y la regla que se utiliza actualmente observando los archivos de progreso de /c50a/run.

Ver los resultados.

Para ver si Crack ha adivinado correctamente nuestras contraseñas, utilice la herramienta Report, que se encuentra en /c50a, de la siguiente forma:

```
$./Reporter
```

Ésta es la salida de la sesión del ejemplo:

```
Guessed marty [marty] Marty Rush [ passwords.txt /bin/bash]
Guessed Nicole [alexalex) Caldera OpenLinux User [passwords.txt
/bin/bash]
Guessed manny [willow] Caldera OpenLinux User [passwords.txt
/bin/bash]
Guessed moe [solace] Caldera OpenLinux User [passwords.txt /bin/bash]
```

Crack ha encontrado cuatro contraseñas, tarea en la que ha invertido unos dos minutos y es que las contraseñas elegidas eran demasiado débiles.

Opciones de la línea de comandos de Crack

Crack admite varias opciones en la línea de comandos. Se resume las más comúnmente utilizadas

Opción	Propósito
-debug	Proporciona información estadística e informes de los procesos en tiempo real.
-fgnd	Se utiliza para ejecutar Crack en primer plano, de forma que se puede observar lo que hace el proceso.
-from N	Se utiliza para ejecutar Crack a partir de un número de regla determinada, representado por el número N.

-mail	Se usa para que Crack envíe un e-mail a todos los usuarios cuyas contraseñas se hayan forzado. De esta forma, se les notifica de manera inmediata que sus contraseñas eran débiles. El mensaje de aviso se puede personalizar, editándolo en e50a/scripts/nastygram.
-network	Se utiliza para ejecutar Crack en modo de red, donde se pueden auditar las contraseñas utilizando varias máquinas a la vez. Para personalizar el funcionamiento de la red.
-nice	Se utiliza para designar Crack como un proceso de baja prioridad, lo que permite a procesos de mayor prioridad utilizar la CPU siempre que la necesiten.
-recover	Se utiliza cuando estamos reiniciando el proceso Crack debido a un fallo o una terminación anormal. Esto protege las construcciones de bibliotecas que ya están disponibles.

Tabla 3.8 Líneas de comando de crack

Accesorios de Crack: listas de palabras

La caja de herramientas de Crack no estaría completas sin listas de palabras (o diccionarios). Las listas de palabras son simplemente listas de palabras, normalmente una por línea, en formato ASCII, que se pueden incorporar al sistema de diccionarios de Crack para ampliar el ámbito de ataque de los diccionarios. Tenga en cuenta que cuanto mayores sean las listas de palabras, más tiempo tardará Crack en completar una pasada. Sin embargo, también se incrementarán las posibilidades de dar con una contraseña
Crack incluye listas de palabras prefabricadas que se pueden utilizar en la mayoría de las auditorias de contraseñas de escasa importancia. Sin embargo puede agregarle algúnas que hay en Internet que contienen temas sobre computación, literatura, cine y televisión, nombres propios, geográficos, términos religiosos, términos científicos, diccionarios en varios idiomas.
Crack es bastante rápido, pero depende en gran medida del hardware. Sin duda alguna, la configuración ideal es un equipo a 400 MHz con 256 MB de RAM. Desafortunadamente, no todo el mundo tiene esta potencia. Sin embargo, en aquellos sistemas en los que los usuarios no eligen bien sus contraseñas, es probable que vea que se han roto las contraseñas de todos los usuarios en una hora.

Alternativas a Crack

Estas son otras herramientas de auditoria de contraseñas de DES que utiliza Unix-Linux.

Herramienta	Descripción
John the Ripper	Una herramienta de auditoría de contraseñas de propósito general para DOS, Windows y UNIX Sin embargo, pese a que John maneja contraseñas estilo DES, no utiliza el enfoque crypt. En su lugar, utiliza algoritmos propios. No obstante, John es rápida, admite muchas reglas y opciones y está bien documentada.
Killer Cracker	Una herramienta para auditorias de contraseñas de poca importancia creada por el Doctor Dissector en C++. Pese a que Killer Cracker carece de algúnas funcionalidades extendidas disponibles en Crack, sigue siendo rápida.
Lard	Una herramienta de auditoria de contraseñas para Linux y otras versiones de

	UNIX. Lard es lo suficientemente pequeña para caber en un disquete, lo que es útil para auditar equipos no conectados en red, de diferentes departamentos, etc.
PeriCrack	Es un intruso de contraseñas DES de Perl para Linux
Xcrack	Un *script* en Perl para romper contraseñas de Linux. No utiliza reglas complejas, sino que ejecuta un cifrado completo del archivo de diccionarios. Es útil para entornos en los que se espera que los usuarios hayan hecho elecciones de contraseñas excepcionalmente malas.

Tabla 3.9 herramientas de auditoria de contraseña

Algúnas herramientas no ofrecen simplemente ataques a diccionarios, sino ataques por fuerza bruta que prueban todas las posibles combinaciones. Éste es un proceso aparentemente indiscriminado y en algunos casos realmente lo es. Sin embargo, las rutinas de fuerza bruta están diseñadas para probar primero las combinaciones más probables.
No obstante, la mayor diferencia entre estos dos enfoques es que los ataques por fuerza bruta finalmente siempre acaban por imponerse ya que este proceso puede tardar varios meses. Como se podría esperar, los ataques por fuerza bruta llevan su tiempo. Por el contrario, los ataques a diccionarios son tan útiles como lo sean la lista de palabras y las reglas.

Shadowing de contraseñas y la suite shadow

El shadowing de contraseñas es una técnica mediante la que el archivo /etc/passwd sigue siendo legible pero ya no contiene las contraseñas. En su lugar, las contraseñas de los usuarios se almacenan en /etc/shadow.
Hay varias herramientas que realizan el *shadowing,* pero la más popular es *Linux*
Password Shadow Suite ,que lleva años utilizándose. Sin embargo, dependiendo del tipo y antigüedad de la distribución, puede tenerla o no. Para comprobarlo, examine /etc/passwd. Si contiene las contraseñas cifradas en bruto en el segundo campo, el paquete de shadow no está instalado en ese caso, hay que visitar la FTP o el sitio Web del proveedor (o revisar el CD-ROM) para obtener e instalar el paquete.
Tras haber instalado el paquete de shadow y haber comprobado que están todas las utilidades de shadow, examine la base de datos de contraseñas de shadow. /etc/shadow es el punto central de la *suite* shadow.

/etc/shadow: la base de datos de contraseñas de shadow.

Es un archivo especial que almacena no sólo las contraseñas de los usuarios sino también indicadores de reglas especiales.
Desde varios puntos de vista, /etc/shadow se asemeja a /etc/passwd. El archivo consta de un registro por línea y cada registro se divide en nueve campos separados por dos puntos (:):

- El nombre de usuario.
- La contraseña de usuario.
- El número de días desde 1 de enero de 1970, fecha en que se cambió la contraseña por última vez.

- El número de días que quedan antes de que se permita al usuario cambiar su contraseña.
- El número de días que quedan antes de que el usuario tenga que cambiar su contraseña.
- El número de días de anticipación con que se avisa al usuario de que pronto tendrá que cambiar su contraseña.
- El número de días que quedan para que el usuario cambie su contraseña antes de que su cuenta sean canceladas.
- El número de días desde el 1 de enero de 1970 que la cuenta ha sido canceladas.
- El último campo está reservado.

Utilizando estos valores, la *suite* shadow implementa dos nuevos conceptos del mantenimiento básico de las bases de datos de contraseñas:

Vencimiento de la contraseña: es cuando limitamos las contraseñas a un tiempo de vida finito, por ejemplo, 90 días. Cuando este tiempo se acaba, Linux obliga a los usuarios a crear nuevas contraseñas. Si se utiliza el vencimiento de contraseñas junto con la comprobación proactiva de las mismas la seguridad mejora.

Bloqueo automático de cuenta: avisar simplemente a los usuarios de la necesidad de cambiar sus contraseñas es poco realista. Los usuarios son perezosos y propensos a olvidarlo. Lo mejor es bloquear sus cuentas si se niegan a cooperar, pero hacerlo manualmente consume mucho tiempo. Con *la suite* shadow no hay que preocuparse, ya que el bloqueo se efectúa automáticamente.

La *suite* shadow consta de múltiples utilidades para la gestión de usuarios, grupos y contraseñas.

Utilidad	Función
chage	Un comando nativo de la *suite* shadow. chage se utiliza para cambiar la información de expiración de contraseña de los usuarios, como el número de días entre cambios de contraseñas y la fecha en que se cambió la contraseña por última vez.
chfn	Una sustituta de la *suite* shadow para la utilidad chfn estándar de Linux. chfn permite a los usuarios cambiar su información de finger (por ejemplo, sus nombres reales).
chsh	Una sustituta de la *surte* shadow para la utilidad chsh estándar de Linux. chsh es una utilidad que permite a los usuarios cambiar su *shell* predeterminada.
gpasswd	Un comando nativo de la *suite* shadow. Se utiliza para añadir nuevos usuarios a los grupos.
groupadd	Un comando nativo de la *suite* shadow. Se utiliza para añadir nuevos *grupos.*
groupdel	Un comando nativo de la *suite* shadow. Se utiliza para borrar grupos.
groupmod	Un comando nativo de la *suite* shadow. Se utiliza para modificar la información de los grupos.

grpck	Un comando nativo de la *suite* shadow. Se utiliza para realizar la verificación de los campos y la sincronización entre /etc/group y /etc/gshadow. Compárese con pwchk, que verifica /etc/passwd frente a /etc/shadow.
id	Un sustituto de la *suite* shadow para el comando id estándar de Linux. id es una utilidad que muestra el UID (Id. del usuario) y la información asociada.
login	Un sustituto de la *suite* shadow para el login estándar de Linux. Cuando un usuario inicia una sesión, login debe interactuar con la base de datos de contraseñas. Esta base de datos de shadow está estructurado de forma diferente, de ahí que sea necesaria una sustitución de login.
newgrp	Un sustituto de la *suite* shadow para el comando newgrp estándar de Linux. Los usuarios pueden cambiar de un grupo a otro (durante la misma sesión, después de volver a iniciar la sesión) utilizando el comando newgrp.
passwd	Un sustituto de la *suite* shadow para el comando passwd estándar de Linux. passwd sirve para crear nuevas contraseñas de usuario o para cambiar las existentes. La base de datos de contraseñas de shadow está estructurado de forma diferente, de ahí que sea necesaria una sustitución de passwd.
pwck	Un comando nativo de la *suite* shadow. Se utiliza para realizar la verificación de los campos y la sincronización entre /etc/shadow y /etc/passwd. Compárese con grpchk, que verifica la información de los grupos.
pwconv	Un comando nativo de la *suite* shadow. Se utiliza para fusionar los viejos registros de /etc/passwd en una nueva base de datos de shadow.
pwunconv	Un comando nativo de la *suite* shadow. Se utiliza para separar información de/etc/shadow y volver a convertirla al formato /etc/passwd.
su	Un sustituto de la *suite* shadow para el su estándar de Linux. El comando su permite ejecutar una *shell* con UID y GID que no sean los propios, siempre que se conozca la contraseña correcta, lo que es útil para conceder a usuarios corrientes derechos administrativos parciales o totales.
userdel	Un comando nativo de la *suite* shadow. Se utiliza para borrar usuarios(userdel -r jsprat). Este comando borrará al usuario jsprat y su directorio de origen.
usermod	Un comando nativo de la *suite* shadow. Se utiliza para cambiar la información de un usuario (su *shell,* el tiempo de expiración de la contraseña, etc.).

Tabla 3.10 Utilidades de la suite shadow

Añadir usuarios en sistemas con shadowing: useradd.

Para añadir un usuario a un sistema de contraseñas con *shadowing,* se utiliza la utilidad useradd, que gestiona las entradas de /etc/passwd, /etc/group y /etc/shadow.
Aplicación: useradd (/user/sbin/useradd)
Necesita: useradd.
Archivos de configuración: ninguno. Forma parte del paquete shadow.

Opción	Propósito
-b	Esta opción casi nunca se utiliza. Se usa para especificar un directorio inicial para aquellos usuarios que no tienen directorio de inicio. (En ,otras palabras, éste será el primer directorio al que Irán cuando inicie la sesión.)
-c [comentario]	Esta opción se Utiliza para especificar el nombre real del usuario o, alternativamente un comentario. (El texto que escriba rellenará el gecos o campo de en /etc/passwd.)
-d[dir]	Esta opción se utiliza Para especificar el directorio de inicio del nuevo usuario.
-e [fecha de expiración]	Esta opción se utiliza Para especificar la fecha en que expirará la contraseña del nuevo usuario. Para esto se puede utilizar casi cualquier formato de fecha estándar, incluyendo MM/DD/YY, o incluso el formato largo, como en 1 de enero de 2000. Sin embargo, si se utiliza este formato, o cualquier otro que incluya espacios en blanco, la fecha debe escribirse entre comillas. Considere forzar la expiración al menos cada 90 días.
-f [inactivity-lockout]	Esta opción se utiliza Para especificar cuántos días Pueden pasar sin que el usuario se conecte antes de que la cuenta sea cancelada. Este valor debe expresarse en días. Por ejemplo, -f 90 bloqueará la cuenta después de 90 días de inactividad.
G [grupo adicional]	Esta opción se utiliza Para asignar el usuario a grupos adicionales, además de su grupo primario.
-g [grupo]	Esta opción se utiliza Para asignar el usuario a un grupo específico. Éste será su grupo Primario, al que Pertenecerá siempre.
-m	Esta opción se utiliza para que useradd cree el directorio de inicio del nuevo usuario.
-s [shell]	Esta opción se utiliza para especificar la *shell* predeterminada del nuevo usuario (normalmente, /bin/bash).
u [uid]	Esta opción se utiliza para especificar el UID del nuevo usuario

Tabla 3.11 Opciones de líneas de comando de useradd

Si llama a useradd sin argumentos, aparece un resumen de uso:

```
usage: useradd [-u uid] [-g group] [-m] [-d homel [-s shell] [-r
rootdir] [-e expire dd/mm/yyyy] [-f inactive] name
useradd -D
useradd -v
```

Ésta es una línea de comandos mínima que creará una entrada de usuario en /etc/passwd, /etc/group y /etc/shadow:

```
/usr/sbin/useradd jsprat -m -c"jack Sprat" -u510 –g100 –s/bin/bash
```

En /etc/passwd, se añade jsprat a la lista de usuarios, junto con sus UID, GID, nombre real, origen y *shell*:

```
bigdave:x:100:100:Big                    Dave:/home/bigdave:/bin/bash
jackie:x:101:100:Jackie:/home/jackie:/bin/bash
jsprat:x:510:100:Jack Sprat:/home/jspratl:/bin/bash
```

En /etc/shadow, también se añade jsprat a la lista de usuarios. Sin embargo, observe que su contraseña no se ha generado automáticamente:

```
postgres:*:10713:0::7:7::
nobody:*:10713.0::7:7::
bigdave:aNi7cQR3XSTmc:10713:0::7:7::
jackie:7PbiWxVa5ArgE:10713:0:-1:7:-1:-1:1073897392
jsprat:*not set*:10715:0:-1:7:-1:-1:
```

Recuerde esto cuando vaya a crear nuevos usuarios: useradd no genera contraseñas. En su lugar, debe generar las contraseñas del usuario después de haber creado su cuenta El procedimiento para esto es exactamente el mismo que el de creación de la contraseña de un usuario en un sistema sin shadowing. Utilice el comando passwd:

```
[root@linuxbox2/rootl# passwd jsprat
Enter new UNIX password
Retype new UNIX password
passwd: all authentication tokens updated successfully
```

Más adelante, cuando consulte /etc/shadow, observará que se ha actualizado la información de la contraseña del usuario:

```
bigdave:aNi7cQR3XSTmc:10713:0::7:7::
jackie:7PbiWxVa5ArgE:10713:0:-1:7:-1:-1:1073897392
jsprat:cALtUMRf4OVbU:1071 5:0:-1:7:-I:-1:1073897392
```

Transferir archivos de inicio: /etc/skel

Cuando un usuario inicia una sesión, Linux lee la información sobre el entorno de uno o varios archivos de inicio y a continuación almacena copias originales de estos archivos en /etc/skel. He aquí un listado típico de /etc/skel:

```
$ ls -al /etc/skel
drwxr-xr-x    4 root    root     1024 May 2 13:32.
drwxr-xr-x   23 root    root     3072 May 3 22:18..
-rw-r--r--    1 root    root     49 Nov 25 1997.bash_logout
.rw-r--r--    1 root    root     913 Nov 24 1997.bashrc
-rw-r--r--    1 root    root     650 Nov 24 1997.cshrc
```

Tras crear la cuenta de un nuevo usuario, copie estos archivos al directorio de inicio del usuario y cambie su propietario y grupo en consecuencia. Si los deja en su estado original, todavía serán propiedad del *root* y el usuario no podrá utilizarlos.

Borrar usuarios en sistemas con shadowing: userdel

Para borrar un usuario en un sistema con *shadowing* se utiliza userdel. Esto suprime la información del usuario de /etc/shadow, /etc/passwd y /etc/group y, normalmente, la borra del todo.
Aplicación: userdel.
Necesita: userdel
Archivos de configuración: ninguno. Forma parte del paquete shadow.

Para borrar un usuario con userdel, introduzca el siguiente comando: $ userdel -r username
La opción -r borra el directorio de inicio del usuario, lo que resulta muy útil.
Tenga en cuenta que cuando vea el archivo /etc/skel, deberemos utilizar la opción -a porque la mayoría de los archivos son archivos punto. Estos archivos punto no aparecen en la salida simple del listado
ls -l.

Modificar el registro de un usuario existente en sistemas con shadowing: usermod

Para modificar el registro de un usuario existente, utilice usermod.
Aplicación: usermod.
Necesita: usermod
Archivos de configuración: ninguno. Forma parte del paquete shadow.

Opción	Propósito
-C [comentario]	Esta opción se utiliza para modificar la información del campo gecos del usuario (su nombre real).
-d [directorio de inicio]	Esta opción se utiliza para modificar el directorio de inicio del usuario.
-e [fecha de expiración]	Esta opción se utiliza para modificar la fecha de expiración de la contraseña de usuario.
-f [bloqueo por inactivada]	Esta opción se utiliza para modificar los parámetros de bloqueo por inactividad de la cuenta del usuario.
-g [grupo inicial]	Esta opción se utiliza para modificar los datos de pertenencia al grupo inicial del usuario.
-G [otros grupos]	Esta opción se utiliza para modificar los datos de pertenencia del usuario a otro grupo.
-l [nombre de usuario]	Esta opción se utiliza para modificar el nombre de inicio de sesión del usuario.
-s [shell predeterminada]	Esta opción se utiliza para modificar la *shell* predeterminada del usuario.
-U [UID]	Esta opción se utiliza para modificar el UID del usuario

Tabla 3.12 Opciones de la línea de comandos de usermod

Verificar la base de datos de contraseñas: pwchk

No hay duda de que con el tiempo realizará numerosos cambios en la base de datos de contraseñas. Dado que existe un potencial riesgo de errores que se incremento con el tiempo, debe verificar periódicamente la integridad de la base de datos de contraseñas, para lo que se utiliza pwchk.
Aplicación: pwchk.
Necesita: pwchk
Archivos de configuración: ninguno.
pwchk verifica que toda la información de /etc/passwd y de /etc/shadow es válida. Se asegura de que el usuario y los grupos son válidos y de que tienen

shells de inicio de sesión válidas, de que todos los campos están presentes y justificados y de que todos los usuarios tienen un grupo apropiado y un UID única.

Añadir un grupo en sistemas con shadowing: groupadd

Para añadir un grupo se usa la utilidad groupadd.
Aplicación: groupadd.
Necesita: groupadd.
Archivos de configuración: ninguno
Opciones de la línea de comandos de groupadd

Opción	
-g *[id del grupo]*	La opción -g se utiliza para especificar el GID.
-o	La opción -o es suplementaria. Se utiliza cuando se desea crear un GID que no sea único.

Tabla 3.13 Opciones de línea de comando de groupadd.

Los cambios realizados con groupadd quedan reflejados en /etc/group

Modificar la información de un grupo en un sistema con shadowing: groupmod

Para modificar la información de un grupo se utiliza el comando groupmod.
Aplicación: grouprnod.
Necesita: groupmod.
Archivos de configuración: Ninguno.
groupmod admite tres opciones de línea de comando que se resumen a continuación.
Opciones de la línea de comandos de groupmod

Opción	Propósito
-g [id del grupo]	Esta opción se utiliza para modificar el GID.
-n [nombre del grupo]	Esta opción se utiliza para modificar el nombre del grupo.
-o	Esta opción es suplementaria. Se utiliza cuando se desea crear un GID no único.

Tabla 3.14 Opciones de la línea de comandos para groupmod.

Los cambios realizados con groupmod quedan reflejados en /etc/group.

Borrado de grupos en sistemas con shadowing: groupdel.

Para borrar un grupo se usa la utilidad groupdel.
Aplicación: groupdel.
Necesita: groupdel,
Archivos de configuración: ninguno.
groupdel admite un solo argumento: el nombre del grupo. He aquí un ejemplo:
$ groupdel contabilidad

Esto borrará el grupo contabilidad.

Gestionar el acceso a grupos: gpasswd

En algún momento, deseará asignar administradores de grupos a grupos de usuarios. Un administrador de grupos es alguien que puede añadir o eliminar usuarios del grupo que está administrando. Además, es posible que desee limitar el acceso a los grupos e, incluso, protegerlos mediante contraseña. Para ello, se usa la utilidad gpasswd.

Aplicación: gpasswd.

Necesita: gpasswd

Archivos de configuración: ninguno.

Opciones de la línea de comandos de gpasswd

Opción	Propósito
-A [nombre de usuario del administrador]	Esta opción se utiliza para especificar un administrador de grupo que se identifica por su nombre de usuario. Por ejemplo, gpasswd -A jsprat contabilidad hace que jsprat sea administrador del grupo contabilidad.
-a [nombre de usuario]	Esta opción se utiliza para añadir un usuario a un grupo.
-d [nombre de usuario]	Esta opción se utiliza para borrar un usuario de un grupo.
-m [nombre de usuario del miembro]	Esta opción se utiliza para especificar miembros.
-r [grupo]	Los administradores de grupo utilizan esta opción para quitar una contraseña de grupo.
-R [grupos]	Esta opción se utiliza para desactivar el acceso a los grupos a través del comando newgrp. Los cambios hechos con gpasswd quedan reflejados en /etc/group.

Tabla 3.15 Opciones de comando de gpasswd

Verificación de datos de los grupos: grpchk

Dado que existe el riesgo potencial de cometer errores y que se incremento con el tiempo, debe verificar periódicamente la integridad de la información de los grupos, para lo que se utiliza el comando grpchk sin argumentos (grpchk) o, si se prefiere, en modo de sólo lectura (grpch k -r).

Aplicación: grpchk.

Necesita: grpchk.

Archivos de configuración: ninguno.

grpchk examina los datos de los grupos buscando posibles errores en el número
de campos y en la validez de sus nombres, sus usuarios y sus administradores. Si grpchk encuentra dichos errores, le solicita que los corrija.
Si prevé que grpchk va a encontrar errores, quizá deba iniciarlo, ya que ciertos errores provocan que grpchk borre todo un registro. Antes de hacerlo, es conveniente examinar manualmente dicho_registro. Quizá pueda reparar el daño sin eliminar todo el registro.

Cambiar los datos de expiración de la contraseña de un usuario

existente: chage
Aplicación: chage.
Necesita: chage.
Archivos de configuración: ninguno.
chage permite cambiar una, varias o todas las reglas utilizando las opciones de la línea de comando.
Opciones de la línea de comandos de chage

Opción	Propósito
-d [días desde la última]:	Esta opción se utiliza para contar el número de días transcurridos desde que se cambió la contraseña por última vez.
-E [fecha de expiración]:	Esta opción se utiliza para modificar la fecha en que la cuenta del usuario expirará y será bloqueada. Esta fecha se puede expresar tanto en días transcurridos desde el 1 de enero de 1970 como en formato de fecha estándar.
-I [días antes del bloqueo]:	Esta opción se utiliza para especificar cuántos días puede permanecer inactiva una cuenta con una contraseña expirada antes de ser bloqueada. Intente no ser demasiado estricto al respecto: a menudo, los usuarios no vuelven a sus cuentas durante una o varias semanas. Y dado que usted es el administrador del sistema, le importunarán para conseguir el desbloqueo de su cuenta.
-M [n° máximo de días]:	Esta opción se utiliza para modificar el número máximo de días durante los que es válida la contraseña del usuario. Por ejemplo, si desea obligar a los usuarios a cambiar de contraseña una vez cada 60 días, la opción será –M 60.
-m [n° mínimo de días]:	Esta opción se utiliza para modificar el número mínimo de días entre cambios de contraseña. Por ejemplo, si quisiéramos permitir a los usuarios cambiar de contraseña sólo una vez cada 30 días, la opción sería -m 30.
-W [días de advertencia]:	Esta opción se utiliza para modificar el número de días durante los que el sistema avisará al usuario de que hay que cambiar las contraseñas.

Tabla 3.16 líneas de comando de chage.

Mezclar y emparejar las bases de datos de /etc/passwd y /etc/shadow

Es posible que alguna vez tenga que migrar los datos de /etc/passwd al formato de shadow. Si eso ocurre, utilice pwconv, que no sólo permite la migración de datos desde una base de datos de /etc/passwd existente, sino que también permite integrar simultáneamente información con *shadowing* desde una base de datos de shadow existente.
pwconv tiene también varios mecanismos de seguridad automatizados. Uno de ellos es que siempre que se introduzcan entradas que no tengan ninguna contraseña asignada, pwconv no las migre a /etc/shadow. Más aún, pwconv utiliza la configuración predeterminada de expiración, aviso y bloqueo de cuentas que viene definida en /etc/login/defs. Esta configuración ofrece un buen punto de partida para todas las cuentas recién migradas.

Por otro lado, es posible que desee volver a convertir los datos de shadow al formato estándar de /etc/passwd, para lo que se utiliza pwunconv

Posibles ataques a un sistema con shadowing

Básicamente, la *suite* shadow simplemente esconde las contraseñas de los ojos curiosos. Así que, en lugar de que se pueda acceder a las contraseñas en /etc/passwd, se ocultan en /etc/shadow. A corto plazo, esto refuerza el sistema de seguridad. Sin embargo, los atacantes conocen perfectamente la *suite* shadow y en consecuencia han trasladado su interés por /etc/passwd a /etc/shadow. La única diferencia material desde el punto de vista del atacante es que /etc/shadow es más difícil de alcanzar

La *suite* shadow es bastante segura en sí misma, siempre que esté instalada la última versión. Pese a ello, desafortunadamente su seguridad depende generalmente mucho de la seguridad del sistema, ya que muchas otras aplicaciones tienen agujeros que permiten a los atacantes leer o incluso escribir en /etc/shadow. Hay que tener en cuenta que esto no es un fallo del autor de la *suite* shadow. Son sólo cosas que pasan.

La relación siguiente da una imagen ligeramente extensa y muestra lo eclécticos que pueden llegar a ser dichos ataques.

Varios ataques dirigidos al acceso a /etc/shadow

Exploit	Breve descripción y localización
deshadow.c:	Código fuente intruso para desproteger las entradas de /etc/shadow.
imapd hole:	Los fallos de imapd en Linux pueden revelar las contraseñas ocultas.
Telnet hole:	Se puede provocar un error utilizando telnet, que revelará las contraseñas ocultas. shadowyank: Aprovechando un agujero de FFP, shadowyank captura las contraseñas ocultas de los fallos de FTP
imapd crash:	imapd puede romperse y el volcado resultante revelará las contraseñas ocultas.

Tabla 3.17 ataques dirigidos a /etc/shadow.

La suite shadow es una importante innovación y una herramienta vital de cualquier arsenal de seguridad. Además de proteger las contraseñas de ojos no autorizados, la *suite* shadow ofrece controles extendidos sobre las cuentas y contraseñas de los usuarios así como una oportunidad de implementar al menos una política mínima de contraseñas con relativa facilidad.

Tras la instalación de la *suite* shadow

El shadowing de contraseñas es un excelente comienzo, pero no puede garantizar la seguridad de las contraseñas del sistema. Llegados a este punto, vamos a ampliar el alcance de la seguridad tradicional de las contraseñas (bloqueo de /etc/passwd) hasta otros temas como Elección humana de contraseñas y sus efectos sobre la seguridad del sistema, Comprobación proactiva de las contraseñas, Seguridad auxiliar de las contraseñas.

Elección humana de contraseñas y seguridad del sistema

El cifrado es un componente vital de la seguridad. Sin embargo, por muy potente
que sea nuestro cifrado, fallará si los usuarios eligen contraseñas débiles . Los usuarios son propensos a errores y olvidadizos. A menudo, crean contraseñas a partir de los siguientes datos (en parte para ahorrar tiempo y en parte para no complicarse la vida):
Fecha de nacimiento, Número del seguro social, Nombres de los hijos, Nombres de sus artistas favoritos, Palabras del diccionario, Secuencias numéricas (como 90125). Palabras escritas al revés.
Estas elecciones son terribles. Crack rompería cualquier contraseña de este estilo en segundos. De hecho, las buenas contraseñas son difíciles de conseguir, incluso si se tienen conocimientos de cifrado.
Este es el motivo por el que herramientas como Crack son valiosas. Mediante la comprobación regular de la fortaleza de las contraseñas de la red, es posible asegurarse que ningún intruso puede penetrar en ella aprovechando una mala elección de contraseña. Tal medida puede aumentar notablemente la seguridad del sistema. De hecho muchas personas emplean actualmente herramientas que comprueban las contraseñas de los usuarios al crearlas.

Comprobación proactiva de contraseñas

Se eliminan las contraseñas débiles antes de su envío a la base de datos de contraseñas. El proceso funciona de la siguiente forma: cuando un usuario crea una contraseña, ésta se compara en primer lugar con una lista de palabras y con una serie de reglas. Si la contraseña no cumple los requisitos de este proceso (por ejemplo, el comprobador proactivo de contraseñas encuentra una coincidencia o considera que el patrón es demasiado sencillo), se obliga al usuario a elegir otra
Los comprobadores proactivos de contraseñas que prevalecen son:

passwd+. anlpasswd. npasswd
passwd+.

Resaltan entre sus características:

- Grandes capacidades de registro, entre las que se incluyen el registro de todas las sesiones, de los errores, de los usuarios que han cambiado sus contraseñas, de las reglas que no cumplían la contraseña y del éxito o fracaso en el cambio de una contraseña dada.
- Especificación del número de caracteres significativos de la contraseña (es decir, cuántos se utilizarán en la comprobación).

Además, passwd+ permite establecer el mensaje de error que aparecerá cuando un usuario envíe una contraseña débil. Esta funcionalidad se debe utilizar para enseñar poco a poco a los usuarios los motivos por los que sus elecciones de contraseñas no siempre son acertadas.
Éstas son algunas reglas que proporciona passwd+:
El número de oficina, el teléfono de la oficina, el nombre de *host* y el de dominio están prohibidos.

- Las contraseñas deben tener al menos n caracteres de longitud.

- Las contraseñas deben mezclar mayúsculas y minúsculas.
- Las contraseñas que aparecen en el diccionario están prohibidas.
- El nombre y los apellidos (al derecho o al revés) están prohibidos.
- El nombre de conexión (al derecho o al revés) están prohibidos.

anlpasswd.

Este programa, escrito principalmente en Perl, utiliza el archivo de diccionarios que se elija y permite crear reglas personalizadas, las reglas predeterminadas estándar son las siguientes:
Números con espacios y espacios con números.
Mayúsculas y minúsculas con espacios.
Todo en mayúsculas o en minúsculas.
Todo en números.
La primera letra mayúscula y números.
Todas las combinaciones de las anteriores.
El código de Perl está excepcionalmente bien documentado y es fácilmente legible. A partir de ello, se puede obtener una aproximación del diseño de dichos programas, incluso aunque el conocimiento de Perl sea mínimo.

npasswd

Es más que un simple comprobador proactivo de contraseña. En 1993 se le había incorporado módulos de Crack. Actualmente, npasswd es un comprobador proactivo de contraseñas muy avanzadas. Es una exhaustiva solución de nivel comercial que puede reforzar considerablemente la seguridad de las contraseñas. La distribución incluso cuenta con un conjunto de herramientas de desarrollo para poder ampliar npasswd o incorporarlo a las aplicaciones.

Otros aspectos de la seguridad de contraseñas

En los ataques tradicionales a contraseña, los atacantes capturaban los archivos de contraseñas del sistema y ejecutaban utilidades de intrusos contra ellos. Su meta era crearse con el *root.* Como hemos visto, podemos eludir estos ataques con el *shadowing,* con utilidades proactivas de contraseñas y algo de sentido común.
Sin embargo, pese a que estos pasos reducen sustancialmente el riesgo, por sí mismos no garantizan la seguridad completa de las contraseñas, ya que en una red Linux media existen muchos otros mecanismos de contraseña, muchos de los cuales no utilizan /etc/passwd o /etc/shadow para su autenticación.

Proliferación de contraseñas y seguridad

Hasta el momento, nos hemos centrado principalmente en las contraseñas de inicio de sesión, que son verdaderamente importantes. Muchas aplicaciones

cliente-servidor utilizan autenticación estándar basada en /etc/ passwd o /etc/shadow (FFP, telnet y TFTP, por citar algunas). Sin embargo, dentro de un esquema mayor, éstas son sólo el principio.
Existe la posibilidad de tener al menos cinco contraseña en nuestro sistema:
Arrancar la computadora e introducir la contraseña de usuario.
Conectar con el proveedor de servicios de Internet e introducir la contraseña de conexión
Comprobar el correo con una contraseña de POP[82]3.
Conectar con algunas cuentas de correo de Altavista y Hotmail.
Hacer telnet al servidor de la empresa con otras contraseñas
Pero Linux es un sistema multiusuario y sabemos que es posible que tenga la intención de tener, al menos, unos pocos usuarios.
Supongamos también que utiliza Linux en entorno empresarial. En última instancia, se enfrentará a una situación similar a esta.

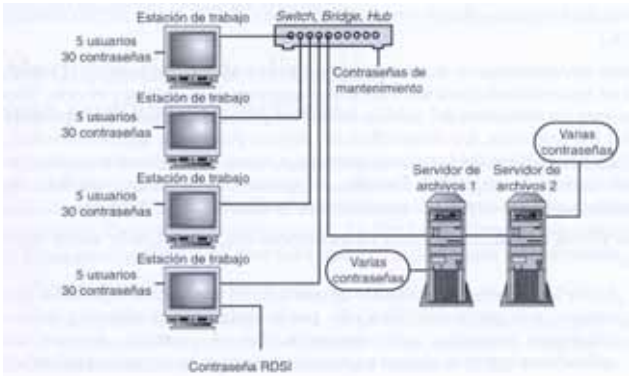


Fig. 3.14 Ejemplo de una pequeña red.

Su pequeña red podría contener unas 200 contraseñas. Hay sólo dos posibilidades para esto y ambas son igual de poco deseables: La mayoría de las contraseñas son iguales o La mayoría de las contraseñas son diferentes.
Cada posibilidad presenta sus propios riesgos en el primer caso, los usuarios crean contraseñas idénticas para varias aplicaciones y servidores, lo que es fatal, y si también cuentas externas con servicios como Hotmail. Suponga además que los usuarios son perezosos y que sus contraseñas de Hotmail son idénticas a las que utilizan en su sistema. La situación es muy comprometedora.
Piense en la situación contraria. Imagine que establece como política de empresa que todas las contraseñas han de ser únicas y los usuarios cumplen dicha política, incluso sobre sistemas que no admiten la comprobación proactiva de contraseñas. En ese caso, la calidad y fortaleza de estas contraseñas invariablemente empeorará. La apatía de los usuarios, unida a su ansiedad por no olvidar las distintas contraseñas, probablemente les hará crear contraseñas que sean rudimentarias o muy similares unas a otras.
Las aplicaciones de terceros utilizan las bases de datos de contraseñas ya establecidas (/etc/passwd o /etc/shadow) para llevar a cabo la autenticación. Aún es menos frecuente que implementen un almacenamiento de contraseñas completamente seguro. Estas circunstancias no harán más que empeorar la situación, porque el uso de la red se está volviendo cada vez más importante

[82] Protocolo de Oficina de Correos, Se traduce como un protocolo que está diseñado para permitir al usuario de manera personalizada leer el correo electrónico almacenado en un servidor.

para los negocios y el ocio. Esto incremento las demandas del público de nuevas y más interesantes herramientas de red. En respuesta, los desarrolladores siguen generando aplicaciones innovadoras y lanzándolas rápidamente al mercado, a menudo sin que estén sujetas a un control estricto de seguridad. Por ello, el mercado de consumo está lleno de aplicaciones que almacenan o transmiten contraseñas de forma inseguras.
Por ello, si va a distribuir Linux en un entorno empresarial, es recomendable que:
Limite los usuarios a un conjunto de aplicaciones establecidas y probadas que conozca a la perfección. Para ello, puede definir tareas caóticas y las herramientas necesarias para ejecutarlas. Por el contrario, descarte las aplicaciones que no se ajusten a estos criterios y prohíba al personal utilizarlas.
En cada aplicación aprobada, verifique el almacenamiento de contraseñas y los procedimientos de transmisión. Si es preciso, póngase en contacto con el distribuidor.
Para evaluar los procedimientos de transmisión de contraseñas, pruebe a espiar una sesión entre dos *hosts* utilizando la aplicación bajo sospecha. Los resultados nos revelarán si la contraseña se transmitió en texto sin formato, texto codificado con UUencode[83], texto XOR o texto cifrado
Elimine cualquier aplicación que emplee un mal almacenamiento de contraseñas y unos procedimientos de transmisión deficientes. Por ejemplo, si descubre que una aplicación
cliente-servidor almacena las contraseñas en el cliente, ésa es una señal de alarma.
Respecto al conjunto de aplicaciones aprobadas, esté al tanto de los avisos urgentes (y de las listas de seguridad) que emitan sus respectivos distribuidores, ya que a través de ellos puede saber en cuestión de horas cuándo se han descubierto nuevos agujeros
Pruebe la fortaleza del sistema de contraseñas una vez al mes, aunque utilice la comprobación proactiva de contraseñas.
Y sobre todo además de todo esto, aún dispone de una gran arma la educación del usuario. Asegúrese de que los usuarios comprenden la importancia de la seguridad de las contraseñas. En particular, intente subrayar la importancia de acatar la política de contraseñas, incluso cuando sea incómoda. Los usuarios nunca deben apuntar las contraseñas, dárselas a terceras personas sin autorización o compartirlas, ni siquiera con compañeros de confianzas.

Módulos de autenticación que pueden conectarse.

Un avance reciente en cuanto a autenticación son los módulos de autenticación que pueden conectarse (PAM[84], que permiten cambiar la forma en que las aplicaciones de Linux ejecutan la autenticación sin tener que rescribirlas y compilarlas. En las últimas distribuciones, los PAM se han

[83] (uucodificación/uucodificar) Se trata de un programa que convierte de manera reversible un archivo binario a uno de formato ASCII, se usa para poder mandar vía correo electrónico archivos binarios. Unix-To-Unix Encoding
[84] "Pluggable Authentication Modules" para Linux, es una suite de librerías compartidas que permiten al administrador local del sistema escoger cómo autentifican a los usuarios las aplicaciones.

integrado en el inicio de sesión y en otros procedimientos que requieren autenticación de contraseña.
Algunos módulos PAM típicos son:

- pam-cracklib: un comprobador proactivo de contraseñas que pueden conectarse. Este refuerza la comprobación de contraseñas de cualquier aplicación que conozca PAM.
- pam-deny: un módulo que puede conectarse y que avisará a una aplicación que conozca PAM de que la autenticación ha fallado. Fuerza la autenticación y deniega cualquier sesión en la que no se haya proporcionado autenticación o ésta haya resultado fallida.
- pam-pwdb: Un módulo de base de datos de contraseñas que pueden conectarse, que proporciona expiración de contraseñas, vencimiento, avisos, etc.
- pam-group: Un módulo que puede conectarse que asigna y rastrea la pertenencia a un grupo de los usuarios y de sus sesiones terminales.

Los PAM[85] proporcionan muchas opciones de gestión de autenticación, de cuentas, de sesiones y de contraseñas, y se han utilizado para desarrollar operaciones de autenticación como la **firma única[86].**

Otras soluciones para la seguridad de las contraseñas

Por último, se pueden utilizar otras soluciones exóticas para la seguridad de las contraseñas:
Controles de acceso biométrico: Estas herramientas autentican a los usuarios basándose en el olor corporal, la estructura facial, las huellas dactilares, los patrones del iris o la retina, el trazado de las venas o la voz. Los controles de acceso biométrico tienen un nivel excepcionalmente alto de precisión. Sin embargo, éstas son soluciones poco realistas debido a su alto costo
.
Los sistemas de contraseñas que se utilizan una sola vez generan contraseñas desechables. Estas contraseñas no se transmiten nunca por la red. En su lugar, el servidor reta al cliente con un valor numérico, que el cliente puede utilizar para generar un valor secreto adecuado para la transmisión de retorno. Los sistemas de contraseñas que se utilizan una sola vez están diseñados para evitar los ataques pasivos a la contraseña, en los que el atacante está monitorizando el sistema de redes con un *sniffer,* un analizador de protocolos.

[85] PAM también es compatible con las contraseñas MD5 y, por consiguiente, se pueden utilizar contraseñas mucho más largas.
[86] Esto es, cuando un usuario se auténtica una sola vez en una red de máquinas de confianza. Una vez que ha iniciado la sesión, el usuario puede desplazarse y su autenticación inicial le sigue.

3.7.2 Código dañino.

¿Qué es el código dañino?
Es código no autorizado dentro de un programa legal que realiza funciones que el usuario no conoce y no desea. Un programa legal que se ha modificado mediante la inserción en él de código no autorizado que ejecuta funciones desconocidas y no deseadas. Cualquier programa que parezca que realiza una función deseable y necesaria, pero que debido a que contiene código no autorizado ejecuta funciones que el usuario no conoce y no quiere. Código no autorizado diseñado para permanecer oculto y destruir datos.
Existen muchos tipos distintos de código dañino, pero los más frecuentes son:

- Troyanos.
- Virus.

Virus

Los virus informáticos se encuadran en dos categorías principalmente:
Programas ya diseñados para infectar, modificar o sobre escribir el sector de arranque o el registro de inicio maestro. Y los programas ya diseñados para adjuntar código dañino a los archivos del objetivo.
Los virus en archivos son más habituales y variados que los del sector de arranque y tradicionalmente han supuesto una mayor amenaza para las comunidades de redes, principalmente por la forma en que se propagan.
Durante los procesos de adjuntar, el código original del virus se añade a los archivos víctimas del mismo. Este procedimiento recibe el nombre de infección. Cuando se infecta un archivo, suele pasar de ser un archivo ordinario a uno portador de virus.
A partir de ese momento, el archivo infectado ya puede infectar a otros. Este proceso recibe el nombre de replicación. A través de la replicación, los virus pueden extenderse por toda una unidad de disco, con lo que se obtiene una infección sistemática, a menudo no aparece nada que avise de que se va a producir dicha infección antes de que ya sea un hecho consumado y, para entonces, ya puede ser muy tarde para salvar los datos dañados.
Sin embargo, es interesante saber que la mayoría de los virus realmente no destruyen datos, simplemente infectan discos o archivos. Pero aun cuando un virus no sea inherentemente destructivo, puede afectar al servicio. Por ejemplo, si se infectan, los controladores del sistema operativo pueden funcionar de manera irregular.
Sin embargo, no hay que olvidar que también existen virus destructivos. De hecho, uno de los primeros en circular públicamente se convirtió en un virus destructivo. Se llamaba Mertit y apareció en 1987. Este virus podía destruir la tabla de asignación de archivos (FAT) de los disquetes. Con el tiempo, Merrit pasó por varias etapas de evolución, la más peligrosa de las cuales se llamó Golden Gate. Golden Gate realmente formateaba el disco duro de la víctima.
UNIX es sencillamente un campo poco abonado para los virus, ya que Unix emplea un control de acceso basado en propietarios y grupos y restringe tajantemente los accesos de lectura, escritura y ejecución a los archivos. De ahí que sea difícil escribir virus que se extiendan en un entorno UNIX, el virus desea privilegios en los archivos y no puede obtenerlos. Por la contraparte, a

excepción de Windows NT, los entornos de Microsoft no imponen unos controles tan severos, lo que hace que sean los objetivos del virus.

Detectar código dañino.

El motivo es que el método más fiable de detección de código dañino es la reconciliación de objetos. Este método funciona de la siguiente forma: los objetos pueden ser archivos, directorios, dispositivos, etc. La reconciliación es el proceso de comparación de dichos objetos con la versión de una fecha anterior de ellos mismos.
Existen varios métodos para realizar la reconciliación de objetos, pero todos se basan en la detección de cambios en la información del estado de los archivos, un método muy primitivo es generar una lista de comprobación de todos los archivos y examinarla para ver si se ha producido algún cambio en: La última fecha en que se han modificado, Su fecha de creación, Su tamaño.
Este método no es suficiente, ya que dichos valores fecha y tamaño se pueden manipular con facilidad.
Otro método es utilizar sumas de comprobación básicas. Las sumas de comprobación son valores numéricos que se componen de las sumas de los bits de un archivo y suelen utilizarlas los programas que realizan transferencias de datos en red. Al transferir los datos del punto A al punto B, tanto el cliente como el servidor almacenan una suma de comprobación para cada bloque de datos. En el destino, esta suma de comprobación se compara con los datos recibidos. Si ambos valores coinciden, significará que los datos se han transferido correctamente y no son peligrosos. Sin embargo, si difieren, significará que se han dañado durante la transferencia y se produce un error.
Las sumas de comprobación de archivos estáticos se pueden generar utilizando varias utilidades, entre las que se incluye sum o en algunas plataformas, cksum
sum, como se define en man (archivo de ayuda de Linux). Calcula e imprime una suma de comprobación de 16 bits para el archivo con nombre y también imprime el número de bloques del archivo. A la hora de computar la suma de comprobación, se ignoran los caracteres NULL (con el valor ASCII cero). sum se suele utilizar para buscar puntos malos o para validar que un archivo se ha comunicado a través de una línea de transmisión.
Es muy fácil calcular las sumas de comprobación de los archivos estáticos. Éste es un ejemplo del listado de un directorio:

```
drwxrwxrwx  6 1046      sys        138 Jul   7 04:16  SSLftp-0.8
-rwxrwxrwx  1 mikal     user    368640 Jul   7 04:15  SSLftp-0_8_ tar
-rwxrwxrwx  1 mikal     user    189795 Jul   8 06:06  User-Manual.pdf
-rwxrwxrwx  1 mikal     user     21243 Jul   6 01:42  ftpsec.txt
-rwxrwxrwx  1 root      sys        556 Jul   7 04:18  junk.txt
-rwxrwxrwx  1 mikal     user      4005 Jul   7 04:30  morejunk.txt
-rwxrwxrwx  1 root      sys     39   Jul    8 21:21 test-checksum.txt
-r,wxrwxrwx 1 mikal     user      6191  Jul   8 06:45  tripwire.txt
-rwxrwxrwx  1 mikal     user     18952  Jul   8 06:46  twpol.txt
```

Para obtener sumas de comprobación básicas de 16 bits en estos archivos, se puede introducir el siguiente comando:

```
# sum *
```

Ésta es la salida con las sumas de comprobación:

```
Read error on SSIftp-0.8: Is a directory
0 0 SSLftp-0.8
9784 720 SSIftp-0_8_tar
33473  371 User-Manual.pdf
28778  42 ftpsec.txt
```

Si se desea una prueba de integridad rápida y poco fiable, se puede generar una instantánea del sistema operativo, con:

```
sum 'find /. -print' > os_datebase.txt
```

Este comando generaría una suma de comprobación de 16 bits para todos los archivos de la unidad de disco duro y pondría la salida en el archivo os_database.txt
Este método es, sin duda, preferible a confiar en la hora, en la fecha o en la fecha en que se modificó por última vez. Sin embargo, las sumas de comprobación de 16 bits no son suficientes. Por consiguiente, el método imperante es el uso de algo como MD5. MD5 pertenece a una familia de funciones hash unilaterales llamadas algoritmos de síntesis de mensajes. El algoritmo MD5 toma como entrada un mensaje de longitud arbitraria y crea como salida una "huella digital" o una "síntesis de mensaje" de 128 bits de la entrada. Se cree que es computacionalmente imposible crear dos mensajes que tengan la misma síntesis o crear cualquier mensaje que tenga una síntesis de mensaje dada con un objetivo previamente especificado.
Los algoritmos de síntesis de mensajes ofrecen una gran garantía y son muy útiles para probar la integridad de los archivos. La clave consiste en utilizar herramientas que puedan tomar una instantánea del sistema operativo inicial y generar valores de MD5 (o comparables) para poder realizar comparaciones posteriores. Para estas funciones, la mejor herramienta es Tripwire.

Tripwire

Es una flexible y sencilla herramienta que se utiliza para comprobar la integridad de los archivos y que emplea varios algoritmos:

- CRC32: CRC32 es una versión de 32 bits de CRC. El CRC general se utiliza para comprobar la integridad de los archivos que se van a transmitir digitalmente.
- MD2: MD2 se encuentra en la familia MD5 de algoritmos de síntesis de mensajes. Es muy potente. Ya que en sus especificaciones se indicaba que la dificultad de hacer frente a dos mensajes que tienen la misma síntesis de mensajes se encuentra en torno a las 2^64 operaciones y que la dificultad de hacer frente a cualquier mensaje que tenga una síntesis de mensaje dada se encuentra en torno a las 2^128 operaciones.
- MD4: Abreviatura de Message Digest 4. Aunque se considera un sistema inseguro, es importante porque ha servido de base para la creación de otras funciones hash. Un sistema de ataque desarrollado por Hans Dobbertin posibilita el crear mensajes aleatorios con los

mismos valores de hash (colisiones), por lo que ya no se usa. De hecho, existe un algoritmo que encuentra una colisión en segundos.

- MD5: MD5 es un algoritmo más lento, pero más seguro que MD4 y es, por tanto, una mejora. Para conocer el diseño y los objetivos de MD5. abreviatura de Message Digest 5, se creó para dar seguridad a MD4, y que ha sido ampliamente usado en diversos campos, como autenticador de mensajes en el protocolo SSL y como firmador de mensajes en el programa de correo PGP. Si embargo, fué atacado en 1996 por el mismo investigador que lo hizo con MD4, se consiguió crear colisiones en el sistema MD5, aunque por medio de ataques parciales. Pero lo peor es que también consiguió realizar ataques que comprometían la no-colisión, por lo que se podían obtener mensajes con igual hash que otro determinado. A pesar de todo esto, MD5 se sigue usando bastante en la actualidad.
- SHA Secure Hash Algorithm (el algoritmo NIST de *hash* seguro): SHA es excepcionalmente eficaz y se ha utilizado en entornos de defensa.
- Snefru (función de *hash* segura de Xerox): Snefru puede generar síntesis de mensaje de 128 ó 256 bits. *Snefru* lo desarrolló Xerox y la versión actual es la 2.4. *Snefru*.

De forma predeterminada, Tripwire usa tanto MD5 como la función de *hash* segura de Xerox para generar huellas digitales de los archivos sin embargo, puede aplicar cualquiera de las funciones hash anteriores a cualquiera o a una parte de los archivos. Por consiguiente, *Tripwire* ofrece una alta garantía de integridad del sistema de archivos como punto de referencia inicial. A continuación se enumeran algunas características:

- Puede llevar a cabo su función sobre las conexiones de red. Por tanto, puede generar una base de datos de huellas digitales de toda la red en el momento de la instalación.
- Se escribió en C con la intención de que fuera transportable. Se puede compilar en la mayoría de los entornos sin ninguna modificación.
- Incluye un lenguaje de procesamiento de macros, por lo que se pueden automatizar determinadas tareas.

Es una herramienta magnífica, pero solamente cuando se usa junto con otras medidas de seguridad. Por ejemplo, no se saca ningún partido de Tripwire si no se protege la base de datos inicial de instantáneas y huellas digitales. Desde el principio, los creadores de esta herramienta han dejado claro que la base de datos utilizada por el comprobador de la integridad debe protegerse de modificaciones no autorizadas; cualquier intruso que pueda cambiar la base de datos puede perturbar todo el esquema de comprobación de la seguridad.
En un principio, Tripwire se diseñó para UNIX, no para Linux. Actualmente, la única distribución de Tripwire prefabricada se puede ejecutar en Red Hat 5.x. También puede hacer funcionar su propio Tripwire en otros sistemas de Linux. Por ejemplo, se sabe que Tripwire 1.3 se compila perfectamente en Debian, Openlinux y en otras distribuciones.

Instalar Tripwire

Tras descargar el paquete de Tripwire, cree un directorio y cópielo en él. Por ejemplo:

```
[root@linux9 /]#       mkdir tripwire
[root@linux9 /]#       cp Tripwire_2_0_RedHat_Linux_tar tripwire/
```

Seguidamente, vaya al nuevo directorio y descomprima el paquete de Tripwire ya que está comprimido con los métodos zip y tar

```
[root@linuxg /tripwire]# gunzip Tripwire_2_0_RedHat_Linux-tar.gz
[root@linuxg ltripwire]# tar -xvf Tripwire_2_0_RedHat_Linux_tar
```

El archivo debería descomprimirse en los siguientes archivos y directorios:

```
-r--r--r--     1root     root     2732 Feb  26 02:00  README
-r--r--r--     1root     root    10955 Feb  26 02:00  Release_Notes
-r--r--r--     1root     root   189795 Feb  26 02:00  User_Manual.pdf
dr-xr-xr-x     2root     root     1024 Feb  26 02:00  bin/
-rw-r -----    1roct     root     1318 Feb  26 02:00  install.cfg
-r-xr-x ---    1root     root    22072 Feb  26 02:00  install.sh*
-r--r--r--     1root     root     7238 Feb  26 02:00  license.txt
dr-xr-xr-x     2root     root     1024 Feb  26 02:00  pkg/
```

install.sh es *el script* de instalación e install.cfg es el archivo de configuración de la instalación. Antes de llevar a cabo la instalación, lea el archivo install.cfg, ya que define los directorios de destino de la instalación.
De forma predeterminada, esta configuración coloca todo en /usr/TTS. Si no tenemos en cuenta el caso improbable de que ya tenga dicho árbol de directorios, es probable que no tenga que cambiar esta configuración. Sin embargo, si prevé que Tripwire va a tener que sobrescribir archivos existentes, tiene que modificar la línea 21:

```
# If CLOBBER is true, then existing files are overwritten.
# lf CLOBBER is false, existing files are not overwritten.
CLOBBER=false
```

En caso contrario, si no tiene que realizar ningún cambio en install.cfg, comience el proceso de instalación, tal como sigue:

```
[root@linux9 /tripwire]# ./install.sh
```

Tripwire mostrará un resumen de las opciones elegidas y le pedirá que las confirme.
Si estos valores son los correctos, elija y. A continuación, Tripwíre le solicitará una frase de paso clave a los archivos. Antes de escribirla, piénsela detenidamente.

Generar frases de paso

Varía levemente con respecto a la generación de contraseñas, ya que las opciones son mayores. Una frase de paso puede ser cualquier cosa y, aunque

debe tener un mínimo de ocho caracteres, no hay límite máximo, las frases de paso pueden tener espacios en blanco.
Sin embargo, de igual forma que al generar una contraseña, hay que tener en cuenta determinadas convenciones para asegurarse de que no se averigua con facilidad. De forma errónea, muchos usuarios suponen que dado que al ser más largas que las contraseñas estándar, las frases de paso son automáticamente más difíciles de averiguar, lo que no es cierto. Si la frase se puede predecir con facilidad, es tan fácil de romper como una contraseña de ocho caracteres, así que elija con cuidado la frase que desea utilizar a modo de contraseñas.
Tripwire solicitará varias frases de paso, comenzando por la frase de paso clave a los archivos.
Tras escribirla, Tripwire le pedirá que la confirme.

```
Verify the site keyfile passphrase:
```

Y para finalizar, Tripwire generará una clave:

```
Generating key (this may take several minutes)
```

A continuación, Trypwire le pedirá las frases de paso clave locales y del sitio
Y de nuevo, le pedirá que las verifique:

```
Verify the local keyfile passphrase:
```

y generará las frases:

```
Generating key (this may take several minutes)
```

Finalmente, Tripwire le solicitará la frase de paso del sitio:

```
Generating Tripwire configuration file
Creating signed configuration file
Please enter your site passphrase:
```

Cuando haya acabado, Tripwire le notificará que la instalación se ha realizado correctamente:

```
The installation succeeded.
Please refer to /usr/TSS/Release_Notes
for release information and to the printed user documentation for
further instructions on using Tripwire 2.0 for Unix.
```

Antes de ejecutar realmente Tripwire, hay que personalizar dos archivos:
El archivo de configuración de Tripwire.
El archivo de políticas de Tripwire.

El archivo de configuración de Tripwire

El archivo de configuración almacena información específica del sistema principalmente sobre el lugar en que están instalados las utilidades y los

archivos de configuración de Tripwire. De forma predeterminada (twcfg.txt) se encuentra en /usr/TSS/bin y es similar al siguiente:

```
[root@linux9 bin]# more twcfg.txt
ROOT            =/usr/TSS
POLFILE         =/usr/TSS/policy/tw.pol
DBFILE          =/usr/TSS/db/$(HOSTNAME).db
REPORTFILE      =/usr/TSS/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE     =/usr/TSS/key/site.key
LOCALKEYFILE    =/usr/TSS/key/$(HOSTNAME)-local.key
MAILPROGRAM     =/usr/lib/sendmail -oi –t
EDITOR          =/bin/vi
LATEPROMPTING   =false
LOOSEDIRECTORYCHECKING =false
```

Servicio	Explicación
DBFILE	La variable DBFILE señala a la ubicación del archivo de su base de datos es el que almacena la "instantánea" del sistema operativo.
EDITOR:	La variable EDITOR almacena la ubicación de su editor favorito. Para utilizar Tripwire en el modo de edición interactivo hay que definir esta variable. Además, una vez que se especifica este valor, no es posible cambiarlo manualmente manipulando las variables de entorno de la *shell.*
LATEPROMPTING:	La variable LATEPROMPTI NG se utiliza para especificar si Tripwire debe esperar hasta el último momento antes de solicitar una frase de paso, ésta es una medida de seguridad para los extremadamente obsesivos a los que preocupa que mientras sus frases de paso estén en memoria otros las puedan capturar.
LOCALKEYFILE :	La variable LOCALKEYFILE apunta a la ubicación del archivo clave local.
LOOSEDIRECTORYCHECKING:	La variable LOOSEDIRECTORYCHECKING afecta a la forma en que Tripwire informa de los cambios que se producen en los directorios. Si LOOSEDIRECTORYCHECKING no está activada, Tripwire no sólo informará de que se ha eliminado o modificado cualquier archivo, sino también de la repercusión que ha tenido dicho cambio en el directorio en el que se encuentra o se encontraba el archivo. Sin embargo, si se activa, Tripwire informa solamente de la modificación de los archivos y no de la modificación de los directorios.
MAILPROGRAM:	La variable MAILPROGRAM almacena la ubicación del programa de correo especificado y de todas las opciones de la línea de comandos que se van a pasar a él.
POLFILE:	La variable POLFILE apunta a la ubicación del archivo de políticas normalmente /usr/TSS/policy/tw.pol).
REPORTFILE:	La variable REPORTFILE apunta al lugar en que Tripwire va a almacenar sus informes.

SITEKEYFILE:	La variable SITEKEYFILE apunta a la ubicación de la clave del sitio.

Tabla 3.18 Variables del archivo de configuración de Tripwire

Antes de ejecutar Tripwire por primera vez, estos valores pueden mortificarse a voluntad.

El archivo de políticas de Tripwire

El archivo de políticas de forma predeterminada, twpol.txt almacena la especificación de qué objetos (archivos, directorios, etc.) debe monitorizar Tripwire y de sus ubicaciones.
Tripwire incluye un archivo de ejemplo llamado /usr/TSS/policy/twpol.txt que está optimizado para Red Hat Linux 5.x. Antes de ejecutar Tripwire por primera vez, es aconsejable que lo busque.
Sin embargo, si no tiene que realizar ninguna modificación en el archivo de políticas, ya puede configurar y ejecutar Tripwire.
Para configurar y ejecutar Tripwire, hay que cambiar el directorio de archivos binarios:

```
[root@linux9 /root]# cd /usr/TSS/bin
```

Una vez en él, escriba el siguiente comando:

```
./twadmin --create-cfgfile --site-keyfile ../key/site.key twcfg.txt
twadmin le pedirá la frase de paso:
Please enter your site passphrase:
```

Tras verificarla, twadmin dará formato al archivo de configuración y se cerrará:

```
Writing config file: /usr/TSS/bin/tw.cfg
Wrote configuration file: /usr/TSS/bin/tw.cfg
```

Seguidamente, hay que actualizar el archivo de políticas, como se muestra a continuación:

```
./twadmin --create-polfile ../policy/twpol.txt
twadmin le volverá a pedir la frase de paso:
Please enter your site passphrase:
```

Tras verificarla, twadmin escribirá el nuevo archivo de políticas y se cerrará:

```
Wrote policy file: /usr/TSS/policy/tw.pol
```

Ya estamos listos Para generar la base de datos de Tripwire, para lo que debe escribir el siguiente comando:

```
[root@linuxg bin]# ./tripwire –init
```

Tripwire solicitará la frase de paso: Please enter your site passphrase:

Lo que ocurra a continuación depende de la configuración del sistema. Si no se han eliminado las posibles rutas erróneas del archivo de políticas, es posible que aparezcan varios errores como el siguiente:

```
# Error 101, Unable to get object type: file:/usr/lib/tcIX8.0.3/help
No such file or directory
#Error 101: Unable to get object type: file:/usr/lib/tkX8.0.3/help
No such file or directory
```

Corríjalos posteriormente cambiando las reglas del archivo de política.
Tras la primera prueba, es aconsejable corregir las reglas de las políticas para evitar estos errores. Ello se debe a que estos errores volverán a aparecer cada vez que utilice Tripwire para verificar la integridad de los archivos o cualquier otra función en la que haya que acceder a la base de datos.
Para finalizar, Tripwire creará la base de datos y el informe:

```
Wrote database file: /usr/TSS/db/linuxg.samshacker.net.db
The database was successfully generated.
Exiting.
```

Verificar la integridad de los archivos con Tripwire
Tras la primera vez que se ejecute Tripwire, éste almacenará la instantánea de todo el sistema operativo. A partir de ese momento, para probar la integridad de los archivos del sistema, hay que introducir el siguiente comando:

```
[root@linux9 bin]# ./tripwire --check
```

Tripwire rastreará todos los objetos del sistema lo que puede tardar un tiempo e informará de los resultados:

```
Total objects scanned: 18303
Total violations found: 1
-------------------------------------------------------
Severity Level: 100          Rule Name: Root config files (/root)
Total objects scanned: 22
 -------------------------------------------------------
Modified:
               Mode                        UID         Size        Access
Time
---------                 --------- -------------   --------    -----
------- --------
/root/tripwire2.txt  -rw-r--r-- root (0)      3262 XXXXXXXXXXXXXXXXX
-------------------------------------------------------
Object Detail:
Severity Level: 100                   Rule Name: Root config files
(/root)
Total objects scanned: 22
Modified Objects:
Rule Name: Root config files (/root)
Total number of modified objects: 1
Modified object name: /root/tripwire2.txt
Property:          Expected:            Observed:
     ............      ............          ............
Device Number      770                  770
Inode Number       39013                39013
```

```
Mode                -rw-r-.r--           -rw-r--r--
Num Links           1                    1
UID                 root (0)             root (0)
GID                 root (0)             root (0)
*Size                3000                 3262
*Modify Time Thu    Jul 8 17:21:19 1999 Thu Jul 8 17:26:30 1999
*Blocks             6                    8
Object Type      Regular File           Regular File
*MD5 C1l9Xm9xh64Qmh+tlMkYn2           DHJt4XbO7rvVNJtQyr5G9Q
*SHA GWDiSuyABuaQl6F+IvjkqlnwHjF    FlCvT/HMyTZMkFcUklkVZ5PCKjf
 ---------------------------------------------------------
 ***End of report***
Integrity check complete.
Exiting ...
```

Aquí se puede ver que Tripwire ha detectado que un archivo ha cambiado:
Modified object name: /root/tripwire2.txt
De hecho, Tripwire ha determinado que el tamaño del archivo, la fecha de modificación y las huellas de MD5 y de SHA han cambiado:

Otro software para comprobar la integridad de los archivos

Además de Tripwire, existen otros comprobadores de la integridad de los archivos y algunos de ellos incluyen el código fuente. Se sabe que todos ellos compilan en varias clases de UNIX, pero ninguno es específico de Linux. Hacemos mención a ellos por si desea probarlos pero recomendarnos Tripwire.

TAMU

El conjunto de programas TAMU (siglas de Texas A&M University)
La distribución de TAMU incluye un paquete de scripts tiger, que forma la base de la autenticación de las huellas digitales de la distribución.
Los scripts utilizan programas de suma de comprobación criptográfica de Xerox que verifican tanto los archivos binarios modificados del sistema (posibles trap doors o troyanos) como la presencia de parches necesarios relacionados con la seguridad.
La distribución de TAMU es exhaustiva y puede utilizarse para resolver varios problemas de seguridad, además de para buscar troyanos. Incluye un monitor de la red y un filtro de paquetes.

ATP (Anti-Tampering Program)

Hasta cierto punto, ATP funciona igual que Tripwire. en ATP:
ATP toma una instantánea del sistema, siempre que se encuentre en una configuración confiada, y realiza una serie de comprobaciones para monitorizar los cambios que se puedan haber realizado en los archivos.
A continuación, ATP establece una base de datos de valores para cada archivo. Uno de estos valores (la firma) consta de dos sumas de comprobación. La primera es una suma de comprobación CRC32, mientras que la segunda es una suma de comprobación MD5. A causa de su velocidad, la suma de comprobación CRC32 se utiliza en las comprobaciones que se realizan regularmente (quizás diariamente).MD5, que es más exhaustivos (y

por tanto con más recursos y más tiempo), está pensado para comprobaciones periódicas programadas (quizás una a la semana).
La base de datos se cifra con DES. Por consiguiente, ATP proporciona un método flexible (pero bastante seguro) para monitorizar la red e identificar posibles troyanos.

Hobgoblín

Ofrece una mezcla interesante de comprobación de la integridad de los archivos y del sistema.
Hobgoblin es más rápido y configurable que COPS y generalmente recopila información con más detalle. Aunque lo que hace que Hobgoblin sea muy interesante es que no sólo es un lenguaje, sino que también es un intérprete. Los programadores le han dado sus propios descriptores únicos y sus propias convenciones estructurales. Sin embargo el intérprete de Hobgoblin reserva metacaracteres conocidos y muy utilizados que tienen un significado especial.

sxid

Creado por Ben Collins de Debian, hace seguimientos de archivos suid y sgid mediante sumas de comprobación MD5 y puede detectar si se ha instalado un kit de *root*. Lo diseñó para que funcionara como una tarea cronológica y automáticamente hará un seguimiento, detectará y avisará de todos los cambios sospechosos.

Trojan.pl

Creado por Bruce Barnett, comprueba los permisos de los archivos, directorios y usuarios en una ruta determinada de aquellas configuraciones que podrían invitar a ciertos usuarios a instalar caballos de troya. Curiosamente, realmente averigua la probabilidad de que un ataque pueda instalar un troyano.

3.8 Sniffers

Las estaciones de trabajo solo escuchan y responden solamente a los paquetes que van dirigidos a ellas. Sin embargo, es posible modelar el software que lanza la interfaz de red de una estación de trabajo en algo llamado modo promiscuo. Teniendo esto en cuenta, la estación de trabajo puede monitorizar y capturar todo el tráfico de red y los paquetes que pasen por ella, independientemente del destino que tengan.
Los sniffers se suelen escribir en C, o se puede utilizar Perl, y raras excepciones, abren su fuente con directivas include:

```
#include   <linux/if.h>
#include   <Iinux/if   ether.h>
#include   <linux/ip.h>
#include   <linux/socket.h>
#include   <linux/tcp.h>
#include   <netinet/in.h>
#include   <signal.h>
#include   <stdio.h>
```

```
#include   <sys/socket.h>
#include   <sys/time.h>
#include   <sys/types.h>
```

El motor LXR es una versión en hipertexto del código fuente de Linux que se explora perfectamente. El LXR es tan complejo que todos los archivos de cabecera, todas las llamadas al sistema, la mayoría de las funciones, etc. tienen referencias cruzadas. Utilizándolo, se puede acceder a cualquier punto del código de Linux desde cualquier otro punto. De esta forma, independientemente de cuál sea su situación personal. Algunos de los archivos de cabecera antes mencionados y sus funciones:

- Linux/if.h: contiene definiciones para controlar la interfaz de Ethernet.
- Linux/if-etherh: contiene definiciones para la interfaz IEEE 802.3 de Ethernet y para varios protocolos de Ethernet como AppleTalk, bucle Ethernet y protocolo Internet.
- Linux/in.h: contiene definiciones de las estructuras de las direcciones de Internet.
- Linux/ip.h: una implementación de IP para Linux.
- stdio. h: gestiona las entradas, las salidas y las salidas de errores estándar.
- sys/socket.h: gestiona las operaciones de los *sockets,* incluyendo listen, bind, connect, accept, send. También contiene definiciones de varios tipos de *sockets* (incluyendo AppleTalk, IPX.), cuyos representantes más importantes son AF_UNIX o los *sockets* de UNIX.
- tcp.h: contiene definiciones para varios estados de las conexiones TCP, como TCP_ESTABLISHED (conexión establecida), TCP-LISTEN (escuchando), TCP ClOSE (cerrando),

La mayoría de los *sniffers* se han diseñado con estos archivos de cabecera. Cada uno de ellos gestiona un aspecto distinto de la escucha, grabación y generación de informes sobre el tráfico de TCP/IP Sin embargo, los piratas ponen la interfaz en modo promiscuo utilizando una marca de if.h (actualmente, en la línea 34) muy similar a ésta:

```
#define IFF-PROMISC     OX100   /* recive all packets */
```

Linsniffer, su creador Mike edulla, abre la interfaz en modo promiscuo de la siguiente manera:

```
int openintf(char *d)
{
int fd;
struct ifreq ifr;
int s;
fd=socket(AF_INET, SOCK_PACKET, htons(OX800));

if (f d < 0)
{
perror("cant get SOCK_PACKET socket");
```

```
exit(0);
}
strcpy(ifr.ifr-name, d);
s=ioctl(fd, SIOCGIFFLAGS, &ifr),
if(s < 0)
{
close (f d) ;
perror("cant get flags”);
 exit(0);
}
ifr.ifr_flags ¦= IFF_PROMISC;
s=ioctl(fd, SIOCSIFFLAGS, &ifr);
if(s < 0) perror("cant set promiscuous mode");
 return fd;
}
```

Una vez que la interfaz se encuentra en modo promiscuo y, por tanto, escucha todos los paquetes de la red, lo que queda es escuchar el tráfico TCP/IP y darle un formato que pueda leerse en la salida estándar o escribirlo en un archivo.
Los distintos *sniffers* realizan tareas diferentes, que oscilan entre las sencillas (capturar nombres de usuarios y contraseñas) y las extremas (grabar todo el tráfico de la interfaz de red:

- linsniffer.
- linuxsniffer.
- hunt.
- sniffit.

El propósito principal de linsniffer es capturar nombres de usuarios y contraseñas y ésta es una función en la que sobresale.

3.8.1 Sniffit

Usaremos la versión beta de sniffit.0.3.7.beta.tar.gz, para instalar sigamos los siguientes pasos:

```
tar xzvf  sniffit.0.3.7.beta.tar.gz
sniffit se descomprime en sniffit.0.3.7.beta/ entrar en este y
ejecutar
./configure
make
strip sniffit
```

El escucha está instalado

Se puede emplear con comandos en línea o con un archivo de configuración. Para los comandos en línea vea lo que sucede en el siguiente ejemplo: corra snffit en el hosts 10.0.0.200, como se indica a continuación.

```
./sniffit -p 23 -t 10.0.0.@
```

Observe que la tarjeta red esta en modo promiscuo

-p puerto 23 telnet
-t paquetes que van a cualquier hosts de la red 1 0.0.0.0
-s paquetes que vienen
Ver más opciones en archivo sniffit.0.3.7.beta/README.FIRST

Desde el hosts 1 0.0.0. 1 se inicia sesión telnet en 10.0.0.200
y se captura un archivo 10.0.0. 1. 1031-10.0.0.200.23 con algo como:

```
ÿû’ÿü ÿü#’yü 'ÿú 'ANSI ÿÿýÿûypyü!yüý rlopez
lopez36
ls –alF
who
ps –a
su
root
logica01
vi texto
kill%
jobs
OA
```

Donde se ve que el usuario rlopez, con pasword lopez36, ejecuta varios comandos, entre ellos hace ingenuamente un su a root y también se captura el password que es logica01.
sniffit con la opción -c envía la información capturada a un archivo. La sintaxis para configurar el archivo emplea 5 campos:

1	2	3	4	5
select	from	host	lista	# de puerto
deselect	to	mhost		
both	port			

campo 1: select: captura paquetes de host, mhost, en campo3, o no deselect
campo 2: from, to, both: paquetes que vengan de, hacia o ambos, de hosts o mhost
campo 3: host, mhost: un hosts, múltiples hosts o puerto. mhosts admite comodines
campo 4: lista: un host 10.0.0.1 múltiples host 192.168.0. o número de puerto
campo 5: número de puerto.

ejemplos:

select both host 192.168.0.20
deselect both port 80
select both mhost 192.168.0 23

Todos los paquetes hacia o de host 192.168.0.20, excepto el puerto www, todos los paquetes de o hacia la subred 192.168.0, a telnet, puerto 23.
sniffit interpreta el archivo en forma secuencial.

Veamos otro ejemplo:

select from host 192.168.0.25
select to host 192.168.0.21 80
deselect both port 21

Todos los paquetes que provengan del host 192.168.0.25, todos los paquetes que vayan hacia el host 192.168.0.1 por el puerto 80, ningún paquete de o hacia cualquier hosts de la subred por el puerto 21.

Escribir el archivo *escucha* con el siguiente ejemplo:
select both mhost 192.168.0 port 23

Los paquetes de o hacia el rango de hosts de 192.168.0.1 a 192.168.0.255 que vayan hacia el puerto telnet 23. Ahora digamos sniffit que use el archivo escucha.
. /sniffit -c ./escucha -L telnet

sniffit se inicia y la tarjeta de red entra en modo promiscuo

sniffit Loggíng started.
Supported Network device found. (eth0)
eth0: Promiscuous mode enabled.
Sniffit.0.3.7 Beta is up and running ... (Config File Used)

sniffit termina cuando tecleamos ctrl C
Gracefull shutdown ...
eth0: Promiscuous mode enabled.
Sniffit Logging session ended.

se crea el archivo sniffit. log

3.9 Protección de datos en transito

3.9.1 Secure Shell (ssh)

El protocolo SSH Secure Shell es un conjunto de herramientas de conectividad de red usadas para encriptar conexiones a través de Internet. SSH encripta todo el tráfico, incluyendo logins y passwords para eliminar el rastreo de red (sniffing) y otros ataques de red. SSH surgió con la intención de añadir seguridad a las comunicaciones entre las máquinas remotas a través de los comandos rcp, rsh, rlogin y telnet. Si embargo ha llegado a ser un programa de uso más que recomendable para administradores de sistemas y personas que utilizan sus máquinas de forma remota, en un entorno de red inseguro, como lo puede ser internet. Esto llevó al equipo de SSH a implementar funcionalidades a la aplicación, como la posibilidad de usar sesiones X (X Windows permite el uso de clientes gráficos remotos). SSH también incluye el servicio de ftp seguro, conexiones de SAMBA seguras, así como en servidores web y los servidores POP y SMTP.

Características de la clave.

Una clave es esencialmente es un número muy grande que tiene propiedades matemáticas especiales. Que alguien pueda romper el esquema de encriptación depende de su habilidad para encontrar cuál es esa clave. Así, cuando mayor sea la clave, más difícil será descubrirla. La encriptación de grado bajo,
(La clase que permite exportar el gobierno de Estados Unidos) tiene 56 bits. Esto significa que hay 2^{56} claves posibles. Esto es 65,536 trillones.

Criptografía de clave pública.

La shell segura SSH se basa en una tecnología llamada criptografía de clave pública. Funciona de un modo similar a una caja fuerte de un banco: necesita 2 llaves para abrir la caja. En el caso de la criptografía de clave pública, necesita dos claves matemáticas: una pública y otra privada. Estas 2 claves se combinan para encriptar los datos, cada combinación de claves públicas y privadas es única

Secure Shell admite varios algoritmos, entre los que se incluyen:

- BlowFish .
- Triple DES.
- IDEA.
- RSA.

Los autores incorporaron esta compatibilidad para crear un producto más flexible y ampliable. La arquitectura de ssh es tal que al protocolo básico le da igual el algoritmo que se utilicé. Por tanto, si posteriormente se descubre que uno o varios de los algoritmos compatibles tienen defectos en sus fundamentos, es posible cambiar rápidamente de uno a otro sin modificar el protocolo clave y las funciones de ssh.
ssh también tiene otras ventajas con respecto a sus competidores. La ventaja más significativa es que ssh no modifica mucho las rutinas. En todos los aspectos, iniciar una sesión de ssh es tan sencillo como iniciar una sesión de telnet. Tanto la autenticación como el posterior cifrado de sesiones son transparentes.

El programa OpenSSH:

OpenSSH que es libre y viene con la versión de Linux Red Hat 7.x y mayores soporta los protocolos SSHI y SSH2. Para verificar si esta instalado OpenSHH ejecutar el comando rpm.

```
[root@redhat80 root]# rpm -qa ¦ grep ssh
openssh-server-3.4pl –2
openssh-clients-3.4pl -2
openssh-askpass-gnome-3.4pl –2
kdessh-3.0.3-3
openssh-3.4pl-2
openssh-askpass-3.4pl –2
```

```
Los programas son para:
Cliente: openssh-clients-3.4pl-2-
Server: openssh-server-3.4pl-2
```

Común a cliente y server: openssh-3.4pl-2
Los otros son para cliente Gnome y KDE
Nosotros requerimos el común y el cliente o server, si no estuviesen, proceder a instalarlos de los RPMS de los cdroms de Linux Red HAT 9.0, de acuerdo a cliente o server.

Archivos de claves, hosts y configuración:

Los archivos de configuración, hosts y claves están en /etc /ssh, ejecute el comando Is

```
[root@redhat80 root]# Is -alF /etc/ssh
-rw------     1 root  root 88039 ago 13 2002        moduli
-rw-r--r--    1 root  root   1137  ago 13 2002      ssh_Config
-rw-----       1 root  root   2449  ago 13 2002      sshd_config
-rw-----       1 root  root   668   feb  13 17:03    ssh_host_dsa_key
-rw-r—r--      1  root    root      590      feb  13  17:03         ssh_
host_dsa_key.pub
-rw-------    1 root  root   515   feb 13 17:03     ssh- host_key
-rw-r—r--   1 root  root   319   feb 13 17:03     ssh_host_keypub
-rw-------    1 root  root   887   feb 13 17:03     ssh_host_rsa_key
-rw-r—r--         1   root      root         210         feb   13   17:03
ssh_host_rsa_key.pub
```

Moduli

Algoritmo Diffie-Hellman usado para intercambio de claves. Cuando se intercambian las claves al inicio de una sesión ssh, se crea un valor secreto y compartido que no puede ser determinado por ninguna de las partes individualmente. Este valor se usa para proveer autenticación de hosts.
ssh_config
El archivo de configuración del cliente ssh.
sshd_config
El archivo de configuración del demonio sshd en el server.
ssh_host_dsa-key
The DSA private key used by the sshd daemon.
ssh_host_dsa_key.pub
The DSA public key used by the sshd daemon.
ssh_host_key
The RSA private key used by the sshd daemon for version 1 of the SSH protocol.
ssh_host_key.pub
The RSA public key used by the sshd daemon for version 1 of the SSH protocol.
ssh_host_rsa_key
The RSA private key used by the sshd daemon for version 2 of the SSH protocol.
ssh_host_rsa_key.pub
The RSA public key used by the sshd for version 2 of the SSH protocol.

Las claves de usuario

La información para la configuración SSH específica para el usuario está almacenada en el directorio principal ~/.ssh/:

`Authorized_ keys`
Este archivo que contiene una lista de claves públicas autorizadas. Cuando un cliente se conecta al servidor, éste valida al cliente comprobando su clave pública firmada y almacenada dentro de este archivo.

Id_dsa Contiene la clave privada DSA del usuario.
Id_dsa.pub Contiene la clave pública DSA del usuario.
Id_rsa La clave RSA privada usada por ssh para la versión 2 del protocolo SSH.
Id_rsa.pub La clave pública RSA usada por ssh para la versión 2 del protocolo SSH.
identity La clave privada RSA usada por ssh para la versión 1 del protocolo SSH.
identity.pub La clave pública RSA usada por ssh para la versión 1 del protocolo SSH.

Known_hosts

Este archivo contiene las claves de host DSA de los servidores SSH accesados por el usuario. Este archivo es muy importante para asegurarse de que el cliente SHH está conectado al servidor SSH correcto.

Configurando el cliente con SSH2:

Para configurar el cliente con protocolo SSH2, cambie la línea correspondiente a la versión de ssh al archivo de configuración del cliente /etc/ssh/ssh_config:

```
#     $OpenBSD: ssh-config, v 1. 15 2002106/20 20:03:34 stevesk Exp $

# This is the ssh client system-wide configuration file. See
# ssh-Config (5) for more information. This file provides defaults
for # users, and the values can be changed in per-user configuration
files
# or on the command line.

# Configuration data is parsed as follows:
# 1. command line options
# 2. user-specific file
# 3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should he at the heginning of the
# configuration fíle, and defaults at the end.

# Site-wide defauits for various options

# Host *
# ForwardAgent no ForwardX 1 1 no
```

```
# RhostsAuthentication no
# RhostsRSAAuthentication no RSAAuthentication yes
# PasswordAuthentication yes BatchMode no
# CheckHostIP yes
# StrictHostKeyChecking ask
#IdentityFiie ~/.ssh/identity
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# Port 22
# Protocol 2,1
```

protocol 2

```
# Cipher 3des
#  Ciphers  aesl28-cbc,3des-cbc,blowfish-cbc,cast  l28-cbc,arcfour,aes
l92-cbc,aes256-cbc
# EscapeChar ~
Host *
ForwardX 1 1 yes
```

Configurando el server con SSH2:

Para configurar el server con protocolo SSH2, cambie la línea correspondiente a la versión de ssh al archivo de configuración del server /etc/ssh/sshd_config:

```
$OpenBSD: sshd-config,v 1.56 2002/06/20 23:37:12 markus Exp $
# This is the sshd server system-wide configuration file. See
# sshd_Config(5) for more information
# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin

# The strategy used for options in the default sshd_Config shipped
with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options change a
# default value.
#Port 22
#Cambie esta línea a protocolo 2
#Protocol 2,1
```

protocol 2

```
#ListenAddress 0.0.0.0
#ListenAddress ::
# HostKey for protocol version 1
# HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key

# Lifetime and size of ephemeral version 1 server key
#KeyRegeneration lnterval 3600
#ServerKeyBits 768
# Logging
#obsoletes QuietMode and Fascist Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO
# Authentication:
#LoginGraceTime 600
```

```
#PermitRootLogin yes
#Strictmodes yes
```

Generando las claves pública y privada del cliente:

```
Siga el siguiente procedimiento:

[root@redhat80 root} # ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/rlopez/.ssh/id_dsa):
Created directory /home/rlopez/.ssh
Enter passphrase (empty for no passphrase): elmer j food
Enter same passphrase again: eImer j food
Your identification has been saved in /home/rlopez/.ssh/id_dsa.
Your public key has been saved in /home/rlopez/.ssh/id_dsa.pub.
The key fingerprint is:
6c:ca:           56:5c:72:           1b:d4:b7:16:5e:cf.76:99:24:86:ce
rlopez@pentium4.linux.com
```

Archivos generados que son mis claves privada y pública.

~ /.ssh/id_dsa
~/.ssh/id_dsa.pub

Control del demonio sshd:

Para ver el estado del servidor OpenSSH (sshd) use:

```
 [root@redhat80 root]# chkconfig -list sshd
sshd  0:off      1:off  2:off      3:on  4:on  5:on      6:off
Para habilitar o deshabilitar sshd desde el arranque de Linux se usa
chkconfig
[root@redhat80 root]# chkconfig -level 345 sshd on/off
Para control de sshd durante una sesión use:
[root@redhat8O root]# /etc/rc.d/init.d/sshd opción
[root@redhat80 root]# service sshd opción
opciones :
Usage:                                        /etc/rc.d/init.d/xinetd
{/start/stop/status/restart/condrestart/reload }
```

Acceso al server con shell segura ssh

La shell segura ssh es un reemplazo seguro para los comandos rlogin, rsh y telnet. Para accesar al servidor llame como en una sesión telnet; pero ahora con shh

```
[root@redhat80 root]# ssh server.linux.com
```

La primera vez que entre al server ocurre lo siguiente:

```
The authenticity of host 'server.linux.com' can't be established.DSA
key                          fingerprint                          is
94:68:3a:3a:bc:f3:9a:9b:O1:5d:b3:07:38:e2:11:0c.Are you sure you want
to continue connecting (yes/no)?
```

Responda que si, le aparece el siguiente mensaje:

```
Warning: Permanently added 'serverlinux.com' (RSA) to the list of
known hosts.
```

Lo cual agrega serverlinux.com a la lista de hosts conocidos en el archivo ~/.ssh/known_hosts. Y se le muestra en pantalla el shell de la máquina remota. La próxima vez que inicie sesión ssh con el host remoto ya no se le preguntará lo anterior y la máquina remota envía su shell.
Si no se específica el login del usuario, el username con que se inicia sesión en la computadora local es pasado a la computadora remota. Si desea especificar diferente username se emplea el shell segura como se indica a continuación:

```
ssh username@server.linux.com
```

o también puede utilizar la siguiente sintaxis:

```
ssh -l username server.linux.com
```

La shell segura ssh también puede ser utilizado para ejecutar comandos en la computadora remota sin iniciar sesión y tener el shell de ésta, por ejemplo:

```
ssh penguin.example.net Is /usr/share/doc
```

Después de que se introduce el correspondiente password y el comando se muestra en pantalla, la máquina local retorna a su shell prompt.

Autentificación de cliente a server SSH:

- El cliente invoca un login al server y envía su clave pública.
- El server checa la clave pública del cliente, si la clave del cliente es válida genera un número aleatorio lo encripta con la clave pública del cliente y envía este valor al cliente.
- El cliente desencripta el valor con su clave privada genera un cheksum que envía de vuelta al server.
- El server calcula el cheksum de los datos recibidos y compara los cheksum Si son correctos la autentificación es aceptada

Autentificación en intercambio de datos:

- La computadora A solicita datos a la computadora B.
- La computadora B solicita la clave pública de A, A envía su clave pública. Con la clave pública de A, la clave privada de B y los datos. La computadora B encripta los datos. Los datos encriptados se envían a A.
- La computadora A pide la clave pública de B, B envía su clave pública. Con la clave pública de B, la clave privada de A. La computadora B desencripta los datos.

OpenSSH

Configuración del servidor sshd

La configuración del servidor ssh, no sólo consiste en elegir la versión del protocolo. Para una configuración más específica consulte la presente sección.

Tipos de Autorización.

Para iniciar la configuración de sshd, es importante, que sepa cuales son las distintas formas de autorización que desea utilizar para identificar una sesión de sshd, por ello, antes de comenzar con la configuración del demonio de sshd, mostraremos brevemente cuáles son estos tipos y su funcionamiento.

Los tres tipos de autorización que soporta el protocolo sshd:

1. A través de la clave
   Consiste en que el cliente se conecta con el servidor y nos solicita la contraseña de ese usuario.

2. El sistema de autorización por host
   La autorización mediante host consiste en instalar la clave pública del host (hostkey.pub) en el servidor al que deseamos conectarnos, y si el host y usuario desde el que accedemos se encuentra autorizado en alguno de los siguientes ficheros: hosts.equiv, shosts.equiv. rhosts, shosts. Y siempre que la configuración del demonio sea la adecuada permitiendo el uso o no de estos ficheros, entonces se conseguirá el acceso a la cuenta que estemos accediendo.

3. Autorización mediante el sistema de claves pública y privada
   Consiste en crear las dos claves y situar la pública en el ordenador remoto. Y la privada se mantiene en el host desde el que queremos acceder. Cuando nos conectemos se comprobarán las claves de ambos ordenadores con la contraseña que nos solicitará en caso de haber puesto una de estas claves. En caso de que la comprobación resulte correcta se nos permitirá al acceso.

Parámetros de Configuración.

Para configurar el demonio deberá editar el *fichero letclsshlsshd_Config* y agregar tantas secciones de configuración como desee, estas secciones se usan para poder mantener diferentes tipos de configuración según la dirección que acceda al servidor de SSH. Si lo desea es más sencillo mantener una única sección para todos los hosts, para declarar esta sección use *: y a partir de ahí, esta configuración se utilizara para todos los hosts a no ser que se declare otras secciones.

Configuración del demonio sshd

Allowed Authentications

Con esta opción especificamos cuáles de los métodos de autorización anteriormente explicados, permitimos para conceder la entrada. La opción por defecto es *publickey, password,* las posibles opciones son las siguientes: *publickey, passwordy hostbased.*
Allowed Groups
Permite especificar qué grupos son los que tienen acceso mediante SSH, se pueden usar comodines, por defecto se les permite el acceso a los usuarios pertenecientes a cualquier grupo. (Se pueden usar patrones para especificar los grupos).

AllowHost

Similar al anterior pero para especificar desde qué hosts se les permite el acceso mediante SSH.

AllowUsers

Permite especificar a que usuarios se les permite al acceso, por defecto se permite a cualquier usuario.

Authorization File

Especifica cuál es el fichero de autorización para cada usuario. Por defecto es ~/.ssh/ authorization

Check Mail

Si se activa esta opción cuando el usuario acceda al sistema, en caso de que tenga correo se lo hará saber. Por defecto esta opción está activada. Opciones “yes” o “no”

ChRoot Groups

Especifica a qué grupos se le aplicará chroot a su dirección home.

ChRoot Users

Similar al anterior pero se especifican patrones de usuarios, en vez de grupos de usuarios.

Deny Groups

Similar a AllowGroups pero para denegarles el acceso.

Deny Hosts

Similar a AllowHosts pero para denegarles el acceso.

Deny Users

Similar a AllowUsers pero para denegarles el acceso

HostKeyFile

Especifica cuál es el archivo que contiene la clave privada de ese ordenador. Por defecto es: /etc/.ssh/hostkey.

IdentifyFile

Especifica cuál es el archivo que contiene la clave pública del usuario. Por defecto *identification.*

IgnoreRhosts

Especifica si ignorar o no el contenido de los ficheros. shosts, rhosts. Opciones “yes” o “no” Por defecto “no”.

loginGraceTime

Tiempo (en segundos) antes del cual si no se ha autorizado un usuario se procederá a cerrar esa conexión. Por defecto “600”. Si usa 0 no habrá límite de tiempo.

MaxConnections

Especifica el número máximo de conexiones simultáneas permitidas. Si usa 0 no habrá límite.

PermitEmptyPasswords

Especifica si se permite el acceso a cuentas que no posean claves. Opciones “yes” o “no”. Es una buena idea desactivar esta opción pues numerosos agujeros de seguridad se dedican a crear una cuenta con uid 0 pero sin clave. De esta manera “evitaremos” este tipo de ataques.

PermitRootlogin

Especifica si se permite el acceso a la cuenta *root* del server ssh. Opciones “yes” “nopwd” o “no” donde nopwd solo permite el acceso a la cuenta del *root,* mediante cualquier método que no sea especificando la clave, es decir, sólo mediante hostbased o publickey.

Port

Especifica el puerto en el que escucha las conexiones del demonio de SSH.

PublicHostKeyFile

Especifica el fichero que contiene la clave pública de ese host. Por defecto /etc/ssh/hostkey.pub

RequiredAuthentications

Especifica cuáles de las autorizaciones permitidas son obligatorias para permitir el acceso a una cuenta.

ssh1Compatibility

Especifica si desea o no compatibilidad con la anterior versión de SSH. Opciones “yes” o “no”

sshd1Path

Especifica la localización del demonio de SSH, para la versión anterior.

UserConfigDírectory

Especifica cuál es el directorio donde se almacenan las configuraciones y opciones especificas para cada usuario. Por defecto ~/.ssh2/

UserKnownHosts

Especifica si los ficheros de $HOME/.ssh/knownhosts pueden usarse para las autorizaciones del tipo *hostbased.* Opciones “yes” o “no”.

Una vez realizadas las opciones más interesantes para la configuración del demonio de SSH, ya puede editar el fichero de configuración y asignar los valores que desee a estos parámetros, recuerden que éstos son los parámetros más interesantes; para ver la lista completa consulte man sshd_config.

3.9.2 scp y sftp

Copia remota de archivos y ftp seguros

scp copia remota de archivos segura

El comando scp se usa para transferir archivos entre computadoras remotas sobre una conexión encriptada y segura, es similar al comando R de Bekerly rcp.

La sintaxis para transferir un archivo local a una computadora remota es:

scp localfile username@tohostname:/newfilename local file archivo(s) de la computadora local a transferir

username@tohostname :/ newfilename destino de archivo(s) a transferir

La sintaxis para transferir un archivo remoto local a una computadora local es:

scp username@tohostname:/remotefile /newlocalfile remotefile archivo(s) de la computadora remota a copiar *newlocalfile* donde se copia el archivo(s) en la computadora local

Múltiples archivos pueden ser transferidos usando comodines, por ejemplo:

scp /downloads/ username@amaq2O.linux.com:/uploads/* o pueden transferirse arboles de archivos usando el comando con la opción recursiva: *scp -r*
sftp transferencia de archivos FTP seguro
sftp se usa para abrir una sesión FTP interactiva segura, es similar a ftp, excepto que se usa una conexión encriptada segura. Para iniciar una sesión se usa:
sftp username@server.linux.com
Una vez autentificado, la transferencia es similar a las opciones usadas por FTP. Vea la lista de comandos con: man sftp

3.10 Seguridad Linux en Internet

3.10.1 ¿Qué es un firewall?

Normalmente, es un direccionador, una computadora autónoma con filtro de paquetes o software *proxy,* o un paquete de firewall (un dispositivo de hardware patentado que filtra y hace proxies).
Un firewall puede servir como punto de entrada único a su sitio, normalmente llamado punto de estrangulamiento. A medida que se reciben las peticiones de conexión, el firewall las va evaluando. Sólo se procesan las peticiones de conexión de los hosts autorizados; el resto de las peticiones son descartadas.
Los firewall actuales realizan todo tipo de tareas, como:

- Filtro y análisis de paquetes. Los firewall pueden analizar paquetes entrantes de múltiples protocolos. Basándose en ese análisis, los firewall pueden realizar evaluaciones condicionales (“Si se encuentra este tipo de paquete, haré esto”).
- Bloqueo de protocolo y contenido. Los firewall le permiten proteger contenidos. Puede explotar esta capacidad para bloquear Java, JavaScript, VBScript, ActiveX y otras cosas en el firewall. De hecho, incluso puede crear normas para bloquear firmas de ataque particulares.
- Autentificación y encriptación de usuario, conexión y sesión. Muchos firewall utilizan varios algoritmos y sistemas de autentificación (DES, TripleDES, SSL, IPSEC, SHA, MD5, BlowFish, IDEA, etc.) para verificar la identidad de sus usuarios, comprobar la integridad de la sesión y proteger los datos en tránsito de los rastreos.

Dependiendo de su diseño, un firewall protege a su red al menos en dos de estos niveles (y en algunos casos en todos):

- Quién puede entrar.
- Qué puede entrar.
- Dónde y cómo pueden entrar.

En su comienzo, un firewall es un concepto más que un producto. Es la suma total de todas las normas que quiera aplicar a su red. Generalmente, proporcionará a su firewall normas que reflejen la normativa de acceso de su propia organización.
Existen dos tipos principales de firewall:

- Firewall a nivel de red, o filtros de paquetes.
- Pasarelas de aplicaciones.
-

Firewall a nivel de red: filtros de paquetes

Normalmente, los firewall a nivel de red son direccionadores con capacidades de filtro de paquetes. Al utilizar un firewall a nivel de red, puede dar o negar acceso a su sitio basándose en varias variables, como pueden ser:

- Dirección de fuente.
- Protocolo.
- Número de puerto.
- Contenido.

Los firewall basados en direccionadores son populares porque son soluciones de perímetro. Son dispositivos externos.

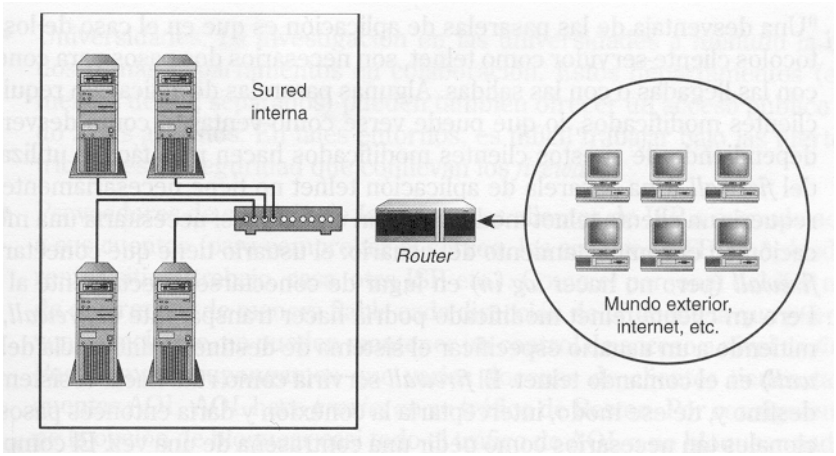


Fig. 3.15 Su direccionador es la única entrada desde el exterior

Como queda representado en la Figura todo el tráfico exterior debe pasar a Través de su direccionador, que manipula todos los procedimientos de aceptación y negación. Este método ofrece una gran ventaja: es de sistema operativo y neutral en cuanto a la aplicación. Por tanto, los firewall basados en direccionadores ofrecen una solución limpia y rápida.
Además, los firewall basados en direccionadores avanzados pueden vencer al spoofing y a los ataques DoS, e incluso convertir a su red en invisible para el mundo exterior.

Por otra parte, los firewall basados en direccionadores tienen algunas deficiencias algunos direccionadores son vulnerables a algunos ataques. Y su actuación puede deteriorarse cuando utilice procedimientos de filtrado excesivamente estrictos. Esto puede ser o no un problema dependiendo de la cantidad de trafico entrante que anticipe.
Por último, los basados en direccionadores buenos son caros y se obtiene por lo que se paga. Los sistemas baratos no mantienen el estado en los paquetes entrantes y son, por tanto, vulnerables a varios ataques.

Firewall de aplicación-proxy/pasarelas de aplicación

Las pasarelas de aplicación sustituyen a las conexiones entre los clientes externos y su red interna. Durante este cambio nunca se envían los paquetes IP En su lugar, se produce una especie de traducción, actuando la pasarela de conducto y de intérprete.
La otra cara de esto es que obtiene más control global sobre cada servicio individual. Y, en muchos casos, puede mantener la información del estado del paquete.
Sin embargo, las pasarelas de aplicación también tienen deficiencias. Una es que muchas de ellas requieren una implicación substancial por su parte porque debe configurar una aplicación proxy para cada servicio de la red. Además, los usuarios internos deben utilizar clientes que estén al tanto del proxy. Si no, tendrán que adoptar nuevas normativas y procedimientos.

3.10.2 tcpd: TCP Wrappers

Los TCP Wrappers son unas de las herramientas más conocidas del mundo para reforzar el control de acceso a la red.
Aplicación: tcpd.
Requiere: tcpd.
Archivos de configuración: hosts.deny, hosts.allow
Los TCP Wrappers añaden un control de acceso a la red a través de un sencillo pero seguro mecanismo.
En los *hosts* que no tienen TCP Wrappers, inetd comienza a comprobar varios servidores permitidos en /etc/inetd.conf. Aquí tenemos un inetd.conf típico de un *host* de ese tipo sin comentarios:

```
# Internet serve.r configuration database
# $Revision: 1.66 $
ftp      stream tcp nowait root /usr/etc/ftpd ftpd -1
telnet stream tcp nowait root /usr/etc/telnetd telnetd
shell stream tcp nowait root /usr/etc/rshd rshd
login stream tcp nowait root /usr/etc/rlogind rlogind
exec      stream tcp nowait root /usr/etc/rexecd rexecd
finger stream tcp nowait guest /usr/etc/fingerd fingerd
http      stream tcp nowait nobody ?/var/www/server/httpd httpd
ntalk dgram udp wait root /usr/etc/talkd talkd
tcpmux stream tcp nowait root internal
echo      stream   tcp   nowait    root  internal
cara   stream   tcp   nowait    root  internal
chargen   stream   tcp   nowait    root  internal
daytime   stream   tcp   nowait    root  internal
time      stream   tcp   nowait    root  internal
```

```
echo      dgram   udp   wait   root   internal
cara   dgram   udp   wait   root   internal
chargen   dgram   udp   wait   root   internal
daytime   dgram   udp   wait   root   internal
time      dgram   udp   wait   root   internal
```

Cada línea es una entrada separada y cada entrada especifica un servicio, su tipo de toma, su tipo de protocolo, el usuario de ejecución y el servidor. Por ejemplo, examine la entrada de fingerd:
finger stream tcp nowait guest /usr/etc/fingerd fingerd
Esto es lo que especifica la entrada fingerd:

- El servicio es finger.
- El tipo de toma es STREAM.
- El protocolo es TCP.
- La instrucción nowait indica que inetd debería generar nuevos procesos fingerd cuando se necesiten.
- La instrucción quest indica que fingerd debería ejecutarse como usuario quest.
- La instrucción /usr/etc/fingerd indica la localización del programa fingerd.

Cuando inetd recibe una petición de un cliente finger, comienza un fingerd, que después satisface la petición de finger. El motivo de esto es que es más fácil ejecutar un demonio sencillo como inetd, que ejecutar permanentemente 12 ó 20 servidores diferentes. De este modo, un servidor sólo se ejecuta si es necesario.
El Problema de este método es que estos servicios pueden no aplicar el control de acceso por defecto, por lo que no puede aceptar o negar conexiones selectivamente de una manera fácil.
Con los Tap Wrappers instalados, cuando inetd llama a un servidor, tcpd intercepta la llamada y evalúa la petición de conexión. Durante este proceso, tcpd compara la petición con respecto a varias normas. Si pasa estas pruebas, tcpd arranca el servidor requerido, que a su vez satisface la petición del cliente. Pero si la conexión no pasa, la evaluación de tcpd no será aceptada.
tcpd es un paquete bastante bueno. Cuando tcpd evalúa una petición de conexiones, también le hace *log* igual que sys log.

En resumen, los TCP Wrappers le dan dos poderosas ventajas:

- Logging de conexión.
- Control de acceso a la red.

Tcpd hace *log* de las conexiones sin su ayuda. Sin embargo, para el control de acceso a la red debe establecer las normas.

TCP Wrappers y control de acceso a la red

Los TCP Wrappers leen las normas del control de acceso a la red de dos archivos:

- /etc/hosts.allow. (especifica los hosts autorizados)
- /etc/hosts.deny. (especifica los hosts no autorizados)

En una instalación reciente, estos archivos están normalmente vacíos y tienen

este aspecto:

```
# hosts.deny This file describes the names of the hosts which are
#  *not* allowed to use the local INET services, as decided
# by the '/usr/sbin/tcpd' server

# The portmap line is redundante but it is left to remind you that

# the new secure portmap uses hosts.deny and hosts.allow.

# In particular
# you should know that NFS uses portmap !
```

Cómo configurar /etc/hosts.deny y /etc/hosts.allow

La configuración requiere alguna consideración hosts-options soporta gran cantidad de características y, a medida que se va familiarizando, puede desarrollar normas complejas como: "Si una conexión cumple con estos criterios, ejecuta este comando shell". Sin embargo, hasta que adquiera más experiencia, lo mejor es que se ciña a lo básico, que es esencialmente esto:

```
daemon-list : client list
```

Por ejemplo, supongamos que se ha introducido esta línea en /etc/hosts.allow-
.

```
ALL: mycompany.net EXCEPT techsupport.mycompany.net
```

Aquí, a todas las máquinas dentro de mycompany.net excepto a techsupport se les permite conectarse a todos los servicios. Esto es muy útil, pero sólo si también añade esta entrada a /etc/hosts.deny:

```
ALL: ALL
```

Si especifica sólo la entrada/etc/hosts.allow, el único *host* que se le niega la entrada es techsupport.mycompany.net

Por regla general, debería añadir ALL: ALL a su archivo /etc/hosts.deny en primer lugar. Eso niega el acceso a todo el mundo. Después de esto, puede empezar a introducir los *hosts* autorizados. La razón de todo esto es que es **más** sencillo y más seguro especificar que "aquél que no es autorizado es rechazado", que especificar que aquél que no es rechazado es autorizado". De este modo elimina posibles circunstancias desconocidas.

Supongamos que /etc/hosts.deny contiene estas entradas:

```
ALL: aol.com, msn.com
All EXCEPT in.telnetd: techsupport.theircompany.net
```

Aquí se bloquea a la gente de AOL y de MSN, pero aquellos que están en el host techsupport.theircompany.net pueden acceder a sus servicios telnet.

All	Utilícelo para generalizaciones de barrido, incluidos los servicios All y los *hosts* remotos ALL. Por ejemplo, ALL: ALL en /etc/hosts.deny niega el

	acceso a todos los servicios a todos los *hosts. (Por* el contrario, ALL: ALL en /etc/hosts.allow permite el acceso a todos los servicios a todos los *hosts,* algo que, definitivamente, querríamos hacer.)
KNOWN	Utilícelo cuando quiera aplicar una norma a usuarios y *hosts* que sean explícitamente nombrados en sus normas de control de acceso.
LOCAL	Utilícelo para nombres de *host* que no tengan puntos (como su *host* local).
PARANOID	Utilícelo si quiere que tcpd suprima *hosts* cuando su nombre no cuadre con su dirección IP
UNKNOWN	Utilícelo cuando quiera negar el acceso a *hosts* o nombres de usuario desconocidos. En otras palabras, si estos usuarios y *hosts* no están explícitamente nombrados en sus normas de control de acceso, se les negará la entrada.

Tabla 3.19 Comodines, operadores y funciones Shell de hosts-Options

El operador EXCEPT

Finalmente, hosts-options soporta un operador: EXCEPT. Puede utilizar EXCEPT para crear excepciones a normas específicas en listas de clientes o demonios. Supongamos que introduce esta línea en /etc/hosts.deny:

```
ALL EXCEPT in.telnetd: techsupport.mycompany.net
```

Aquí, está negando todos los servicios excepto telnet al *host* techsupport. Pero también puede agrupar declaraciones EXCEPT, de este modo:

```
list EXCEPT list EXCEPT list
```

Esto puede ser complicado incluso sin añadir comandos *shell* ejecutados condicionalmente. Por consiguiente, TCP Wrappers viene con herramientas que puede utilizar para verificar sus normas:

- tcpdchk.ElcomprobadordeconfiguracióndeTCPWrappers.
- tcpdmatch.EloráculodeTCPWrappers

tcpdchk: el comprobador de configuración de TCP Wrappers

tcpdchk examina su configuración de TCP Wrappers e informa de todos los problemas potenciales y reales que pueda encontrar. El programa examina los archivos de control de acceso de tcpd (por defecto son /etc/hosts.allow y /etc/hosts.deny), y compara las entradas en estos archivos con las entradas en los archivos de configuración de red de inetd o de tlid.

ttcpdchk analiza su configuración para los siguientes problemas:

- Mala sintaxis.
- Malos nombres de ruta.
- Malos nombres de *host* o direcciones de IP.
- Nombres de *host* con direcciones de IP que no corresponden (una extensión de la funcionalidad del comodín PARANOID).
- Servicios en los que especifica normas, pero que no están cubiertos por tcpd.

Opción	
-a	Utilícela para especificar que tcpdchk debería informar sobre las normas de permiso que no vayan acompañadas por un comodín ALLOW explícito.
-d	Utilícela para especificar que tcpdchk debería comprobar las normas de hosts.allow y hosts.deny en el directorio actual en lugar de /etc. Esto es útil si está creando normas en otro directorio antes de utilizarlas.
-i [inetd.conf]	Utilícela para especificar un inetd.conf alternativo. tcpdchk necesita saber qué inetd.conf está utilizando, si no es el valor por defecto, por que evalúa si los servicios a los que ha aplicado normas de control de acceso están cubiertos.
-V	Utilícela para obtener una salida ampulosa y limpiamente formateada.

Tabla 3.20 Opciones de línea de comando

Mientras que tcpdchk comprueba sus normas para asegurar que son sólidas, tcpdmatch le muestra lo que ocurrirá cuando se utilicen. "tcpdmatch predice cómo manipularían los TCP Wrappers una petición de servicio específica."
La sintaxis es tcpdmatch [daemon] [host], de este modo:

```
tcpdmatch in.telnetd techsupport.theircompany.net
```

3.10.3 Ipchains

Ipchains, disponible en el paquete *kernel* 2.2, es el sucesor de ipfwadm y soporta toda la funcionalidad de ipfwadm y más. La diferencia principal, desde el punto de vista de su uso, es que los comandos están ahora en mayúsculas, mientras que los argumentos están en minúsculas

-A	Utilice este comando para agregar una norma nueva a la cadena. En ipfwadm, era antes -a.
-D	Utilice este comando para eliminar una norma de una cadena. En ipfwadm, era antes -d.
-F	Utilice este comando para limpiar todas las normas de una cadena o cadenas. En ipfwadm, era antes -f.
-I	Utilice este comando para insertar una norma a una cadena. En ipfwadm, era antes -i.
-L	Utilice este comando para listar todas las normas de una cadena. En ipfwadm, era antes -l.
-P	Utilice este comando para cambiar las normativas por defecto de una cadena. En ipfwadm, era antes -p.
-R	Utilice este comando para reemplazar una norma en una cadena.

Tabla 3.21 ***Comandos de ipchains***

Objetivo	
ACCEPT	Utilice este objetivo para permitir que el tipo de paquete descrito pase a través del firewall. Obsérvese de que ahora debe expresarse en mayúsculas.
DENY	Utilice este objetivo para denegar un paquete definitivamente. Obsérvese de que ahora debe expresarle en mayúsculas.

MASQ	Utilice este objetivo para aceptar el paquete descrito y dirigirlo a la red interna. Obsérvese de que ahora debe expresarle en mayúsculas.
REDIRECT	Utilice este objetivo para redireccionar el paquete descrito a un enlace o proceso local. Obsérvese de que ahora debe expresarle en mayúsculas.
REJECT	Utilice este objetivo para echar abajo un paquete y mandar el mensaje "ICMP Host Unreachable" (Host ICMP inaccesible). Obsérvese de que ahora debe expresarle en mayúsculas.

Tabla 3.22 Objetivos para ipchains

	Función
-b	Utilícelo para especificar que la norma especificada debería aplicarse sin importar la dirección (entrante o saliente) que tome el paquete.
-d ! [dirección]	Utilícelo para especificar la dirección destino. En ipfwadm, era antes -D
-i ! [interfaz]	Utilícelo para especificar la interfaz de red. En ipfwadrn, era antes –W.
-p ![protocolo]	Utilícelo para especificar el protocolo. En ipfwadm, era antes -P.
-s ![dirección]	Utilícelo para especificar la dirección de fuente. En ipfwadm, era antes -S.

Tabla 3.23 Predicados de Ipchains

Vamos ahora a mencionar solo a IPFWADM para tener una referencia de que se esta hablando con respecto a IPCHAINS

Ipfwadm

Ipfwadm es una herramienta. De filtrado de paquetes para Linux. Ipfwadm se utiliza para configurar, mantener e inspeccionar el firewall y las normas de cuenta en el kernel de Linux. Estas normas pueden dividirse en cuatro categorías diferentes:

- Cuenta de paquetes de IP.
- El firewall de entrada de IP.
- El firewall de salida de IP.
- El firewall de envío de IP.

Para cada una de estas categorías se mantiene una lista de normas separadas

3.10.4 Configurar un cortafuegos IP

Para configurar un cortafuegos que simplemente hace filtrado de IP, haremos lo siguiente.

1. Entre en el sistema como root.
2. Escriba "ipchains -L" para ver qué reglas y política están ya establecidas, si es que las hay. Debería obtener las siguientes a no ser que tenga algunas reglas establecidas:

```
Chain input (policy ACCEPT):
Chain forward (policy ACCEPT):
Chain output (pólicy ACCEPT):
```

3. Escribiremos "ipchains -P input DENY" para establecer una política para la cadena input que rechaza cualquier tráfico de entrada IP que no se ajusta a una regla individual.
4. Ahora debemos pensar seriamente en el tipo de tráfico al que quiere permitir el acceso a través del cortafuegos hacia su red. Sus reglas serán leídas en el orden que las agregue. Aunque podemos insertar elementos en medio de reglas existentes empecemos anotando cosas en un papel e intente imaginar cómo podrían fluir. Las reglas que tienen más probabilidad de coincidir con los datos entrantes deberían ir primero, yendo a lo más específico y a lo ya fijado. En otras palabras, intente ir de lo genérico a lo específico.
5. Después de saber lo que queremos agregar, es el momento de añadirlo. Aquí es donde su sintaxis adquiere importancia. Con el fin de configurar un cortafuegos IP, las sentencias de creación de su regla empezarán con **ipchains** -A **input** o **ipchains –A -output,** dependiendo de si estamos intentando regular el tráfico entrante o saliente de la red.
6. Crear la propia sentencia de la regla requiere precisión. El apartado "Cortafuegos de filtrado", se ocupa de los trozos de regla individuales. Juntemos las sentencias de regla para crear las reglas lo más precisas que se pueda.
7. Señalemos el último destino para los paquetes que coincidan con nuestras reglas y asignemos la opción final adecuada como se describe anteriormente en la tabla de objetivos de Ipchains. Para ilustrar todo esto con un ejemplo, supongamos que deseamos controlar los paquetes entrantes, lo que significa comenzar con **ipchains** -A **input,** para evitar que tengan acceso a cualquiera de nuestros servidores Web internos, que solamente se mencionan para uso de intranet. El tráfico desde Internet llega a través de una interfaz primaria en la máquina cortafuegos, que es eth0, y el tráfico hacia la red oculta va a través de la interfaz secundaria, eth 1. Al mirar en /etc/services vemos que el tráfico HTTP viaja a través de TCP y UDP al puerto 80 por defecto. Las máquinas que queremos que permanezcan ocultas se encuentran todas en la red privada 192.168.60.0, por lo que la sentencia con la que acabamos es la siguiente:

```
ipchains –A input i eth0 –d 192.168.60 / 225.225.225.0   80 - j
REJECT.
```

8. Continuemos añadiendo las reglas hasta que tengamos las que creamos que son necesarias en el lugar correcto.
9. Ahora, necesitamos asegurarnos de que estas cadenas no se pierdan cuando reiniciemos la máquina. Linux no las guarda de forma automática. Escriba "ipchains-save>/root/firewall_chains" con el fin de ejecutar el script que guarda las cadenas, y envíe los datos a /root/firewall_chains. O bien escoja otra ubicación si lo desea.
10. Escriba después “ vi/etc/rc.d/rc.local ” para abrir el ultimo archivo de inicio del sistema.
11. Escriba "G" para saltar al final del archivo.
12. Escriba "o" para abrir una nueva línea al final de este archivo.
13. Escriba "ipchains-restore < /root/firewall_chains" para ejecutar el script. Que lanza los datos de /root/firewall_chains y reconstruye el cortafuegos de forma automática.
14. Pulse la tecla Esc y escriba "ZZ" para guardar y cerrar el archivo.

3.10.5 Configurar un cortafuegos Proxy

Para configurar un cortafuegos de Proxy, haremos lo siguiente:

1 Entre en el sistema como root.
2 Lea a continuación la licencia FWPK en www.tis.com/research/software/ fwtk_readme.html.
3 Siga las instrucciones para aceptar la licencia si está de acuerdo. Si no, puede que necesite elegir otra herramienta.
4 Debería obtener un correo electrónico de respuesta que le diga de dónde puede descargar archivos usando FTP para conseguir el equipo de herramientas. Debe utilizar una herramienta FTP ya que un navegador Web no funcionará.
5 Descargue el archivo equivalente a fwtk2.1.tar.Z. También, consiga la versión sólo doc, pues es el único modo de obtener cualquier documentación si la necesita. No se incluye documentación en la página principal.
6 Desempaquete y descomprima el archivo en /usr/sre (o su ubicación preferida para el código fuente).
7 Escriba "cd fwtk" para cambiar al nuevo directorio.
8 Escriba "mv Makefile.config Makefile.config.original" para cambiar de nombre el archivo de configuración Makefile original.
9 Escriba "cp Makefile.config.linux Makefile.config" para utilizar el archivo de configuración Makefile específico de Linux como punto de partida.
10 Escriba luego "vi Makefile.config" para abrir este archivo de configuración para editar.
11 Coloque el cursor en una línea similar a:
CC=cc
12 Escriba "i" para introducir el modo Insertar.
13 Cambie la línea para leer:
CC=gcc
14 Pulse la tecla Esc y luego escriba "ZZ" para guardar y cerrar el archivo.
15 Escriba "Is/usr/X11 ". Si obtiene un mensaje similar a "Archivo no encontrado", entonces escriba "In-s/usr/X11R6/usr/X11” para crear un

enlace simbólico con objeto de que el compilador pueda encontrar las librerías que necesita.

16 Escriba "make" para construir el equipo de herramientas del cortafuegos. Esto le puede llevar varios minutos dependiendo de la velocidad de su ordenador.

17 Escriba "make instala" para colocar los archivos del programa en el lugar adecuado.

18 Escriba "vi/usr/local/etc/netperm-table" para abrir el archivo de configuración del daemon **netacl.** Este archivo afortunadamente contiene valores predeterminados con los que puede trabajar.

19 Lea este archivo con cuidado y edite donde crea apropiado.

20 Cuando termine, pulse la tecla Esc y después escriba "ZZ" para guardar y cerrar el archivo.

21 Escriba "vi/etc/rc.d/rc. local" para abrir el último archivo de inicio de sistema para que se ejecute al arrancar.

22 Escriba "G" para ir al final del archivo.

23 Escriba a continuación "o" para abrir una línea nueva para editar al final del archivo.

24 Para cada servicio para el que decida utilizar Proxy, necesita ejecutar **netacl** en su puerto por defecto. Escriba una serie de líneas en el siguiente formato para cada servicio:
/usr/local/etc/netacl –daemon puerto service servicio

25 Cuando haya terminado, pulse la tecla Esc y después escriba "ZZ" para guardar y cerrar el archivo.

26 Reinicie la máquina y asegúrese de que el servidor funciona como es de esperar.

En el presente capitulo tratamos el tema de la segurida en, especifico en Linux y como por medio del mismo podemos controlar los acesso de nuestros usuarios a la red, tambien pudimos conocer como hacer una detección de intrusos y como los codigos dañinos son tratados en Linux y que son.
Tambien tratamos el tema de cómo personalizar nuestro propio firewall tambien es necesario mencionar que el propio Linux puede definir por default pero es mejor personalizarlo, probamos los snifer (escuchas) y como podemos implementarlos en nuestro favor y nuestra red, tambien se toco el tema del secure shell(ssh).
En el siguiente capitulo veremos ahora la seguridad a nivel leyes y legislaciones, y como la falta de esta en nuestro pais ocasiona problemas a nivel seguridad en los sistemasy como atañe a la falta de una cultura informatica, y damos una propuesta de cómo reducir esto y comenzar a implantar una cultura informatica que comienza en nosotros mismos.

IV. HACIA UNA LEGISLACIÓN INFORMÁTICA EN MÉXICO

En estos últimos años la tecnología de la información y la comunicación han revolucionado nuestras vidas en todos los ámbitos como el social, científico, comercial, laboral, profesional, etc.
La tecnología avanza a una velocidad asombrosa y el derecho en particular el derecho mexicano se ha quedado muy rezagado en la regulación de una materia que lo ha rebasado, y que ha perdido la carrera en cuanto al crimen cibernético.
También tenemos que tomar en consideración que el derecho surge como un medio para regular la conducta del hombre en sociedad, pero es necesario destacar que la sociedad no es la misma en cada uno de los lugares del mundo ni la misma en cada momento de la historia.
El derecho regula la conducta y los fenómenos sociales a través de leyes, mismas que deben de acatar los integrantes de una sociedad.

4.1 Derecho informático.

Cuantas veces no hemos escuchado en las noticias que alguna persona fue estafada por medio del Internet o le pusieron una trampa con un portal parecido al de un banco (phishing), o que algún hacker hizo de las suyas con alguna información valiosa de alguna empresa, o el caso de alguna persona extorsionada por la red.
Todos los ejemplos señalados anteriormente son denominados como delitos informáticos[87] y para ellos también se creo el derecho informático que es una rama de las ciencias jurídicas que contempla a la informática como instrumento, es decir, a la informática jurídica[88], y como objeto de estudio, considera al derecho de la informática.
A diferencia de la política informática, la legislación informática es un conjunto de reglas jurídicas de carácter preventivo y correctivo derivadas del uso de la informática, es decir, que aquí se trata de una reglamentación de puntos específicos, pero esta circunstancia implica las siguientes consideraciones:

- Se recurriría a un cuestionamiento de reglas existentes para determinar si es posible su aplicación análoga frente al problema o si fuera necesaria una ampliación en cuanto a su ámbito de cobertura.
- Esperar la evolución de la jurisprudencia mediante la creciente presentación de casos ante los órganos jurisdiccionales pautas resolutorias o conciliatorias.
- Crear nuevas reglas integrándolas a ordenamientos ya existentes, o en un caso dando lugar a una nueva ley de carácter específico.

[87] Son todas aquellas conductas ilicitas susceptibles de ser sancionadas por el derecho penal, que hacen referencia al uso indebido de cualquier medio informático.
[88] Es el conjunto de estudios e instrumentos derivados de la aplicación de informática al derecho, es decir, a los procesos de creación, aplicación y conociemiento en materia del derecho

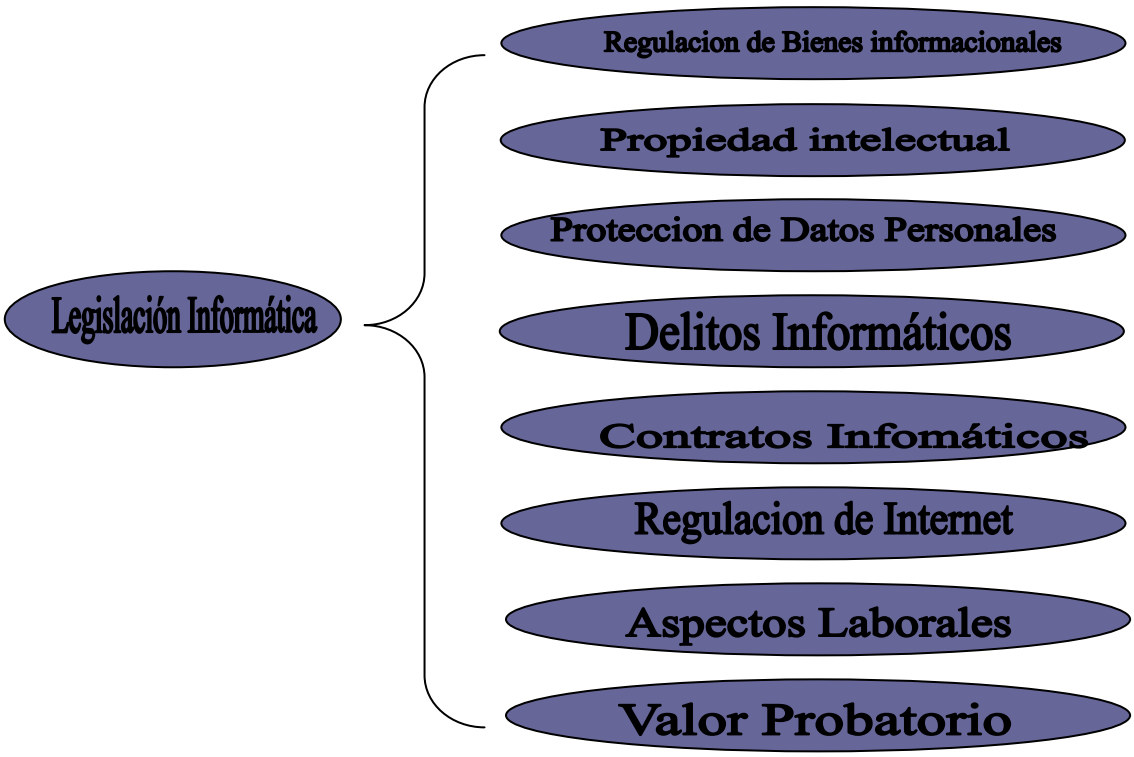


Fig. 4.1 Esquema de una legislación informática

4.1.1 Problemática a la que se enfrenta la legislación informática

Regulación de bienes informacionales: Esto es porque la información como producto informático requiere un tratamiento jurídico en función de su indudable carácter significativo y económico.
Protección de datos personales: Se refiere al atentado a los derechos fundamentales de las personas provocado por el manejo de informaciones nominativas.
Regulación jurídica de internet implica favorecer o restringir la circulación de datos a través de las fronteras nacionales.
Propiedad Intelectual e informática: Debe abarcar los temas de protección de los programas de cómputo y regulación de nombres dominio, ambos derivados de las acciones de piratería.
Delitos informáticos: Sancionar la comisión de verdaderos actos ilícitos en los que se tengan a las computadoras como instrumento para realizarlos.
Contratos informáticos: En función de esta categoría contractual con evidentes repercusiones fundamentalmente económicas.
Comercio electrónico: Nueva forma de comercialización automatizada de bienes y servicios de todo tipo.
Aspectos laborales de la informática: Como aquellos problemas laborales suscitados por la informatización de actividades. Ergonomía y teletrabajo.
Valor probatorio de los soportes modernos: *P*rovocado por la dificultad en la aceptación y apreciación de elementos de prueba derivados de estos soportes entre los órganos jurisdiccionales.

4.2 Lagunas legales en México en materia informática.

La seguridad en cómputo no debe de enfocarse como una situación exclusivamente informática, relacionada con las intrusiones o la alteración de los datos en las redes o los sistemas; debe incluir, además, cuestiones como el resguardo de la información y un proceso de toma de decisiones del mas alto nivel, que se refleje en políticas y estrategias a seguir.
Es necesario definir aspectos que hasta ahora solo has estado en el limbo en nuestro país, como una protección efectiva de la privacidad de los datos personales, para de ahí derivar la protección informática.
La actual falta de protección adecuada a la privacidad puede conducir a la explotación impune de la información personal, por meramente poner un ejemplo, la venta o comercialización ilegal de bases de datos en nuestro país con la información de los ciudadanos ya ha causado estragos ha demostrado que la legislación mexicana enfrenta severas lagunas y vacíos.
Ya que falta la aprobación de una ley de protección a los datos personales que imponga responsabilidades al sector público que dispone de grandes bases de datos con información la sociedad.
Nuestra constitución requiere declarar la protección a los datos personales o privados, ya que solamente contempla las garantías relacionadas de inviolabilidad del domicilio o inviolabilidad de la correspondencia.
¿Frente a esto nuestro poder legislativo que ha hecho?
Ha hecho adaptaciones a algunas leyes como:

- Ley federal de los derechos de autor.
- Ley federal de telecomunicaciones.
- Código penal federal.
- Código civil federal.

A continuación analizamos unos delitos informáticos.

- Fraude mediante el uso de computadoras y la manipulación de información que contengan estas (mejor conocido como la técnica salami)

El artículo 231 del código penal para el D.F. en su fracción XIV nos dice claramente lo siguiente:
Se impondrán las penas previstas en el artículo anterior, a quien:
“Para obtener algún beneficio para si o para un tercero, por cualquier medio accese, entre o se introduzca a los sistemas o programas de informática del **sistema financiero** e indebidamente realice operaciones, transferencias o movimientos de dinero o valores, independientemente de que los recursos no salgan de la institución;”
¿Solo esto se aplica para el sistema financiero? ¿Y que pasa con los que crean su propio portal “financiero” y engañan a la gente para sacarles el numero de cuenta y datos personales?

- Accesos no autorizados a sistemas o servicios y destrucción de programas o datos.

El código penal federal en el titulo noveno, referido a la revelación de secretos y acceso ilícito a sistemas y equipos de informática, que en su capitulo II prescibe lo siguiente:

Articulo 211 bis 1. “al que sin autorización modifique, destruya o provoque pérdida de información contenida en sistemas o equipos de informática **protegidos por algún mecanismo de seguridad**, se le impondrá de seis meses a dos años de prisión y de cien a trescientos días de multa”.
Aquí tendrían que ser más explícitos que es ese algún mecanismo de seguridad, si yo no tengo ese algún mecanismo de seguridad ¿Que pasa?
“al que sin autorización conozca o copie información contenida en sistemas ó equipos de informática protegidos por algún mecanismo de seguridad, se le impondrán de tres meses a un año de prisión y de cincuenta a ciento cincuenta días multa.”

Articulo 211 bis 2. “al que sin autorización modifique, destruya o provoque pérdida de información contenida en sistemas o equipos de informática del estado, **protegidos por algún mecanismo de seguridad**, se le impondrá de uno a cuatro años de prisión”.
“al que sin autorización conozca o copie información contenida en sistemas o equipos de informática del estado, protegidos por algún mecanismo de seguridad, se le impondrán de seis meses a dos años de prisión y de cien a trescientos días multa”.

Articulo 211 bis 3. “Al que estando autorizado para acceder a sistemas y equipos de informática del estado, indebidamente modifique, destruya o provoque pérdida de información que contengan, se le impondrán de dos a ocho años de prisión y de trescientos a novecientos días multa.
Al que estando autorizado para acceder a sistemas y equipo de informática del estado, indebidamente copie información que contengan, se le impondrá de uno a cuatro años de prisión y de ciento cincuenta a cuatrocientos cincuenta días multa”.

Artículo 211 bis 4. **“**Al que sin autorización modifique, destruya o provoque pérdida de información contenida en sistemas o equipos de informática de las instituciones que integran el sistema financiero, protegidos por algún mecanismo de seguridad, se le impondrán de seis meses a cuatro años de prisión y de cien a seiscientos días multa”.
“Al que sin autorización conozca o copie información contenida en sistemas o equipos de informática de las instituciones que integran el sistema financiero, protegidos por algún mecanismo de seguridad, se le impondrán de tres meses a dos años de prisión y de cincuenta a trescientos días multa”.
Artículo 211 bis 5. **“**Al que estando autorizado para acceder a sistemas y equipos de informática de las instituciones que integran el sistema financiero, indebidamente modifique, destruya o provoque pérdida de información que contengan, se le impondrán de seis meses a cuatro años de prisión y de cien a seiscientos días multa”.
“Al que estando autorizado para acceder a sistemas y equipos de informática de las instituciones que integran el sistema financiero,

indebidamente copie información que contengan, se le impondrán de tres meses a dos años de prisión y de cincuenta a trescientos días multa".
Las penas previstas en este artículo se incrementarán en una mitad cuando las conductas sean cometidas por funcionarios o empleados de las instituciones que integran el sistema financiero.

Artículo 211 bis 6. **"**Para los efectos de los artículos 211 bis 4 y 211 bis 5 anteriores, se entiende por instituciones que integran el sistema financiero, las señaladas en el artículo 400 bis de este Código".

Artículo 211 bis 7. "Las penas previstas en este capitulo se aumentaran hasta en una mitad cuando la información obtenida se utilice en provecho.
Seguimos viendo como están nuestras leyes".

- Uso no autorizado de programas y bases de datos

En la ley Federal del Derecho de Autor en sus artículos 106 a 110 que es lo más relevante tenemos lo siguiente:

Artículo 106. El derecho patrimonial sobre un programa de computación comprende la facultad de autorizar o prohibir:

I. La reproducción permanente o provisional del programa en todo o en parte, por cualquier medio y forma;

II. La traducción, la adaptación, el arreglo o cualquier otra modificación de un programa y la reproducción del programa resultante;

III. Cualquier forma de distribución del programa o de una copia del mismo, incluido el alquiler, y

IV. La descompilación, los procesos para revertir la ingeniería de un programa de computación y el desensamblaje.

Artículo 107. "Las bases de datos o de otros materiales legibles por medio de máquinas o en otra forma, que por razones de selección y disposición de su contenido constituyan creaciones intelectuales, quedarán protegidas como compilaciones. **Dicha protección no se extenderá a los datos y materiales en sí mismos".**

Que quieren decir con que no se extenderá a los datos si una base de datos tiene datos, ó ¿tiene otra cosa?

Artículo 108. Las bases de datos **que no sean originales** quedan, sin embargo, protegidas en su uso exclusivo por quien las haya elaborado, durante un lapso de 5 años.

¿A que se refieren con que no sean originales, y porque cinco años o sea le dan 5 años para lucrar con los datos, ese tiempo hasta le queda largo no?

Artículo 109. El acceso a información de carácter privado relativa a las personas contenidas en las bases de datos a que se refiere el artículo anterior, así como la publicación, reproducción, divulgación, comunicación pública y transmisión de dicha información, requerirá la autorización previa de las personas de que se trate.

Quedan exceptuados de lo anterior, las investigaciones de las autoridades encargadas de la procuración e impartición de justicia, de acuerdo con la legislación respectiva, así como el acceso a archivos públicos por las personas autorizadas por la ley, siempre que la consulta sea realizada conforme a los procedimientos respectivos.

Artículo 110. El titular del derecho patrimonial sobre una base de datos tendrá el derecho exclusivo, respecto de la forma de expresión de la estructura de dicha base, de autorizar o prohibir
I. Su reproducción permanente o temporal, total o parcial, por cualquier medio y de cualquier forma;
II. Su traducción, adaptación, reordenación y cualquier otra modificación;
III. La distribución del original o copias de la base de datos;
IV. La comunicación al público, y
V. La reproducción, distribución o comunicación pública de los resultados de las operaciones mencionadas en la fracción II del presente artículo.
En conclusión podemos observar lo siguiente.
Protege como compilaciones a las bases de datos legibles por medio de máquinas que por razones de disposición de su contenido constituyan obras intelectuales, otorgándole a su organizador el uso exclusivo por cinco años; a si mismo, exceptuando las investigaciones de autoridades, la información privada de las personas contenida en bases de datos no podrá ser divulgada, transmitida ni reproducida salvo con el consentimiento de la persona de que se trate.
Haciendo una retrospectiva y un análisis más a fondo, si no mal recuerdo por ahí escuche que se había vendido alguna vez el padrón electoral a una compañía de tele marketing, creo que gracias a esta laguna ¿paso eso no?, o pensemos en la actualidad y á quien no le ha pasado que le llaman a su casa u oficina ofreciéndole servicios, ya sea de una tarjeta o cualquier otro producto, de ¿donde sale la información si usted no la proporciono previamente al banco?¿se lo ha preguntado? Pues la respuesta es sencilla al negocio de vender y comprar bases de datos quizás podamos adquirir algunas incluso en la red ¿de que las quiere? de ingenieros, doctores o talvez abogados

- Intervención de correo electrónico

En el articulo 167 fr. VI del código penal Federal sanciona con uno a cinco años de prisión y 100 a 10000 días de multa al que dolosamente o con fines de lucro, interrumpa o interfiera comunicaciones alámbricas, inalámbricas o de fibra óptica, sean telegráficas, telefónicas o satelitales, por medio de las cuales se transmitan señales de audio, video o de datos.
Nota: Aquí tipificaran el interceptar un correo antes de que llegue a su destinatario, pero el abrir el buzón o los correos una vez recibidos. Y que hay con respecto de los sniffers.

4.3 Insuficiencia de Recursos.

Realmente no hay insuficiencia de recursos porque con poco podemos hacer mucho realmente tratándose de Linux, lo que nos hace falta es una cultura informática, y una conscientisacion de las personas que tienen una máquina enfrente.
Retomando nuestro tema de interés los recursos de una empresa que nos importan y nos atañen son los siguientes cuanto a la seguridad de datos personales e información confidencial:

- Recursos informáticos.
- Recursos legales.
- Las TI (Tecnología de la información) como recurso.
- Recursos Humanos.

Recursos informáticos: Los recursos informáticos de una institución, son todos los datos que involucran información personal de la gente que trabaja ahí o la propia institución, que es invaluable para la empresa y las personas que para dicha empresa laboran, y que puede contener información de su operación y valiosa para los competidores, por ello es para nuestro menester un punto crucial en cuanto seguridad informática, ya que es lo que vamos a proteger a toda costa y evitar que se haga mal uso de ella.

Recursos legales: bueno realmente en cuanto este punto resulto ser un punto de suma importancia para esta tesis, porque expongo las faltas de estos recursos en México, y las lagunas legales en materia de protección de datos.
Como se pudo ver en párrafos anteriores no hay una ley, tal cual en forma, ni una legislación que proteja los datos informáticos.
Por otro lado las grandes dudas y preguntas que me cuestiono son las siguientes: ¿Que hacer ante un atentado de este tipo?
¿Con quien denuncio dicho ilícito?
¿Que hago si me roban mis datos personales?
¿Que hacen las autoridades y las leyes ante esto?

Y lo que encontré tristemente es que las autoridades y las leyes no hacen absolutamente nada.
En parte mi propuesta ante este panorama es Linux como sistema operativo para las empresas, y usuarios, pero me tope con otro problema y es la cultura informática, al parecer el problema radica en que somos esclavos de Microsoft, y no queremos experimentar otro software dejando así fuera a otros proveedores de software.

Las TI (Tecnología de la información) como recurso.

Hoy en día las Ti son una herramienta desde la pequeña, la media y las macro empresas ya que también se encarga de la transferencia así como la recuperación de información, así como los sistemas de procesamiento personal de información en las cuales influye de manera eficiente esta

tecnología, ya que si desde un principio, ambos puntos que se tocaron anteriormente son seguros y fiables no tendremos problemas mas adelante.
Por otro lado con el incremento del uso de Internet y e-mail y con el desarrollo de intranets o redes de comunicaciones entre empresas, esta acelerando el flujo de datos de personas e información en las empresas.
Gracias a estos recursos también permiten la conexión de computadoras personales a potentes servidores. Actualmente las redes de computadoras son usadas como el canal primario de información interna de una organización.
Con las TI podemos tener información clara y oportuna acerca de todos los movimientos del entorno industrial y empresarial, mientras no sean corrompidas.
Dichas tecnologías mejoran la productividad de todas las funciones de la empresa y mejora el flujo de la información.
Las nuevas tecnologías y aplicación de los estándares ya existentes en las industrias TI hacia el mercado de los proveedores de dispositivos de seguridad corporativa ha modificado el pérfil de las soluciones que requieren las empresas. Ahora se habla de video digital en vez del sistema habitual de videocámaras para monitorear la actividad de los empleados y las visitas; así mismo, se lleva un registro de los correos electrónicos que entran y salen de la red corporativa para verificar la información contenida en ellos y prevenir el hurto de datos.
Los mecanismos biométricos y los sistemas de detección de metales también se han desarrollado para poder detectar los cada vez mas ágiles e ingeniosos intentos de fraude; ya no es útil un dedo sin su dueño, pues estos sistemas pueden reconocer un miembro sin vida, así como los detectores son capaces de ubicar armas de fuego que se introducen en partes, bombas caseras, y otras amenazas a la seguridad corporativa, disfrazadas de pilas o botellas de spray.
Las nuevas exigencias de las empresas exigen que los sistemas de seguridad se integren de forma optima en una red corporativa para ofrecer mayor seguridad, empequeñecer los huecos que se puedan generar entre las distancias areas, y minimizar costes de operación e implantación de las tecnologías.

Recursos humanos.

A nuestra empresa le lesiona, porque si no elegimos correctamente a las personas que van a integrarse a nuestra empresa a laboral, en un mañana lo podremos lamentar, y esto porque tendrán acceso a nuestras instalaciones y por consiguiente en cierta medida a la información que ahí se maneja, aquí el punto es crear una política interna para nuestros empleados comprometiéndolos como parte de la empresa.
Por otro lado en este punto también hay que tocar la seguridad física, como comenzarnos quizás a preguntar:

- ¿Que personal es de confianza y cual no?
- Restringir acceso en los cuales se tenga la información de mayor importancia.

- Así como también tener una estrategia de seguridad, y hasta donde se puede llegar sin dañar los derechos de los empleados pero sin dejar fuera la seguridad de nuestros datos y su confidencialidad.

Para mi criterio el siguiente diagrama ejemplifica como quedarían los recursos para comenzar una buena estrategia de seguridad esta seria mi propuesta.

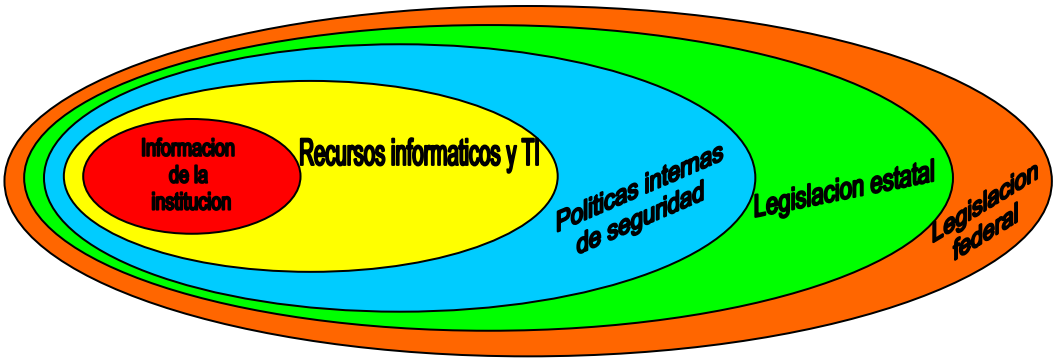


Fig. 4.2 Organización para proteger los recursos

4.4 Falta de la cultura de la seguridad.

El monitoreo de la actividad de los empleados, aún cuando se convenga en un contrato, puede considerarse invasión de la privacidad. Aquí entra en juego un concepto denominado cultura de seguridad.
Antes de imponer sanciones por conductas no adecuadas dentro de las instalaciones o el uso indebido de recursos de la empresa, es necesario aseverarse de que todos los empleados conozcan el manual de procedimientos de seguridad de la empresa para iniciar, es necesario que exista el manual.
No podemos hablar de una cultura de seguridad si no existe conciencia entre nuestros empleados. Ellos deben de saber que se les permite y que se le prohíbe, y que actividades consideradas como dañinas para los interés de la compañía y cuales son las sanciones a que se hacen merecedores en caso de no cumplir con los liniamientos.
También los empleados deben conocer sus derechos y responsabilidades en lo que respecta a la seguridad empresarial. Por poner un ejemplo, si los empleados saben que las actualizaciones de su antivirus se realizan de forma automática desde el servidor de la compañía, no podrán ser engañados cuando un individuo se acerque a ellos diciendo que es del departamento de soporte técnico y que necesita utilizar su computadora un momento. Y aquí evocamos lo que explicaba un hacker en ocasiones es mas fácil hacerse pasar por alguien de sistemas o soporte y robar la información.
Por igual modo, si los empleados conocen que están sujetos a sanciones por la divulgación de información confidencial de la empresa, y datos susceptibles como contraseñas, serán mas escrupulosos con la información que proporcionen a los extraños o sus compañeros de trabajo, al menos deben

saber que antes de proporcionar cualquier dato deben verificar la identidad del demandante.
Además de una falta de la cultura de la seguridad también se detecta que esta también se da por un factor muy importante y es por falta de cultura informática, en lo particular yo me he topado con gente que no sabe ni utilizar lo programas mas básicos de su computadora como Word, Excel, o simplemente el out look, creo que aquí las empresas deberían de tener una mayor cultura laboral y capacitar también a sus empleados para no comenzar con hoyos negros en cuanto la seguridad informática.

4.5 Alternativas ante esta problemática.

Desde los inicios de este trabajo, hemos propuesto como alternativa esta tesis como una ayuda ante la falta de legislaciones informáticas en nuestro país, desde un principio, propongo como una alternativa ante la falta de legislación en México, el sistema operativo Linux, con todas las herramientas que en capítulos anteriores se estudian y se proponen, pero para esta alternativa , yo comenzaría por los usuarios mismos, ya que mi filosofía es, un usuario bien informado es un usuario que no nos dará problemas, yo comenzaría a construir la siguiente alternativa así:

- Informar y educar a nuestros usuario en cuanto la cultura informática.
- Constantemente actualizar a nuestro personal ya que muchas veces no saben ni usar el office.
- Una buena infraestructura en nuestro sistema informático
- Un buen administrador de red, y como consejo no hay que dejarle toda la responsabilidad a un solo administrador de nuestra red.

Y por último mi propuesta de una alternativa contra la inseguridad informática:

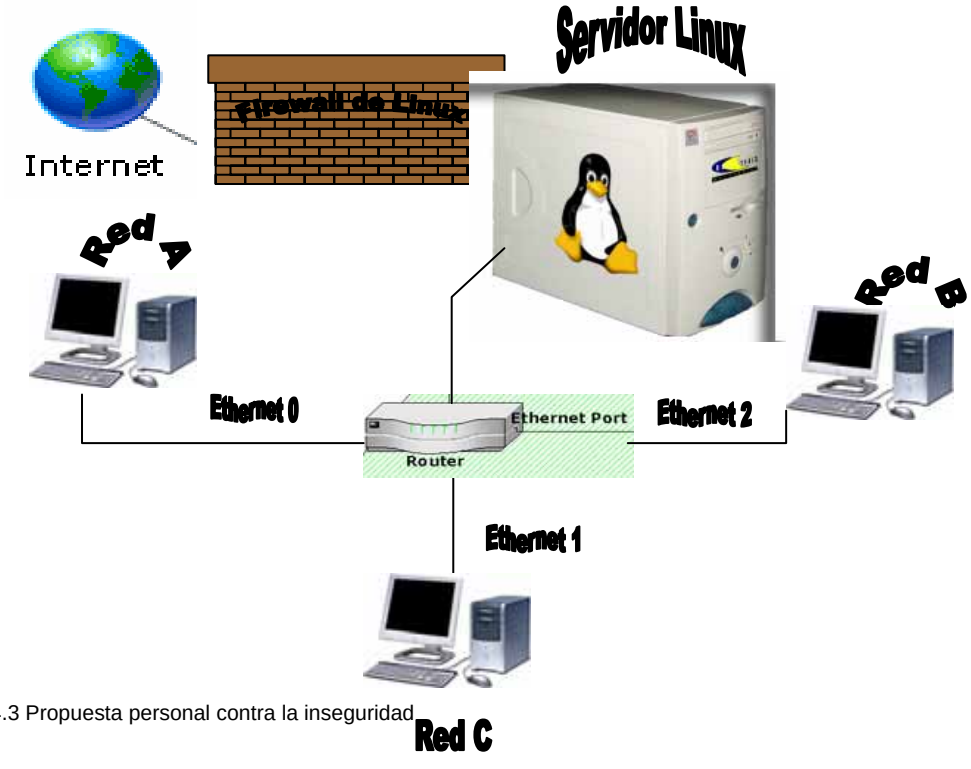


Fig.4.3 Propuesta personal contra la inseguridad

Características de la propuesta 1 (Para Pymes).

Red A	
eth0	10.0.0.1
Network	10.0.0.0
Netmask	255.0.0.0
Broadcast	10.255.255.255
Gateway	10.0.0.1
Nameserver	10.0.0.1
Hostname	Amaq20.linux,com
Alias	Usuario
Alias	anodo20 con dns
IP	=>20
Dominio	Linux.com

Red B	
Eth1	172.16.0.1
Network	172.16.0.0
Netmask	255.255.0.0
Broadcast	172.16.255.255
Gateway	172.16.0.1
Nameserver	172.16.0.1
Hostname	Bmaq20.curso,com
Alias	Usuario
Alias	bnodo20 con dns
Ip	=>20
Dominio	Curso.com

Red C	
Eth2	192.168.0.1
Network	192.168.0.0
Netmask	255.255.255.0
Broadcast	192.168.255.255
Gateway	192.168.0.1
Nameserver	102.168.0.1
Hostname	Cmaq20.Taller,com
Alias	Usuario
Alias	cnodo20 con dns
Ip	=>20
Dominio	Taller.com

4.6 Sugerencias.

La primera sugerencia ante la falta de una legislación informática en México es empezar por nosotros mismos, como personas responsables, ya que la seguridad empieza por nosotros mismos, la seguridad informática no es solo responsabilidad del administrador de la red, es responsabilidad de todos, desde el hecho de no compartir las claves de el sistema, nunca abrir mails de desconocidos, proteger adecuadamente mi información, reportar los incidentes al administrador de red o al encargado de sistema, destruir adecuadamente información importante que solo se usa una sola vez.

Buenas Prácticas.

No solo son en el ámbito informático el primer consejo es el siguiente.
Los empresarios deben de pedir asesoria con especialistas en derecho informático para corroborar que cumplen con la normatividad vigente en lo que se refiere a delitos informáticos, así como en materia de derecho laboral, ya que como se toco en renglones posteriores, la seguridad informática puede irrumpir en los derecho de nuestros empleados ya que es un hilo muy delgado que puede ser roto.
La empresa debe destinar recursos para que un responsable del área de sistemas se reúna con tres abogados: el de la compañía, un perito externo en derecho informático y un especialista en materia laboral, para verificar la

existencia de cada caso que se tenga, además redactar un manual de políticas de seguridad empresarial e informática.

Ahora tal vez nos preguntemos que recomiendo a hacer ante un caso de contingencia o un ilícito informático.

- Primeramente los que tienen que tomar cartas en el asunto es nuestra área administrativa, con el área de recursos humanos de la empresa.
- Informar el departamento legal de la empresa.
- En caso de ser actividades de tecnologías de información dar parte al área de sistemas.
- Reunir a los responsables de las áreas involucradas para tomar una acción conjunta que salva guarde los intereses de la empresa y no nos lleve a conflictos laborales.
- Considerar acciones a seguir en caso de que se requiera despedir al empleado o se proceda a una acción legal de carácter penal.

Por esto es importante asesorarnos con los especialistas que se mencionaron anteriormente.

En el ámbito empresarial tenemos tres elementos que conformarían una excelente estrategia de seguridad empresarial integral:

- Respaldo jurídico.
- Respaldo tecnológico.
- Respaldo de procesos.

Hay muchas empresas que creen que lo físico no importa pero están equivocados, ya que en la corta experiencia que llevo en el ambiente del outsourcing, eh notado que este aspecto las empresas lo dejan desprotegido, y poco a poco van tomando conciencia con soluciones como:

- Control de acceso del personal y de los visitantes a las instalaciones, vigilancia y sistema de monitoreo.
- Políticas de acceso a la red, permisos por jerarquías y grupos, herramientas de detección de actividades ilícitas en la red.

Por otro lado la colocación del site también es importante por lo general debe de estar monitoreado las 24 horas del día y también alejado de la mayoría de las personas, ya que en este se encuentra la parte medular de la empresa, también debe de estar restringido el acceso a este ya que aquí solo puede acceder el personal de sistema que debe ser de entera confianza. Y lamentablemente esto no ocurre en todas las empresas.

Y por último las practicas para los administradores de redes y el personal de sistemas.

El primero y como obligación es llevar una bitácora del sistema, me refiero a los servidores y a cada una de las maquinas del personal como en general del

sistema de computo en general. Este documento debe de estar conformado de cada una de las hojas como esta abajo:

Equipo	# de serie	Fecha	Nombre del usuario	Firma del responsable de depto.	Depto.	Estado del equipo	Solución al problema	Firma del area sistemas
Coment arios:								

Fig. 4.4 propuesta para bitácora

Si hubo un incidente de causa mayor en lo que se procedió legalmente también seria necesario anotarlo al pie de página en los comentarios y si es posible añadir una copia de la acta correspondiente que se levanto, claro esto en los que involucre tecnologías de la información.

Por otro lado también tenemos que ser concientes y hacer un calendario de las actividades que se realizara en los equipos y llevar un control apropiado de las aplicaciones que se han ejecutado en el equipo.

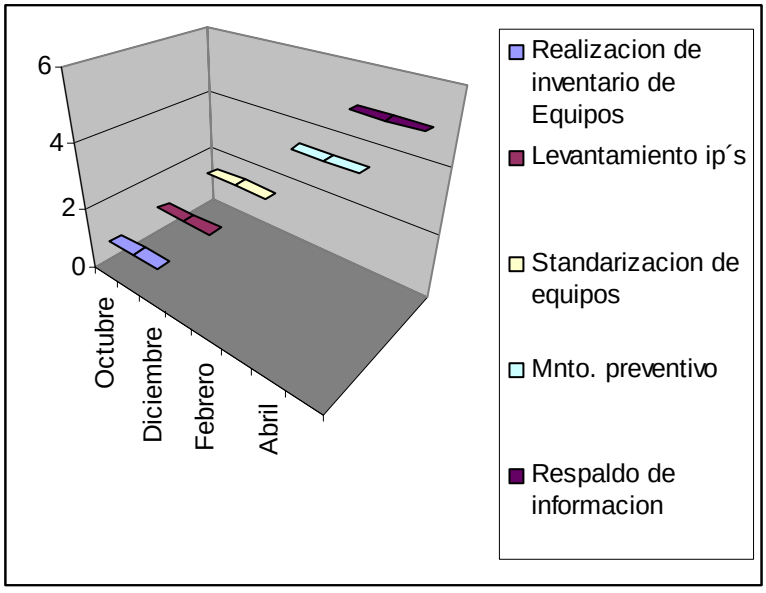


Fig. 4.5 propuesta de calendarización.

También es necesario crearles en el escritorio un acceso directo a un conexión a unidad de red, llamémoslo Share será un espacio compartido que estará destinado para que ellos mismos guarden sus respaldos, y información y que además solo ellos la puedan ver mediante claves o contraseñas en

dichas carpetas también nos servirá para que los usuarios intercambien información entre ellos ya que si es un archivo muy pesado será más fácil para ellos descargarlo de una unidad de red. Que desde otro dispositivo de almacenamiento.

Conclusiones

Es necesario que en nuestro país tengamos una legislación, pero ante la falta de esta es necesario, una debida implementación de políticas de seguridad y Linux en cualquiera de sus distribuciones es una buena herramienta, ante la falta de esta legislación, también hay que llevar acabo una buena política informática con el debido asesoramiento en términos laborales, ante la creciente demanda y crecimiento informático nuestro país se ha rezagado, también tenemos que crear una cultura informática, ya que no la hay, y tenemos que buscar mas alternativas además de las que la empresa Microsoft nos puede ofrecer, también en le presente trabajo podemos darnos cuenta que Linux, no se queda atrás ante esta necesidad, ya que nos ofrece desde una plataforma más segura, robusta y amigable, hasta una suite de contabilidad, también nos ofrece hojas de cálculo navegador en Internet y procesador de palabras entre otras aplicaciones, ya que Linux no solo es para PC de escritorio ya que tenemos versiones también para servidores y portátiles, y no solo queda ahí para los amantes de mac´s también es compatible Linux no tiene limites, ya que también ante la detección de intrusos Linux puede hacerles frente y ante la amenaza de virus tenemos un índice mas bajo y es menos vulnerable comparado con Linux, también si aún en la empresa no todas las pc´s quieren Linux , el mismo sistema puede ver en red a las pc´s con Windows mediante samba y también puede hacer un intercambio de archivos entre ellos, es necesario mencionar que empresas como Dell , HP e IBM ya ofrecen este sistema en sus servidores, y en algunos países de Latinoamérica este sistema ya se implementa, porque quedarnos atrás, en esta carrera informática, porque no pensar en Linux como una alternativa, puede ser difícil como todo cambio, pero como todo cambio requiere de esfuerzo y dedicación y un cambio siempre es para bien, ya que para una PYME que apenas comienza pagar licencias de Windows resulta un ojo de la cara y con Linux es diferente ya que su licencia es de tipo GNU y es un sistema abierto, en el que se puede moldear como la arcilla, y no por eso es mas vulnerable a ataques, ya que podemos crear nuestros propios firewall. Y adecuarlo a nuestras necesidades.

Apéndice A.
Comandos Básicos de Linux.

1. Teclas Útiles.

En las consolas de texto. <ctrl.><Alt.><F1-F6>. Cambia a una de las 6 consolas de texto.
<ctrl><alt><F7-F12>
cambia a una consola gráfica (si hay un clienteX arrancando en ella).
<ctrl><alt><supr>
Fuerza un reinicio del sistema.
<Shift><Repag>
Hace scroll hacia arriba para ver lo que se ha salido de pantalla.
<Shift><Avpag>
Hace scroll hacia abajo.

En la Shell (interprete de comandos).

<Ctrl><C>
Terminar. Envía el proceso una señal para que finalice.
<Ctrl><Z>
Suspender. Envía al proceso una señal para que se pare y quede en suspenso hasta que lo ordene que continué.
<Ctrl><D>
Finalizar, fin de fichero, logout.
<Ctrl><L>
Borra la pantalla.
[Flechas arriba y abajo]
Navega por los comandos introducidos anteriormente.
Si se esta en una sesión telnet, desde un terminal remoto que no soporte las teclas de los cursores, puede usar:
<Ctrl><P> (P de previous) para flecha abajo.
<Ctrl><N> (N de next) para flecha abajo.
<Ctrl><B> (B de back) para flecha izquierda.
<Ctrl><F> (F de forward) para flecha derecha.
<Ctrl><M> Intro.
<tab>
Completa de forma inteligente. Por ejemplo si estamos escribiendo un comando y escribimos sus primeras letras y pulsamos <tab> intenta completar buscando todos los comandos posibles, si hay varios que coincidan con esas letras nos los muestra todos, y podemos añadir alguna letra más y volver a pulsar <tab> si solo hay una coincidencia lo completa si no hay ninguna da un beep. Si ya hemos puesto el comando y estamos escribiendo una de las palabras que lo siguen en forma de parámetros se completa buscando ficheros. Si hemos comenzado la palabra con $ intenta completar con los nombres de las variables de entorno.
<Ctrl><A>
Ir a principio de línea.
<Ctrl><E>
Ir a fin de línea.
<Alt><L>
Convierte a minúsculas la palabra donde esta el cursor.
<Alt><U>
Convierte a mayúsculas la palabra.
<Alt><C>
Capitaliza la palabra (la primera letra en mayúsculas y el resto en minúsculas)
<Ctrl><T>
Transpone la letra donde esta el cursor por la anterior.
<Ctrl><K>
Corta desde el cursor hasta fin de línea.
<Alt><D>
Cortar desde el cursor hasta el final de la palabra.
<Ctrl><U>
Cortar desde el cursor hasta principio de línea.
<Alt><backspace>

Cortar desde el cursor hasta el principio de palabra.
<Ctrl><Y>
Pegar lo cortado.
<Ctrl><R>
Reserve Search (busca en los comandos anteriores lo que vamos tecleando).
En las X (entorno grafico).
<Ctrl><alt><backspace>
Matan a la fuerza el cliente X
<Ctrl><alt> [+] y < Ctrl><Alt.> [-]
(Con el + y el – del teclado numérico) Cambian la resolución de las X si hemos configurando las X si hemos configurado las X para trabajar con varias relaciones simultáneas.
<Ctrl>< Alt.> [flechas izqda o dcha.]
Van anterior o siguiente escritorio virtual.

2. comandos básicos.
man
Páginas del manual (es un help muy potente).
info
Más manuales.
pinfo
Combinan man e info en uno solo (no esta en todas las distribuciones).
Si no sabemos el comando exacto para man
ls
Listar directorio (como dir de msdos).
stat
Estado de un fichero.

file
Información sobre el tipo de fichero.
rm
Borrar un fichero (como del en msdos)
cp
Copiar un fichero (como copy en Msdos)

pwd
Te dice el directorio en el que estás.
cd
Cambiarse de directorio.
chown
Cambia el propietario de un fichero.
Chgrp
Cambia el grupo al que pertenece un fichero
Chmod
Cambia los permisos de un fichero.
Touch toca la hora de un fichero. Si no existe lo crea vació.
locate
Busca ficheros en el disco duro.
updatedb
Actualiza la base de los datos de ficheros del disco duro.
find
Busca ficheros (más avanzado que locate)
cat
Listar ficheros (como type en msdos).
head
Como cat pero solo las primeras líneas.
tail
Como cat pero solo las últimas líneas.
more
Permite ver un fichero con pausas.
less
Como more pero con la ventaja de que se pueda volver atrás.
split
partir ficheros.
grep
Buscar texto en ficheros.
cal
Saca el calendario en pantalla.
wc
Cuenta líneas, palabras o letras de uno o más ficheros.
expr

Evalúa expresiones (cálculos simples).

bc

Para hacer cálculos más complejos.

clear

Limpia la pantalla.

date

Saca fecha y hora actuales.

passwd

Cambia contraseña de un usuario.

vi

Editor de texto.

vim

Otro editor de texto al estilo vi.

emacs

Editor de texto.

joe

Editor de texto.

Xwpe

Editor de texto.

reset

Si enviamos ciertos caracteres de control a la terminal (a veces ocurre que al hacer un cat de un fichero binario) puede quedar desconfigurada y no vemos lo que tecleamos. En ese caso escribiremos este comando a ciegas y la terminal se restablecerá.

Comandos para comunicaciones, redes y multiusuario.

Who, w

Lista de usuarios conectados.

whoami

Información sobre el usuario actual ¿quién soy yo?

finger

Información sobre usuarios.

mail

Programa de correo muy simple.

write

Manda un mensaje a la pantalla de un usuario.

mesg

Bloqueo de mensajes de write.

wall

Mensaje a todos los usuarios

talk

Establecer una charla con otro usuario.

telnet

Se conecta a otra maquina.

Entorno grafico xwindow

Iniciar X.

StartX.

Abrir nuevas sesiones

Startx --:2, :3, :4, etc.

Configuración de XF86

Editar fichero

/etc/X11/XF86config

Configuración de servidor

X/etc/X11/ Xserver

Configurar X

Xconfigurator,xf86cfg,xf86config (en mandrake :Xfdrake).

Salir de las x por las malas

Ctrl-alt-backsapace

Comprimir, descomprimir, archivar...

Ficheros tar (tar empaqueta varios archivos en uno solo, pero no comprime).

Empaquetar

Tar cf archivo.tar ficheros

Desempaquetar

Tar –xvf archivo.tar

Ver contenido tar-tf archivo.tar

Ficheros g zip o bz2 (solo comprimen fichero a fichero no meten varios ficheros en uno)

gzip

Comprimir

gzip fichero

Descomprimir

gzip –d fichero.gz

bz2

Comprimir bzip fichero

Descomprimir bzip2 –d fichero.bz2

Para comprimir y archivar al estilo de los compresores zip hay que combinar el tar y el gzip o el bzip2 de la siguiente manera:

ficheros tar.gz

Comprimir tar –czf archivo.tar.gz ficheros.

Descomprimir tar –xvzf archivo tar.gz

Ver contenido tar –tzf archivo tar.gz

Ficheros tar.bz2

Comprimir tar-c ficheros | bzip2> archivo.tar.bz2

Descomprimir bzip2 –dc archivo.tar.bz2| tar-xv ver contenido bzip2 –dc archivo.tar.bz2| tar –t

Ficheros zip

Comprimir:zip archivo .zip ficheros

Descomprimir: unziparchivo.zip

Ver contenido: unzip –v archivo.zip

Los siguientes no vienen en todas las distribuciones:

Ficheros ha

Comprimir: lha a archivo.lha ficheros

Descomprimir: lha x archivo.lha

Ver contenido lha v archivo.lha o lha l archivo.lha

Ficheros arj

Comprimir: arj a archivo.arj ficheros

Descomprimir: arj x archivo.arj

Ver contenido: arj v archivo.arj o arj l archivo.arj

Ficheros zoo

Comprimir: zoo a archivo.zoo ficheros

Descomprimir: zoo x archivo. zoo

Ver contenido: zoo L archivo.zoo o zoo v archivo.zoo

Ficheros rar

Comprimir: rar a archivo.rar ficheros

Descomprimir: rar x archivo.rar

Ver contenido: rar l archivo.rar o rar v archivo.rar

Ficheros shar

Comprimir: shar

Descomprimir: ejecutarlo o usar unshar.

Manejo de las unidades de diskette y CD ROM.

En Linux para ver cualquier unidad de disco es necesario montarla primero, el hecho de montar una unidad sitúa virtualmente su contenido en un directorio de nuestro árbol, el directorio deberá existir y estar vació. Antes de expulsar una unidad (si es removible) hay que desmontarla.

Mount

Montar unidades, ejemplos:

Montar diskette mount –t msdos/dev/floppy/mnt

(dev/floppy=/dev/fd0)

montar cd-rom mount –t iso 9660/dev/cdrom/mnt (dev/cdrom=/dev/hdb)

Listar unidades montadas mount

umount

Desmontar unidades

super format

Formatea disquetes en modo ms-dos (hay que tener instalado fdutils)

mkfs.ext2

Formatea discos o disketes en formato ext2 de Linux

fsck

Comprueba un disco (como el scandisk de dos), hay quien aconseja imprimir la pagina man de fsck porque si te falla el sistema (no arranca) y necesitas repararlo podrás usar fsck pero no podrás consultar la ayuda.

Apéndice B.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991[89]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

[89] Tomada de http://www.gnu.org/copyleft/gpl.html

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER

EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.
Copyright (C) *yyyy name of author*

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) *year name of author*
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

BIBLIOGRAFÍA

1. **Autor**. Le Blanc Dee-Ann.
2. **Título del Libro**. *Administración de sistemas LINUX*
3. **Traductor** Estudio Rojas Benarroch.
4. **Edición**.
5. **Pie de Imprenta**. España. Ediciones Anaya Multimedia. Febrero 2001.

1. **Autor**. Anónimo
2. **Título del Libro.** *Edición especial Linux Máxima seguridad*
3. **Traductor** José Arroyo Vox pupolli, S.L.
4. **Edición**.
5. **Pie de Imprenta**. España. Prentice Hall. 2000

1. **Autor**. Villalón Huerta Antonio.
2. **Título del Libro**. *Seguridad en Unix y Redes*
3. **Traductor**
4. **Edición**. Versión 2.1
5. **Pie de Imprenta**. GNU Free documentation License, Julio 2002.

1. **Autor**. Téllez Julio.
2. **Título del Libro**. *Derecho informático*
3. **Traductor.**
4. **Edición**.3ª
5. **Pie de Imprenta**. España. MC Graw Hill. 2004.

1. **Autores**. Michael A. Gallo, William M. Hancock .
2. **Título del Libro**. Comunicación entre Computadora y Tecnologías en Redes
3. **Traductor.** Michael A. Gallo
4. **Edición**.
5. **Pie de Imprenta**. Thomson. 2002.

1. **Autores**. Cassel Lilian N.
2. **Título del Libro**. Computer Networks and open systems: an aplication development perpesctive
3. **Traductor. Richard H Austing**
4. **Edición**.
5. **Pie de Imprenta**. Jones and Barttelt. 2000.

FUENTES URL

1. http://www.linux-es.org
2. http://www.slax.org
3. http://www.gentoo.org
4. http://www.centos.org
5. http://www.knopper.net/knoppix/index-en.html
6. http://www.suse.com/
7. http://www.gnoppix.org
8. http://www.ubuntu.com/
9. http://www.gnu.org/copyleft/gpl.html

Nota: Todas las imagines de tux Son propiedad intelectual de cada uno de los creadores y de igual manera las graficas y figuras de apoyo.