El sistema operativo GNU/Linux y sus herramientas libres en el mundo de la seguridad: estudio del estado del arte.

Jorge Ferrer
Mundo GNU

jferrer@acm.org

Javier Fernández-Sanguino
Germinus Solutions

jfs@computer.org

En este documento se estudiarán las distintas capacidades de GNU/Linux y las distintas herramientas de software libre de las que dispone en entornos de seguridad. Se hará un énfasis especial en el uso de en el ámbito elementos conectados a una red global, como Internet. Asimismo, se comparará el estado del arte de dicho software con algunos de sus competidores basados en software propietario.

1. Introducción

1.1. Objetivos de este documento

En este documento se intentará presentar el estado en el que se encuentra el sistema operativo GNU/Linux y las herramientas libres por él proporcionadas en el mundo de la seguridad de sistemas informáticos. Se ofrecerá una panorámica general del mundo de la seguridad y de las ventajas y desventajas que, en este entorno, ofrece el software libre.

El objetivo es que la información presentada pueda ser leída y comprendida por personas sin formación previa en lo que a seguridad se refiere. Por ello se ha decidido estructurarla en una serie de secciones asociadas a casos de uso típicos, en especial cuando se tratan de redes de comunicaciones como, Internet.

1.2. Escenarios considerados

La gran mayoría de los documentos relativos a seguridad siguen un enfoque, que se podría considerar tradicional, basado en los objetivos básicos de un entorno de seguridad. Es decir:

- Confidencialidad de los datos: almacenados, procesados y transmitidos.
- Integridad de los datos.
- Asegurar identidad de origen y destino.
- Disponibilidad de datos o servicios.

Aunque su validez metodológica sea mucho mayor, puede ser difícil entender el enfoque sin entender los principios en los que se basa, algo no exigible a personas no iniciadas en las materias de seguridad en las tecnologías de la información. El enfoque adoptado en este documento, sin embargo, parte de las distintas tareas que podrá ejercer el sistema operativo GNU/Linux, haciendo énfasis en los peligros asociados a un estado de máxima interconexión, es decir, las redes locales e Internet. Desde este punto de vista, se pueden distinguir los siguientes escenarios:

- La seguridad de los clientes (sistemas finales) que hacen uso de servicios en redes hostiles, haciendo un énfasis especial en la red Internet.

- La seguridad en la oferta de acceso a redes públicas, desde la perspectiva de un proveedor de servicios de comunicación que interconecta a usuarios finales con dichas redes.
- La seguridad en la publicación de información y acceso a servicios. Es decir, la perspectiva de servidor, centrando la disquisición, de nuevo, en servicios habituales de Internet.
- La seguridad (cifrado) en el envío, recepción y almacenamiento de información. Con un hincapié especial en los servicios de mensajería.
- Las garantías de la autenticidad e integridad de información transmitida. Aplicado tanto a los servicios de mensajería analizados previamente entre otros.

De esta forma, se trata el uso del sistema operativo en todos los agentes de una comunicación en Internet, desde el cliente hasta el servidor pasando por los encaminadores y sistemas de almacenamiento intermedios. Cada una de estas necesidades será tratada en una sección del documento. De igual forma, para cada una de ellas se intentará realizar una breve introducción en la que se presentan las principales amenazas a los que se pueda enfrentar un sistema si se hace uso de éste en dichas circunstancias. Debido a las evidentes limitaciones en un trabajo de este tipo, no es posible realizar un tratamiento exhaustivo de cada uno de los temas.

1.3. Panorámica general de la seguridad

Habitualmente los usuarios finales no tienen en consideración la seguridad cuando hacen uso de un sistema, ya que, frecuentemente, se ignoran los aspectos relacionados con la seguridad. De igual forma, estos aspectos a veces pueden considerarse una molestia, ya que la seguridad suele ir en el platillo opuesto de la comodidad y facilidad de uso en la balanza del diseño de un sistema. Es por esto que los usuarios a veces puedan tener una imagen negativa de la seguridad, por considerarlo algo molesto y que interrumpe su capacidad de realización de un trabajo determinado. En un entorno seguro, un usuario se encuentra con tareas que le pueden resultar incómodas (como por ejemplo, recordar contraseñas, cambiarlas periódicamente, ...) y que pueden limitar las operaciones que puede realizar así como los recursos a los que se le permite acceder.

Sin embargo, la seguridad es fundamental a la hora de afrontar tareas que se realizan en sistemas informáticos ya que son las únicas medidas que pueden garantizar que éstas se realicen con una serie de garantías que se dan por sentado en el mundo físico. Por ejemplo, cuando se guardan cosas en una caja fuerte en un banco real, no se piensa que cualquier persona del mundo puede llegar a ésta como si de una forma inmediata, en lugar de un banco, se tratara una estación de autobuses. En el mundo intangible de la

informática, tan cerca de un servidor están sus usuarios legítimos como los usuarios que hacen uso de la misma red de comunicaciones. Es más, estos usuarios, en el caso de una red global, se cuentan por millones. Algunos serán "buenos vecinos" pero otros serán agentes hostiles.

1.4. ¿Porqué son necesarios los mecanismos de seguridad?

Para poner de relevancia lo comentado en los párrafos anteriores se han elegido tres casos genéricos que se describen a continuación. Con ellos se pretende mostrar alguno de los peligros, relativos a seguridad, de estar 'interconectados'. Para cada uno de ellos existen mecanismos de seguridad que permiten llevar a cabo las operaciones de manera satisfactoria.

1.4.1. Intercambio de información

Cuando se intercambia información con un ordenador remoto, esa información circula por una serie de sistemas intermedios que son desconocidos a priori (excepto en ámbitos muy específicos). Además, no sólo no se sabe cuales serán estos sistemas intermedios, sino que además no se dispone de ningún control sobre ellos o sobre lo que puedan hacer con nuestros datos al pasar por ellos. Quizá el propietario original es de fiar pero su sistema ha sido comprometido por un atacante que toma posesión de los datos enviados.

Por otro lado tampoco se puede estar seguro de que el sistema al que uno se está conectando es quien dice ser. Existen diversos medios técnicos para suplantar la identidad de un sistema y engañar a un tercero cuando realiza la conexión.

En definitiva, no existe una certeza absoluta de que aquellos sistemas a los que uno envíe información sean realmente los auténticos; además, en el caso de que lo sean no se sabe si les llegará la información que se les envía, o si llegará sin cambios o si, aún si llega sin modificaciones, será leída por terceras partes.

Cuando este tipo de garantías sean necesarias no quedará más remedio que aplicar las técnicas que se tratarán más adelante.

1.4.2. Instalación de software dañino involuntariamente

Otra posibilidad que no se debe descartar es que se instale software en un ordenador sin conocimiento del usuario o administrador. Esto puede ocurrir de muchas formas,

algunas relacionadas con operaciones que se realizan todos los días. Algunos ejemplos son:

- Virus por correo electrónico. Estos se activan ya sea porque el lector de correo lo ejecuta automáticamente sin preguntar al usuario o porque lo hace este involuntariamente creyendo que se trata de otra cosa.
- Virus o troyanos asociados a ficheros descargados de servidores, en principio, confiables, que el usuario se descarga para su instalación y ejecución en su propio ordenador.
- Explotación de una vulnerabilidad de un servicio que se está ofreciendo a través de Internet. Como por ejemplo un servidor web. Un caso similar sería una carpeta compartida donde otros miembros de la red local (y quizá un virus que haya en sus ordenadores) pueden copiar archivos.

Este software dañino no sólo puede obtener o borrar información del sistema en el que se instala, también puede servir como plataforma de ataque a otros ordenadores.

Es por esto que todo ordenador, máxime cuando se encuentra expuesto a recibir información del exterior, debe protegerse con las medidas de seguridad adecuadas aunque se considere que no tiene información ni servicios de gran importancia.

1.4.3. Protección ante accesos no autorizados

Cuando se ofrecen servicios o información en una red para sus usuarios legítimos, al mismo tiempo se abre la puerta a posibles intrusos en estos sistemas. Protegerse de esta posibilidad implica tener un especial cuidado con todo el software empleado, desde el sistema operativo hasta la última de las aplicaciones instalada, y cuidar en gran medida su configuración.

Pero tampoco debería olvidarse la posibilidad de que existan intrusos que accedan físicamente al sistema. La evolución de las comunicaciones ha hecho que se preste una gran atención a la posibilidad de accesos remotos, pero de nada sirve evitar esta posibilidad si se permite el acceso físico al sistema a personas no autorizadas. Es por esto que, en algunos casos pueda ser necesario tomar las medidas de seguridad adecuadas sobre el propio hardware para evitar robos, o pérdidas de información por estos accesos inadecuados.

En definitiva un buen sistema de seguridad debe proteger los sistemas vulnerables ante el posible acceso físico o remoto de intrusos no autorizados. Evidentemente, el nivel de seguridad establecido tendrá que ser consecuente con un análisis previo de los riesgos,

considerando el impacto de dicho acceso no deseado contra las posibilidades de que este se produzca.

Algunas medidas de seguridad que se pueden implantar en estos casos van desde el cifrado de información sensible para impedir su acceso sin la clave adecuada, métodos físicos de destrucción de la información en caso de manipulación mecánica de la misma, etc.

1.5. Organización de este documento

Como se ha comentado este documento se organiza entorno a escenarios de uso de un entorno de red y que han sido ya presentados. Pero antes de empezar con ellos se ha considerado conveniente precederlos de dos apartados:

1. El software libre en el mundo de la seguridad: presenta el concepto software libre y trata de forma genérica sus ventajas e inconvenientes generales en el mundo de la seguridad.
2. El sistema operativo: examina los principales sistemas operativos que existen desde el punto de vista de seguridad y entre ellos GNU/Linux que será el centro de atención del resto del documento.

Estos apartados son seguidos de un estudio de cada uno de los escenarios y unas conclusiones finales. En ellas se valora el estado del arte de GNU/Linux y sus herramientas del software libre en base a la información presentada.

2. El Software Libre en el mundo de la seguridad

2.1. Definición de Software Libre

El concepto de software libre es, en primera instancia, fácil de presentar, aún no existiendo una única descripción reconocida por todos de lo que es realmente este tipo de software. En general se entiende como software libre aquel programa o conjunto de ellos de los que el usuario puede disponer del código fuente, sin restricciones, y el cual puede modificar y redistribuir también sin restricciones. Estas libertades garantizadas al

usuario del software (o a aquel que lo recibe) no son contrarias a los derechos legítimos del autor del programa, es decir, el autor del programa no pierde todos sus derechos sobre el mismo. No se incluye, por tanto, en esta definición software en el "domino público".

Una descripción más completa de lo que podría considerarse software libre, es la dada por las Directrices de Software Libre de Debian (http://www.debian.org/social_contract#guidelines), que constituyen la base de la definición de *Open Source* (Open Source Definition, www.opensource.org), aunque existen entre ellas ciertas diferencias. Entre las licencias más utilizadas para este tipo de software cabe destacar la licencia GNU GPL (http://www.gnu.org/copyleft/gpl.html) y la licencia BSD (http://www.debian.org/misc/bsd.license).

2.2. Fallos de seguridad en la utilización del software

Se puede hacer un análisis agrupando los fallos de seguridad que se pueden dar en el software. Este análisis va a permitir enfocar, más adelante cómo distintos tipos de software ayudan a solventarlos. De una forma simplista, se pueden dividir en tres bloques:

- fallos debidos a errores desconocidos en el software, o conocidos sólo por terceras entidades hostiles.
- fallos debidos a errores conocidos pero no arreglados en la copia en uso del software.
- fallos debidos a una mala configuración del software, que introduce vulnerabilidades en el sistema

El primero de ellos se puede achacar a la calidad del código, el segundo a la capacidad y celeridad de arreglo de los errores descubiertos en el código por parte del proveedor del mismo y a la capacidad del administrador de recibir e instalar nuevas copias de este software actualizado. El tercer tipo de vulnerabilidades puede achacars, sin embargo, a una falta de documentación del software o una falta de formación adecuada de los administradores para hacer una adaptación correcta del mismo a sus necesidades.

Los fallos pueden dar lugar a un mal funcionamiento del programa, siendo en el ámbito de la seguridad preocupantes por cuanto:

- pueden implementarse algoritmos de forma incorrecta lo que puede llevar a una pérdida de seguridad (por ejemplo, un algoritmo de generación de claves que no se base en números totalmente aleatorios)

- pueden diseñarse servicios que, en contra de sus especificaciones, ofrezcan funcionalidades no deseadas o que puedan vulnerar la seguridad del servidor que los ofrezca.
- pueden no haberse tomado las medidas de precaución adecuadas para asegurar el correcto tratamiento de los parámetros de entrada, lo que puede hacer que un atacante externo abuse de ellos para obligar al programa a realizar operaciones indeseadas.

2.3. Ventajas del Software Libre en el mundo de la seguridad

Si se analiza la descripción realizada previamente de la definición de software libre se derivan una serie de ventajas principales de este tipo de software sobre el software propietario, algunas de las cuales son muy adecuadas para el mundo de la seguridad. A saber:

- Al disponer del código fuente de los programas en su completitud, éste puede ser analizado por terceras personas ajenas a sus autores en busca de fallos de diseño o de implementación. Es decir, cualquiera con los conocimientos necesarios puede realizar una auditoría de dicho código.
- La posibilidad de realizar modificaciones libremente al código fuente y distribuirlas permite que cualquiera pueda ofrecer mejoras sobre éste. Estas mejoras podrán ser nuevas funcionalidades que se incorporen al mismo o parches que corrijan problemas detectados anteriormente.
- Las características del software libre hacen que no sea lógico cargar costes sobre el software en sí (dado que se ha de distribuir sin cargo), lo que permite que este tipo de software pueda ser utilizado por organizaciones y personas con menos recursos económicos. Esto se presenta como una ventaja cuando se compara con los precios de lo que cuesta el software de seguridad propietario hoy en día (licencias de cortafuegos, vpns, sistemas de detección de intrusos, etc.). El software libre pone en manos de cualquiera el tipo de tecnología que, hoy por hoy, sólo podían tener grandes corporaciones.
- De igual forma, la posibilidad de modificar libremente el software permite a las organizaciones que lo adapten a sus propias necesidades, pudiendo eliminar funcionalidades que no le sean de interés. En el mundo de la seguridad existe la máxima de “lo más sencillo es más seguro” por ello poder eliminar funciones innecesarias de las herramientas las puede convertir de forma inmediata en más seguras (porque no podrán ser utilizadas estas funcionalidades para subvertirlas).

Frente al análisis de fallos realizado anteriormente, el software libre protege a sus usuarios de una serie de formas determinadas. Entre estas:

- La posibilidad de una auditoría de código en las herramientas software reduce los riesgos de seguridad debido a la aparición de fallos desconocidos, a la introducción de funcionalidades no deseadas en el código o la incorrecta implementación de algoritmos públicos. Aunque no se pueda asegurar que el código esté carente de errores, si es posible garantizar que tantas posibilidades tiene de encontrar un fallo de programación en éste (que lleve implícito un riesgo de seguridad) un atacante externo como la organización lo utilice. Si bien no se puede asegurar que los mejores cerebros del mundo realicen la auditoría de código del software que una compañía utiliza, dicha compañía si tiene la posibilidad, en función de sus necesidades respecto a la seguridad, de realizar ella misma dicha auditoría de código o pagar a alguien para que la realice. Muchos de los proyectos de software libre, entre ellos el núcleo de Linux, el proyecto Apache, y la distribución OpenBSD realizan auditorías del código para asegurar su integridad, seguridad y ajuste a las especificaciones de funcionalidades requeridas.
- La posibilidad de corregir los programas y distribuir dichas correcciones permite que los programas evolucionen de una forma más abierta. En el mundo de la seguridad, un fallo en el sistema significa exponer a éste a una “ventana de vulnerabilidad” que tiene lugar desde la detección del fallo (por parte de sus usuarios legítimos o de terceras partes, hostiles incluso) a la aplicación de la medida correctiva, que pueda ser la instalación del parche adecuado que arregle el problema, pasando por la *generación* de dicho parche. El hecho de que la generación de dicho parche pueda realizarse por un número de personas (confiables) elevado, y no por un sólo fabricante, debe, en teoría, reducir este tiempo de exposición a dicha vulnerabilidad.
- El hecho de que exista una cierta independencia entre el software y su fabricante, o distribuidor original, permite que los usuarios de este software, en caso de pérdida de soporte, puedan realizar el mantenimiento de éste ellos mismos o subcontratarlo a una tercera empresa. Este hecho es, si cabe, de gran importancia en el mundo de la seguridad dado que la seguridad de una entidad no debe depender de la solvencia de terceras compañías a las que adquiere productos de seguridad y actualmente, sin embargo, es así. Debido a la gran variabilidad de riesgos potenciales contra los que un elemento de seguridad informática debe proteger, estos productos han de ser frecuentemente actualizados, muchas veces empujados por el descubrimiento de ataques antes desconocidos. Sin embargo, si una compañía depende de un producto de una tercera entidad y, de forma transitiva, de esta tercera entidad, la pérdida de soporte de este producto (por quiebra de la tercera entidad o abandono de una determinada línea de negocio) da lugar a que la compañía no esté adecuadamente

asegurada contra los nuevos riesgos que puedan surgir. Las únicas opciones posibles serán mantener un sistema de seguridad que, con el tiempo, quedará obsoleto, o migrar a un sistema de seguridad nuevo (otro producto de otro fabricante) con sus consecuencias económicas y de impacto en servicios ya consolidados.

2.4. Desventajas del software propietario

Las mismas garantías que ofrece el software libre son problemas que se le pueden achacar al software propietario. En este documento se entenderá como software propietario aquél que se distribuye en forma de binarios, sin código fuente, por parte de una compañía que licencia dicho software para un uso concreto, con un coste determinado. No se van a realizar comparativas con la nebulosa intermedia de distintos tipos de software cuyas licencias se sitúan entre ambos extremos, por ejemplo: software que se distribuye el código fuente pero no se puede modificar, software que se distribuye con limitaciones para su uso comercial, etc.

Se puede hablar de las siguientes desventajas del software propietario para el usuario final:

- Posibilidad de que existan funcionalidades no deseadas en dicho software. Dependiendo de la programación realizada, algunas funcionalidades podrán ser activadas o desactivadas por el usuario, pero pueden existir también funcionalidades que no se puedan desactivar o que, incluso, no se encuentren documentadas. Llevándolo al extremo se podría hablar de “puertas traseras” abiertas por el fabricante del software que, después de todo, es un agente comercial y, por tanto, tiene sus propios intereses que pueden ser contrarios a los de la compañía que instala un software de seguridad específico.
- Desconocimiento del código por parte del usuario. Esto puede llevar a que el fabricante pueda llegar a tener una falsa sensación de seguridad por oscuridad, es decir, las vulnerabilidades de su producto no tienen por qué ser conocidas porque nadie tiene acceso a las “tripas” del mismo. De igual forma, esto puede llevar a que el fabricante no tenga interés en desarrollar el código de una forma adecuada porque, al fin y al cabo, el usuario no va a ver dicho código ni evaluar la calidad de su implementación.
- Necesidad de confiar totalmente en el fabricante. Esto es así por cuanto éste ha implementado los algoritmos de seguridad y el usuario no puede garantizar por sí mismo que su implementación ha sido correcta y que, por ejemplo, las propiedades matemáticas necesarias para que estos algoritmos funcionen correctamente se cumplen en todas las condiciones.

- Dependencia de una tercera entidad, ya que es el fabricante del producto el único que puede ofrecer nuevas versiones de éste en caso de fallo o incluir nuevas funcionalidades que puedan ser necesarias. Esto es una desventaja debido a que el usuario no puede transferir esta dependencia a otra entidad, en caso de que el fabricante original haya traicionado su confianza (demasiados errores en la implementación, demasiado tiempo en la generación de parches para arreglar problemas graves, etc..)

Cabe hacer nota que, algunos fabricantes de software, observando las ventajas del modelo *Open Source* ofrecen, con restricciones o sin ellas, copias del código fuente a terceras entidades interesadas. Tal es el caso, por ejemplo, de fabricantes de sistemas operativos como Sun Microsystems y Microsoft y de fabricantes de productos de seguridad como PGP (hasta febrero de 2001 con su suite de aplicaciones basadas en cifrado asimétrico) y NAI (con su cortafuegos Gauntlet).

2.5. Desventajas del software libre

Sin embargo, el uso de software libre no está exento de desventajas. Así se podrían enumerar las siguientes:

- la posibilidad de una generación más fácil de troyanos, dado que el código fuente también puede ser modificado con intenciones maliciosas. Si el troyano logra confundirse con la versión original puede haber problemas graves. La fuente del programa, en realidad, será el método de distribución de software, que, de no ser seguro, permitirá que un tercer agente lo manipule. La distribución de software se asegura añadiendo posibilidad de firmado de hashes de la información distribuida
- el método de generación de software libre suele seguir, en la mayoría de los casos, el modelo *bazar*, es decir, muchas personas trabajan sobre partes concretas e integrando sus cambios o personas desde el exterior contribuyen mejoras al proyecto global. Esto puede dar lugar a que se realice una mala gestión del código fuente del software por no seguir métodos formales de seguimiento, la consecuencia final es que falten piezas clave (que nadie ha contribuido) como es el caso de la documentación.
- Al no tener un respaldo directo, la evolución futura de los componentes software no está asegurada o se hace demasiado despacio.

En mayor o menor medida, algunas de estas desventajas pueden tener solución. Por ejemplo, la difusión de troyanos se limita mediatne el uso de técnicas de firma digital para garantizar la inviolabilidad del código o binarios transmitidos. De igual forma, los problemas de evolución futura parecen quedar resueltos con un cambio de paradigma

por parte de las compalías de software. Es el cambio de un modelo de negocio de cobro por productos a cobro por servicios. Ya se observan, en el mundo de software libre, compañías que contratan a personal cualificado para hacer mejoras sobre proyectos libres para cubrir sus intereses pero haciendo públicas las modificaciones realizadas.

3. El sistema operativo

El sistema operativo está formado por el software que permite acceder y realizar las operaciones básicas en un ordenador personal o sistema informático en general. Los sistemas operativos más conocidos son: AIX (de IBM), GNU/Linux, HP-UX (de HP), MacOS (Macintosh), Solaris (de SUN Microsystems), las distintas variantes del UNIX de BSD (FreeBSD, OpenBSD..), y Windows en sus distintas variantes (de la empresa Microsoft). En lo que a seguridad se refiere, un sistema operativo puede caracterizarse por:

Consideración de la seguridad en el diseño

Hay sistemas operativos que han sido creados con la seguridad como objetivo fundamental de diseño. Estos serán de entrada más seguros que los demás. En otros sistemas operativos aunque no fuera el objetivo fundamental sí ha podido ser un parámetro importante y por último en otros no se ha considerado más que a posteriori. Es de esperar que sean éstos últimos los que más problemas de seguridad tienen.

Capacidades de comunicación y configuración de esta

Los sistemas operativos modernos ofrecen grandes capacidades de comunicación. Desde el punto de vista de la seguridad estas capacidades pueden convertirse en puntos de acceso a posibles atacantes y será necesario protegerlos. El sistema operativo deberá proveer de los mecanismos y/o herramientas necesarias para llevar a cabo esta tarea de forma suficientemente fiable. Esto incluye ofrecer la capacidad de cerrar toda vía de comunicación que no se use y limitar la que sí se emplee a los casos y usuarios que realmente se deseen permitir.

Capacidades de auditoría

Estas capacidades son las que van a permitir determinar qué elementos acceden a qué partes del sistema en sus distintos niveles (ficheros, dispositivos, elementos de comunicación), etc.

Herramientas disponibles

Dado que en general las aplicaciones no son portables entre sistemas operativos de distintos fabricantes (exceptuando algunos casos entre las distintas variantes de UNIX) otro elemento a considerar en la seguridad de cada sistema operativo se caracteriza por la cantidad y calidad de herramientas de seguridad que tiene disponibles.

3.1. Sistemas monousuario y multiusuario

En algunos sistemas operativos se accede al sistema por medio de un usuario único que tiene permiso para realizar cualquier operación. Este es el caso de los sistemas operativos más antiguos como MS-DOS y algunos más recientes como la serie Windows 95/98/Me de Microsoft o MacOS (antes de MacOS X) de Macintosh. En estos sistemas no existe una diferenciación clara entre las tareas que realiza un administrador del sistema y las tareas que realizan los usuarios habituales, no disponiendo del concepto de multiusuario, un usuario común tiene acceso a todas las capacidades del sistema, pudiendo borrar, incluso, información vital para su funcionamiento. Un usuario malicioso (remoto o no) que obtenga acceso al sistema podrá realizar todo lo que desee por no existir dichas limitaciones.

Otros sistemas operativos, sin embargo, han estado siempre preparados para soportar sistemas multiusuario, permitiendo agruparlos y asignar distintos privilegios a cada uno de ellos o a sus grupos. Este es el caso de todos los sistemas UNIX y de los sistemas Windows NT/2000. Esta característica es enormemente útil desde el punto de vista de seguridad. Por ejemplo en el caso de que un usuario se vea afectado por un virus, una intrusión, etc. el resto de los usuarios (si los hay) y, sobre todo, el sistema no tendrán por qué verse afectados a menos que vulnerabilidades en éstas puedan ser utilizadas por un atacante para elevar sus privilegios.

Cabe notar que los sistemas operativos libres (Linux y BSD) no soportan una asignación de grupos y usuarios tan versátil como NT y 2000. Los grupos en UNIX son mucho menos versátiles (y más difíciles de administrar) que aquellos aunque también más conocidos.

Queda claro que que en todo ordenador donde la seguridad es un factor que se considera importante debe optarse por un sistema operativo que soporte varios usuarios con distintos privilegios.

3.2. Elección del sistema operativo

Generalmente los motivos por los que un individuo elije un sistema operativo u otro tienen más que ver con la facilidad de uso o la gama de aplicaciones disponibles que con la seguridad. Sin embargo, la elección del sistema operativo determinará en gran medida la garantía de seguridad que se podrá conseguir al acceder a servicios ofrecidos a través de Internet. Por tanto, cuando se necesite esta garantía deben conocerse bien las características en lo que a seguridad se refiere tienen los distintos sistemas operativos. Introducir estas características será el objetivo de este apartado.

Muchas veces se ha dicho que GNU/Linux es un sistema muy seguro (especialmente en comparación con Windows), pero esta afirmación hay que matizarla: la verdad exacta es que *GNU/Linux tiene el potencial para convertirse en enormemente seguro*. Pero de entrada no tiene porque serlo. En particular debemos ser conscientes de que se trata de un sistema operativo pensado para entornos de red y por ello tiene grandes capacidades de conexión con otros ordenadores y de ofrecerles servicios. En la mayoría de distribuciones tras realizar la instalación inicial se dejan activos una serie de servicios como puede ser un servidor web, un servidor de ficheros (FTP), servicios de bajo nivel (hora, caracteres aleatorios, etc). Desde el punto de vista de seguridad conviene desactivar todos los servicios que no se deseen ofrecer para reducir el número de oportunidades que se le dan a un posible atacante para encontrar una vulnerabilidad. Este sería el primer paso para configurar el sistema para que sea más seguro. Sólo mediante una configuración cuidadosa y exhaustiva del sistema operativo y sus aplicaciones se podrá decir que el ordenador es seguro. Y GNU/Linux ofrece las herramientas y la capacidad de configuración necesaria para hacer esto posible.

Otra opción, también libre, son los sistemas operativos de la serie BSD: NetBSD, OpenBSD y FreeBSD. Todos ellos son de tipo UNIX y al igual que GNU/Linux tienen una gran capacidad de configuración, lo que permite prepararlas para que sean enormemente seguros. Cabe destacar el proyecto OpenBSD, que tiene como principal objetivo construir un sistema operativo tan seguro como sea posible. Para ello auditan el código tanto del núcleo como de las herramientas y aplicaciones básicas del sistema. Si se necesita un sistema muy seguro esta es una gran opción. Simplemente debe tenerse en cuenta que habitualmente son las aplicaciones básicas las que son auditadas de forma habitual.

En general los sistemas operativos propietarios están en igualdad de condiciones frente a la seguridad que los sistemas operativos libres que acaban de presentarse. Una ventaja que sí han tenido hasta ahora es que el propio fabricante daba soporte técnico y garantías de seguridad de sus sistemas (especialmente en los UNIX propietarios). De igual forma, otra ventaja adicional ofrecida por estos sistemas operativos, inexistente actualmente en GNU/Linux, es que los fabricantes persiguen de forma activa la

certificación del sistema operativo (o una parte de éste) frente a los estándares de seguridad del mercado. Por ejemplo, Solaris (de Sun) cuenta con una versión reducida (Trusted Solaris 7) que ha sido certificada B1 y que cumple el Common Criteria (CC, http://www.commoncriteria.org/ (http://www.commoncriteria.org/)) al nivel EAL4, AIX de IBM también tiene una versión que cumple EAL4 del CC.

Pero ahora también están apareciendo un número interesante de empresas de soporte que ofrecen este mismo servicio para todo tipo de software libre. La diferencia es que no hay una única opción, sino tantas como el mercado permita. Y con ello se está descubriendo también la ventaja que supone no estar atado a un proveedor del servicio. No sólo por la posibilidad de cambiar sino porque esta posibilidad obliga a las empresas que dan el servicio a esforzarse en hacerlo lo mejor posible.

Por completitud se tratarán a continuación ligeramente las características generales, en lo que a seguridad se refiere, de los principales sistemas operativos propietarios.

La serie de sistemas operativos Windows 95/98/Me no fue diseñada inicialmente para entornos de red como Internet. En particular no se tuvieron en cuenta aspectos fundamentales relacionados con seguridad. Por ello es conocido como uno de los sistemas menos seguros y con más vulnerabilidades. Por un lado debido a que es un sistema operativo limitado en cuanto a la capacidad para ofrecer servicios puede pensarse que se ofrecen menos puntos de ataque. Sin embargo esta ventaja es ficticia, ya que el motivo real de la no existencia de estos servicios no es la seguridad.

Para solventar este problema, Microsoft lanzó Windows NT/2000 rediseñado desde cero y con la seguridad en mente. Este sistema operativo tiene unas capacidades de red muy superiores al anteriormente mencionado y mejores características de seguridad. Como punto negativo podría citarse que durante su existencia se han conocido muchas vulnerabilidades debidos a errores de diseño o implementación. La no disponibilidad de forma pública de su código fuente hace imposible auditar sistemas basados en Windows NT/2000 para garantizar su seguridad. Un problema que en ocasiones es más grave aún y también está relacionado con el hecho de ser un producto propietario y cerrado es que existe la posibilidad de puertas traseras. Por esta razón algunas instituciones y gobiernos han desechado su uso en determinadas situaciones. Microsoft sí tiene una política de distribución de código fuente de sus sistemas operativos, pero dicha distribución se realiza a compañías a las que Microsoft determina "capacitadas" para recibirlo.

Por último, debe considerarse la opción de los sistemas UNIX propietarios: AIX de IBM, HP/UX de HP, etc. Estos sistemas tienen características parecidas a GNU/Linux o BSD, con la excepción de que no se puede auditar su código dado que no es público. El caso del sistema operativo Solaris de SUN es un caso a considerar aparte ya que sí ofrece el código fuente de su sistema operativo (aunque no de forma libre). Sin

embargo, algunos de estos sistemas, como es el caso de AIX y de Solaris (con la definición de roles) pueden ofrecer capacidades de seguridad más avanzadas en el sistema operativo que GNU/Linux. Estas capacidades permiten implementar MAC ("Mandatory Access Control"). Actualmente, GNU/Linux dispone de un parche, aún no implementado dentro del núcleo distribuido oficialmente, que permite incorporar este mismo tipo de niveles de control de acceso de los usuarios (y aplicaciones) al sistema operativo.

Las auditorías de código son, por tanto, posibles o no en determinados sistemas operativos en función de la publicidad dada a su sistema operativo. Aún así, es necesario considerar los resultados de dichas auditorías. Si bien Microsoft y Sun ofrecen el código fuente de su sistema operativo (el primero con más restricciones que el segundo), ninguno de los dos incorporará, necesariamente, los resultados de una auditoría de código sobre la base del sistema operativo, los criterios para tomar dicha decisión no dependen de la auditoría en sí sino de la política de la propia compañía. Sin embargo, en la auditoría que se pueda realizar a sistemas operativos libres, como es el caso de GNU/Linux o BSD, la aplicación de los resultados o no se realiza mediante una discusión pública y es el propio resultado de la auditoría el que debe valer por si mismo para su introducción o no, no existen presiones comerciales de pérdida de imagen, ni el "time to market" ni ningún tipo de consideraciones que no sean las puramente técnicas. Este mismo hecho, la modificación inmediata del código y su distribución es el que puede dar lugar a que, aún cuando Sun distribuya de forma pública el código de Solaris, se audite de forma más intensiva el código de GNU/Linux o BSD, ya que son las propias personas que realizan la auditoría las que pueden implementar las modificaciones.

Nota de los autores: no disponemos información ni experiencia suficiente para hacer una valoración justa de MacOS.

4. Seguridad en el acceso a servicios de Internet

Una vez introducido el tema del documento y presentada la situación del software libre y de GNU/Linux en éste es el momento de entrar en materia.

En este apartado se presenta el primero de los escenarios planteados. Esto es, el caso en el que un sistema final, como puede ser un PC de sobremesa, accede a Internet para usar sus servicios. ¿Qué amenazas aparecen? ¿cómo puede defenderse uno ante ellas?

¿qué herramientas libres existen en GNU/Linux para ello? son algunas de las preguntas a las que se buscará una respuesta.

4.1. Amenazas en el acceso a servicios de Internet

Existen amenazas desde el momento en que un sistema informático accede a Internet para navegar, enviar y recibir correo electrónico, chatear, descargar ficheros, etc. No se pretende enumerar aquí todas, ya que para ello habría que analizar el entorno concreto, pero si se quiere dar una visión general de las más importantes o comunes, como lo son las siguientes:

1. Modificación o borrado de información almacenada en el ordenador.
2. Acceso no deseado a información confidencial almacenada.
3. Instalación no intencionada de programas maliciosos (conocidos como troyanos) en el sistema que puedan vulnerarlo o lo conviertan en plataforma de ataque hacia otros ordenadores.
4. Lectura o modificación (no deseadas) de información transmitida.

Estas amenazas surgen a través de diferentes medios. Entre ellos podemos destacar los siguientes:

-
- Viruses o troyanos que se difunden a través de una aplicación que manipule archivos o realice tareas sin intervención del usuario (esto incluye programas de correo electrónico, programas de descarga de archivos, algunas aplicaciones ofimáticas, etc).
- Malversación de servicios de red ofrecidos por el ordenador (en ocasiones sin que el usuario sea consciente de su existencia) y que tienen fallos de seguridad o una configuración inadecuada. Un ejemplo de esto es el acceso remoto (no permitido) al sistema a través de aplicaciones como telnet, ssh o similares.
- Páginas web dañinas. Generalmente el daño es causado empleando applets, código JavaScript o algún fallo de implementación del navegador empleado.
- La información enviada no va protegida mediante técnicas de cifrado. Esto no debe hacerse siempre, pero si en los casos en los que no se desee que esa información sea leída o modificada por terceros. Es importante se consciente de que, no sólo el sistema final con el que nos comunicamos, sino todos los sistemas intermedios tienen la posibilidad de capturar la información tranmitida.

4.2. Aplicaciones de protección

El primer paso para proteger el sistema es deshabilitar todas las posibles puertas de entrada para un intruso y limitando el impacto de una intrusión, el siguiente paso consistirá en usar una serie de herramientas que en conjunto introducen barreras de protección ante las amenazas descritas previamente.

Dentro de las herramientas de protección a priori se deben destacar las posibilidades de bastionado (o securización) del propio sistema operativo, este bastionado realizará una correcta configuración para reducir la exposición del sistema, por un lado, y limitar las posibilidades de afectar a éste en caso de que tenga lugar una intrusión.

Dentro de las herramientas de protección se van a analizar, por un lado, los denominados cortafuegos personales. Estas herramientas van a permitir el control de la información entrante y saliente de un sistema poniendo en manos del usuario la posibilidad de restringir el flujo en cualquiera de lo sentidos. Estas medidas de protección protegerán al usuario de algunos de los ataques realizados desde el exterior (derivados de intentos de comunicación no deseados) y de la comunicación del sistema con el exterior de forma incontrolado (motivado por la introducción de algún troyano). Por otro lado se comentarán las herramientas de protección contra código malicioso (viruses).

En esta sección se examinan distintas herramientas y aplicaciones que permiten garantizar en gran medida la seguridad de un sistema que accede a Internet. Se mostrarán herramientas libres y propietarias, dejando al lector que extraiga sus propias conclusiones en cuanto a funcionalidad y capacidades.

4.2.1. Herramientas de bastionado libres

En el caso de GNU/Linux el bastionado del sistema implica, por un lado: deshabilitar todos los servicios de red que no se van a usar y que por defecto están habilitados y, por otro, asegurar la configuración interna del propio sistema para reducir el impacto de un troyano o un usuario malicioso que ha entrado en éste.

Existen herramientas para realizar estas tareas de bastionado de forma automática. En particular cabe destacar Bastille (http://www.bastille-linux.org). A diferencia de otras que realizan, silenciosamente, las tareas de limitación de accesos a un servidor o arreglos sobre los sistemas de ficheros, Bastille intenta *educar* al administrador guiando el proceso de bastionado con una serie de preguntas sobe el uso que se le va a dar al sistema. Aunque también dispone de una serie de perfiles que se pueden implementar de forma inmediata sobre el sistema.

También es importante destacar la herramienta TITAN (http://www.fish.com/security/titan.html), una herramienta multi plataforma (soporta Linux, FreeBSD, así como distintas versiones de Solaris) para el bastionado de sistemas operativos UNIX. Se trata de una herramienta orientada más hacia el administrador avanzado y es menos educativa que Bastille a este respecto. Dispone, al igual que Bastille, de una serie de perfiles predefinidos.

4.2.2. Herramientas de bastionado propietarias

Algunos sistemas operativos propietarios también distribuyen herramientas de bastionado para sus sistemas. En concreto:

- Sun ofrece una herramienta para determinar la situación de parcheo de un sistema frente a los parches distribuidos y también ha hecho público el Solaris Security Toolkit (también conocido como JASS) para Solaris. Aunque Sun no ofrece oficialmente soporte para éste. Esta herramienta realiza, a través de más de setenta scripts, restricciones de usuarios y servicios ofrecidos en el sistema operativo.
- AIX. IBM ofrece una herramienta para determinar la actualización de parcheo frente a los boletines de seguridad emitidos por la compañía. De igual forma distribuye, dentro del paquete para Service Providers, Spupsec de AIX, una herramienta de bastionado que se reduce a restringir servicios ofrecidos de red y a hacr algunas modificaciones (mínimas) sobre el sistema de ficheros y las restricciones de acceso de usuarios.
- Microsoft ofrece para Windows 2000 y Windows NT una herramienta (hfnetchk) para determinar la actualización de parcheo de servidores frente a los Service Pack distribuidos por Microsoft. Igualmente, Windows 2000 ofrece una serie de perfiles dentro de la configuración de seguridad del propio sistema operativo que se pueden utilizar para restringir la disponibilidad de servicios.

Los autores no conocen herramientas de bastionado disponibles para otros sistemas operativos: Windows 95/98/ME/XP, HP-UX, etc.

4.2.3. Antiviruses basados en software libre

No existe, prácticamente, antiviruses desarrollados como software libre ya que las plataformas de software libre tradicionales (Linux y las variantes de BSD) no sufren de los defectos de diseño que han hecho de los antiviruses tan populares entre otras plataformas. En particular:

- Los controles de seguridad son parte inherente del sistema operativo, los usuarios no pueden escribir en cualquier zona de memoria ni modificar ficheros del sistema operativo.
- Existe un desacomplamiento entre el sistema operativo y las aplicaciones. La filosofía de UNIX de construir elementos que se van uniendo entre sí hace que las aplicaciones tradicionales (lectores de correo, navegadores de web) que han venido siendo fuente de introducción de viruses no pueden afectar de forma directa al sistema operativo a través de funcionalidades mal implementadas o configuradas (aunque sí a través de fallos en el software).

Esto no quiere decir que los viruses en estos sistemas operativos no existan pero, en realidad, son un conjunto muy básico y que no han dado lugar, hasta el momento, a propagaciones virulentas. Otra características, además, que hace que la propagación de un virus sea más difícil en los sistemas Unix es la gran heterogeneidad de plataformas. Evidentemente, en cuanto a configuración, pero aún más en cuanto a compatibilidad de binarios. El ritmo de alto crecimiento y actualización de Linux hace que haya muy pocos sistemas (en global) que tengan una idéntica disposición de núcleo del sistema, binarios y librerías.

Los sistemas operativos libres sí se han visto afectados, en algún momento determinado, por otro tipo de ataques que no han de confundirse con los viruses: gusanos. Es decir, programas que, aprovechando una vulnerabilidad en el sistema se propagan por sistemas interconectados. Cabe destacar que la única protección frente a este tipo de propagaciones, similares a las víricas, es la actualización constante de los sistemas y la reducción de los servicios potencialmente accesibles desde el exterior, tarea ésta realizada por las herramientas de bastionado.

El papel relegado actualmente a el software antivirus en los sistemas libres ha sido el de análisis, desde el punto de vista de servidor, de los ficheros que se transmiten a través de distintos medios (correo electrónico, servidores de ficheros, transferencia de archivos por ftp/www...). Existen implementaciones propietarias de software antivirus para este propósito pero también existe implementaciones libres que, aunque carecen de la base de datos de definición de virus, delegan en otro software (utiliznando una API estandarizada) la comprobación de los virus en sí. Cabe destacar: AMaViS (http://aachalon.de/AMaViS/), exiscan (http://duncanthrax.net/exiscan/),Wmailscanner (http://messel.emse.fr/~pplantie/wmailscanner/) y Scan4Virus (http://qmail-scanner.sourceforge.net/). Que son distintas implementaciones para el tratamiento de virus en servidores de correo libres (Sendmail, exim, y qmail respectivamente).

4.2.4. Antiviruses propietarios

Evidentemente, al igual que se ha hablado previamente que los sistemas operativos libres carecen de software de antivirus por no ser fruto de ataques de éstos, existen un número muy alto de plataformas de antivirus propietarias. En gran medida debido a que son éstas plataformas las más afectadas por los ataques de viruses. No parece preciso enumerar todas las herramientas, pero si cabe destacar las compañías: Mcafee, Norton (Symantec), Trend Micro, Sophos, F-Secure y Panda Software.

4.2.5. Cortafuegos personales libres

Como se ha comentado, en GNU/Linux y BSD el propio sistema operativo ofrece la funcionalidad de filtrar paquetes. Sobre esta capacidad se han creado diversas herramientas que facilitan su configuración y que adquieren la fiabilidad del sistema sobre el que se usan. Esto es una enorme ventaja respecto a los casos en los que cada herramienta tiene que implementar su filtro y en los que el usuario no puede estar seguro de su fiabilidad. Los distintos sistemas operativos propietarios han empezado a incorporar capacidades de filtrado de paquetes (aunque rudimentarias) en sus últimas versiones (este es el caso de Windows 2000 y Solaris 8).

Estas herramientas se caracterizan por la funcionalidad que ofrecen (en particular sus funcionalidades avanzadas) y por la facilidad de uso. En general su función central consiste en actuar de cortafuegos en el acceso hacia y desde la red. Esto implica que se establecen reglas de filtrado que especifican en que casos se permitirá que se establezcan conexiones desde el exterior hacia dentro y desde dentro hacial el exterior. Además de este filtrado algunas herramientas permiten realizar una configuración de los servicios de red, por ejemplo cerrando aquellos que no se usen. En GNU/Linux la gran ventaja de estas herramientas es que se basan en los servicios de filtrado que ofrece el sistema operativo, es decir, son una capa de abstracción de un servicio común. Esto evita la fragmentación y, al mismo tiempo, hace más fiable las funcionalidades del sistema operativo dado que son probadas de forma intensiva por todas las implementaciones.

A continuación se presentan las que se han considerado más interesantes para un usuario final.

4.2.5.1. FireStarter

Esta es una herramienta de configuración de cortafuegos pensada para usuarios finales que no tienen porqué tener grandes conocimientos de seguridad. Dispone de un asistente que permite crear configuraciones básicas. Posteriormente puede mejorarse

añadiendo y creando reglas (a las que denominan reglas dinámicas). En las siguientes figuras pueden observarse el asistente y la posterior adición de reglas.

Figura 1. Asistente de configuración del cortafuegos

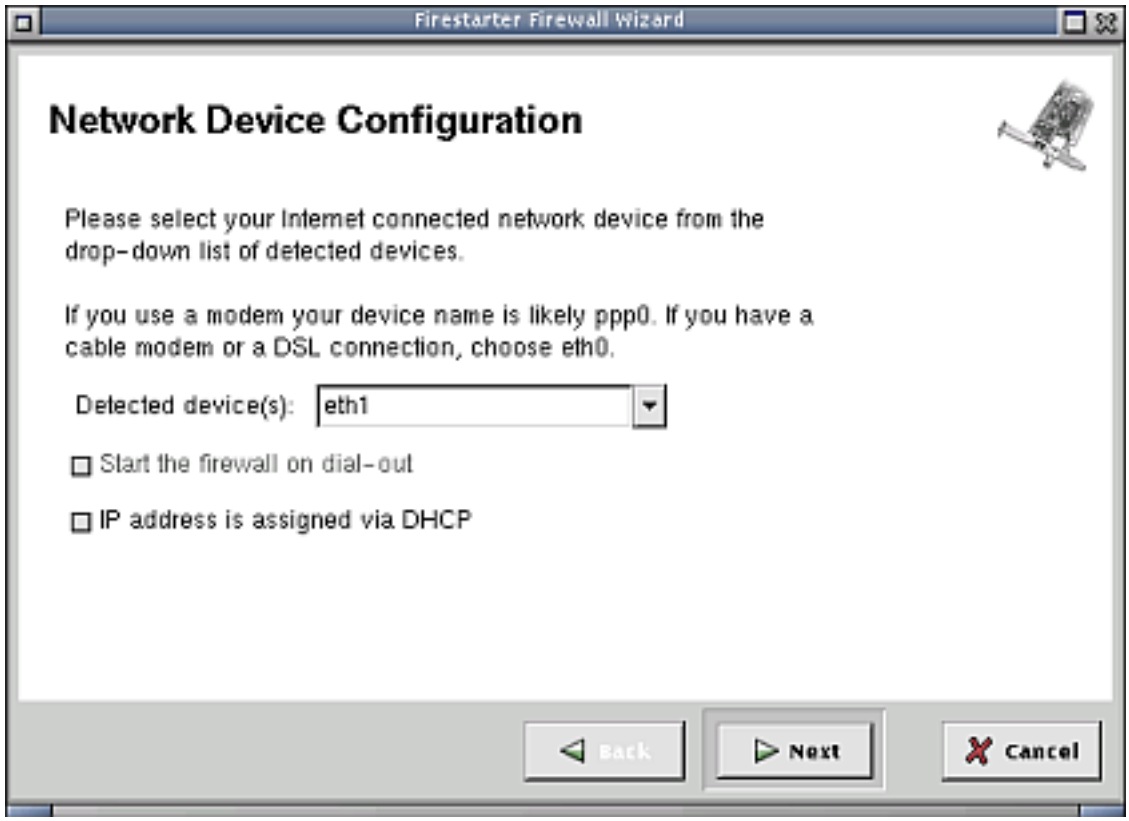
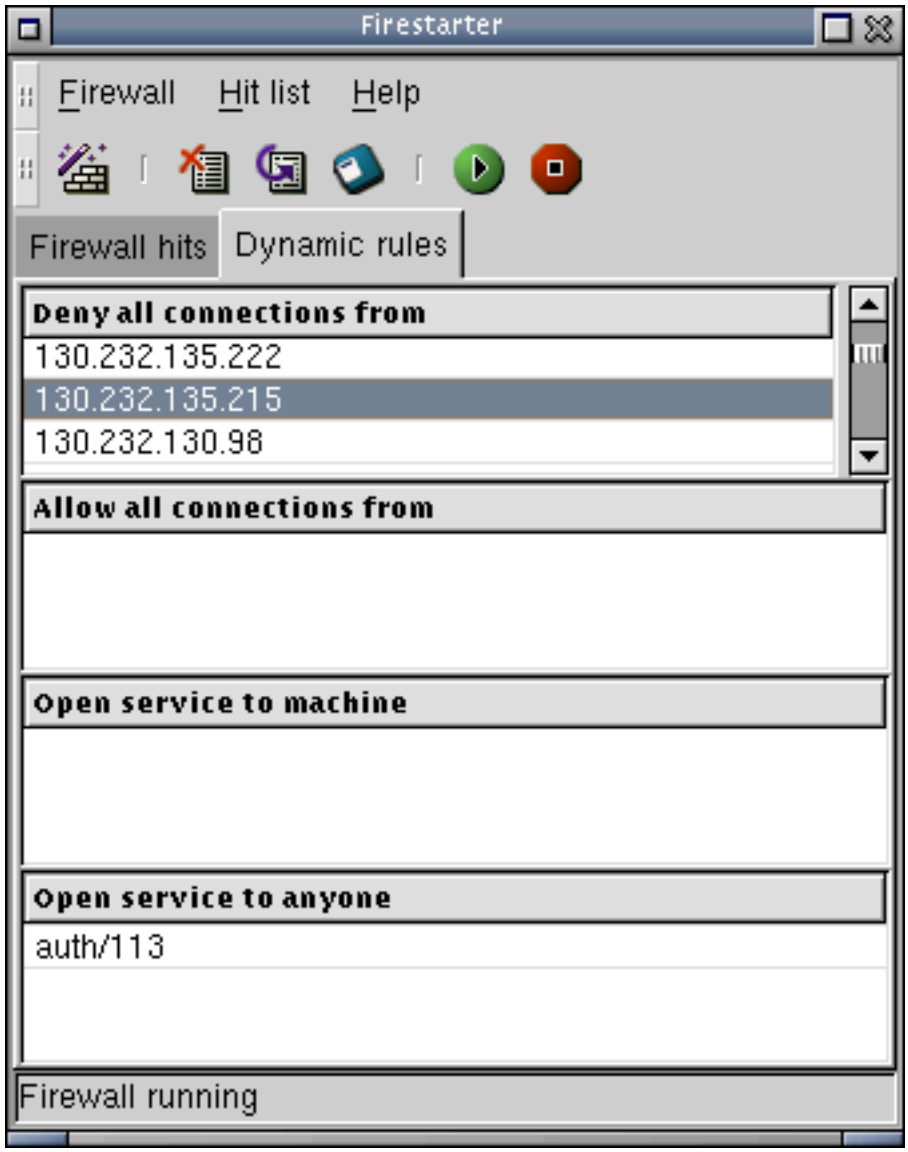


Figura 2. Edición manual de las reglas dinámicas del cortafuegos



Además dispone de un monitor de red que con el que se pueden observar intentos de ataque cuando se estén produciendo.

FireStarter está traducido a un gran número de idiomas, incluyendo el español.

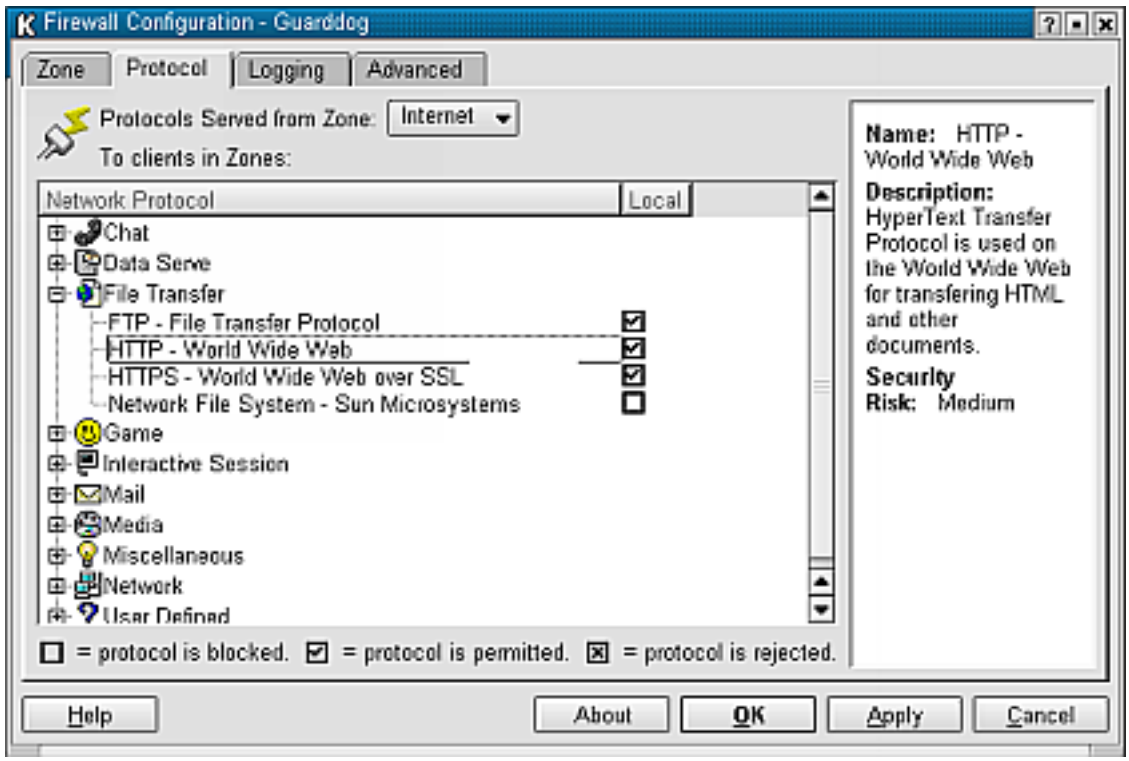
Esta herramienta funciona bajo entornos Linux 2.2 y 2.4 y se distribuye como parte del entorno GNOME bajo licencia GPL. Su página oficial es firestarter.sourceforge.net (http://firestarter.sourceforge.net).

4.2.5.2. GuardDog para KDE2

GuardDog es una herramienta de protección pensada para plataformas Linux (versión 2.2) y completamente orientada a usuarios no expertos. Por ello no es necesario disponer de conocimientos ni de seguridad ni, del funcionamiento de las redes TCP/IP o de los parámetros internos del sistema operativo.

Mediante una interfaz gráfica el usuario debe indicar que desea que el cortafuegos a nivel de cada aplicación, no de IP como ocurre en otras herramientas. Gracias a este mayor nivel de abstracción se reduce la posibilidad de cometer errores. Esta garantía se ve incrementada por el hecho de que sigue una filosofía de que todo lo que no se permite explícitamente está prohibido.

Figura 3. Configuración de los permisos de cada aplicación para los ordenadores de la zona "Internet" en GuardDog



Por su puesto Guarddog, igual que FireStarter, permite gestionar y modificar el cortafuegos una vez creado.

En nuestra opinión, la calidad de su interfaz y un diseño bastante bien pensado hacen que esta sea una de las herramientas de cortafuegos más apropiadas para usuarios finales de todas las existentes entre el software libre y propietario. Esto lo ha conseguido gracias a su orientación completa a satisfacer a este tipo de usuarios a cambio de renunciar a ser útil en entornos más complejos, como los que se discutirán en el apartado de "Aspectos de seguridad para ofrecer acceso a Internet".

4.2.5.3. Lokkit

De nuevo esta es una herramienta diseñada explícitamente para usuario finales. Para usarla no es necesario comprender las reglas de filtrado subyacentes, simplemente va realizando una serie de preguntas y a partir de las respuestas crea un conjunto de reglas personalizadas.

Lokkit ha sido pensado para ordenadores que acceden a Internet a través de un módem telefónico o de cable. Eso permite que sea fácil de usar sin embargo no es la mejor herramienta para entornos más complejos.

Otros detalles que se pueden comentar sobre Lokkit son que aunque está preparada para ser adaptada a múltiples idiomas aún no tiene soporte para español y que las últimas versiones está avanzando en su integración con el entorno GNOME.

Puede obtenerse el software y más información sobre Lokkit en la página oficial de la herramienta (http://roadrunner.swansea.linux.org.uk/lokkit.html).

4.2.5.4. Linux Firewall de linux-firewall-tools.com

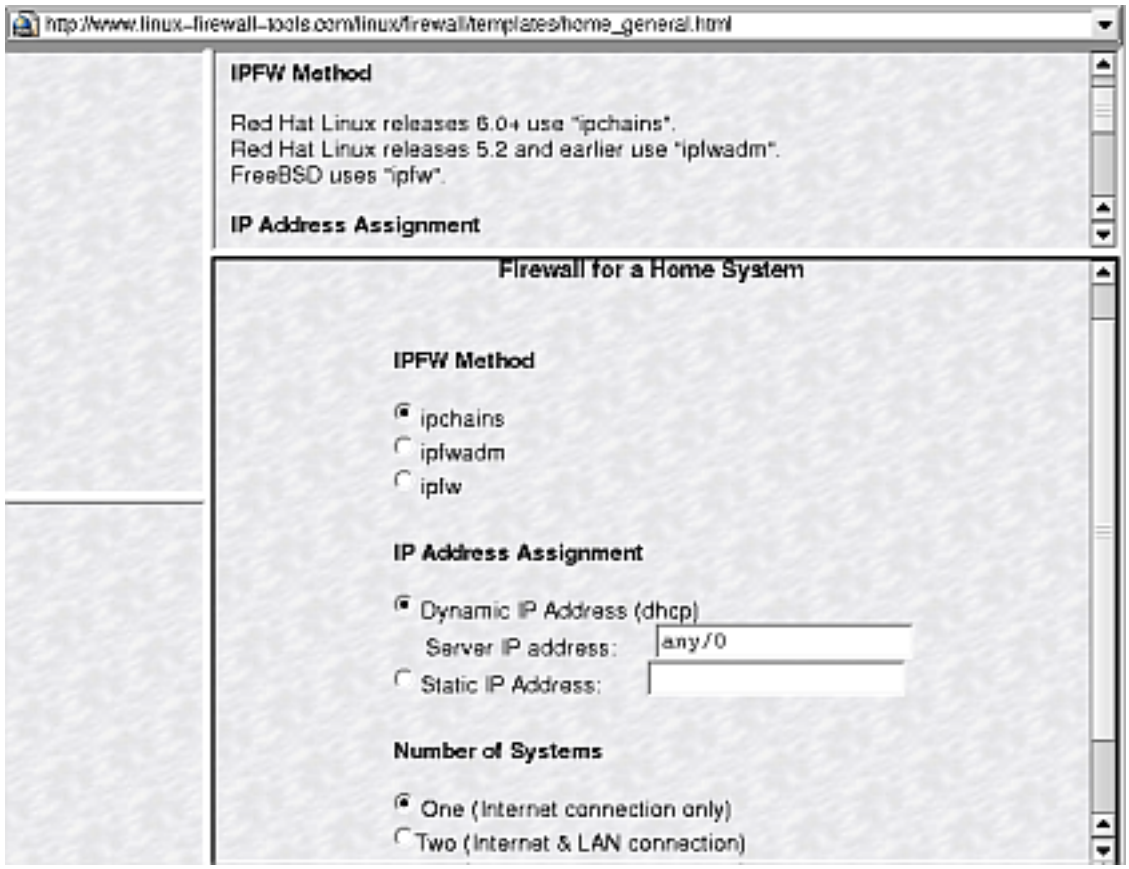
Esta herramienta destaca por que accede a ella a través de una interfaz web. Está disponible a través de la página: http://www.linux-firewall-tools.com/linux/firewall/ (http://www.linux-firewall-tools.com/linux/firewall/)

Tras comenzar la configuración del cortafuegos un asistente va realizando preguntas para conocer nuestras necesidades. Estas preguntas son más o menos sencillas, aunque es necesario tener ciertos conocimientos de redes y seguridad para poder contestarlas.

Con Linux Firewall se permite configurar un cortafuegos a través de un formulario on-line. Incluye soporte para Linux 2.2, Linux 2.4 y FreeBSD.

A continuación se incluye una captura de esta herramienta en pleno funcionamiento.

Figura 4. Comienzo del asistente que guía en la construcción del cortafuegos



4.2.6. Cortafuegos personales propietarios

Los cortafuegos personales propietarios siguen un enfoque similar al ya presentado para los cortafuegos personales de GNU/Linux. Las más importantes están dirigidas a entornos Windows personales (95/98/Me) que carecen de servicios de seguridad avanzados a nivel de sistema operativo. En los entornos Windows profesionales (NT y 2000) ya no se habla de cortafuegos personales. Aunque es posible instalar éstos en dichos entornos, e incluso Windows 2000 incorpora una rudimentaria definición de cortafuegos en la configuración (avanzada) de interfaces de red. Sin embargo, si se desean utilizar estos sistemas como cortafuegos se utilizan cortafuegos de gama alta (entre ellos Checkpoint Firewall-1).

Adicionalmete a las funciones de filtrado, algunos cortafuegos personales suelen incorporar capacidades de establecimiento de comunicaciones a través de redes privadas virtuales (con IPsec) y limitaciones en el envío y recepción de cookies (aún cuando ésta deba ser una tarea gestionada por el propio navegador de web).

Algunos de los cortafuegos personales propietarios más conocidos son:

- zone-alarm
- tiny-personal-firewall
- PGPnet
- AtGuard
- Norton Personal Firewall
- Black Ice Defender

Una fuente de vulnerabilidades en estos cortafuegos dirigidos al entorno doméstico surge de la intención de limitar al máximo las posibilidades ofrecidas a la persona que lo utiliza, dado que puede no estar versado en temas de seguridad (ni que saber que el puerto 80 es el de www). Estas presuposiciones pueden suponer la posibilidades de amenazas en el sistema aún cuando esté “supuestamente” protegido por un cortafuegos . Es interesante consultar el apartado "Personal Firewalls and IRC Zombie/Bot Intrusions" del artículo disponible en http://grc.com/dos/grcdos.htm (http://grc.com/dos/grcdos.htm)se describe vulnerabilidades graves encontradas en uno de los cortafuegos personales más conocidos.

5. Aspectos de seguridad para ofrecer acceso a Internet

5.1. Introducción

Esta sección está dedicada al caso en el que se desee dar acceso a Internet a un conjunto de ordenadores. Esta situación puede darse en el caso de una empresa u organización que quiere conectar a sus empleados a Internet o en el caso de que se desee montar un proveedor de servicios de Internet (ISP).

En la sección anterior se habló en general de aplicaciones de protección. Ahora es necesario distinguir entre los diferentes mecanismos que pueden ofrecerse para conseguir esta protección: filtros de paquetes, “TCP wrappers”, etc.

Cabe destacar que todo lo dicho para el caso de acceso de ordenadores individuales a Internet, sigue siendo válido para este caso más general. Por un lado todos los ordenadores a los que se de acceso a Internet podrían aplicar los mecanismos descritos en el apartado anterior para controlar individualmente su seguridad. Por otro el sistema que sirva de punto de acceso podrá aplicar esas mismas técnicas o similares para protegerse a si mismo y ofrecer una protección global a la red interna.

La principal diferencia es que ahora la preocupación no es por un ordenador individual sino por toda una red. Por ejemplo, en el caso de un virus de mail el objetivo será evitar que llegue a ninguno de los ordenadores de la red a la que se da acceso. Si bien, cabe la posibilidad de proteger individualmente cada uno de los ordenadores es mucho más eficiente (en tiempo y dinero) solucionarlo de forma general para todos, instalando las herramientas adecuadas en los sistemas a través de los que accede toda la red.

También adquieren importancia algunas nuevas amenazas que no eran necesario considerar en el caso de acceso con un ordenador individual. Algunas de ellas se describen a continuación.

Un caso que se da con excesiva frecuencia ocurre cuando los usuarios de una red local usan las posibilidades que ofrecen sus sistemas operativos para compartir ficheros. Si en un momento dado la red local se conecta a Internet los usuarios pueden no ser conscientes de que están compartiendo sus ficheros con cualquier ordenador conectado a Internet. Esta es una amenaza de acceso no autorizado a información y esta información puede ir desde claves de acceso, información personal (económica por ejemplo), datos confidenciales de la empresa, etc. El asunto se agrava por el hecho de que aunque una persona de la red local tenga cuidado con su información compartida es posible que otra a la que le pasa esta información no haya tenido cuidado y al final una persona no autorizada puede acceder a ella.

Además de las amenazas enumeradas en la sección anterior pueden citarse otras dos que pueden tener bastante gravedad:

- Ataques que impidan la conexión a Internet. Este es un tipo de ataque de Denegación de Servicio.
- Intrusión en el sistema que sirve de punto de acceso. Esta es una amenaza muy grave dado que una vez en este sistema el atacante puede acceder con mucha mayor facilidad a cualquier ordenador de la red local.

5.2. Sistemas de cortafuegos

En la sección anterior ya se comenzó el tratamiento de los sistemas de cortafuegos. Sin embargo entonces el tratamiento era desde el punto de vista del uso que podía darle un individuo para proteger su ordenador conectado a Internet.

Ahora se profundizará en estas herramientas para abarcar el caso de un cortafuegos que se sitúa entre Internet y una red local a la que debe proteger.

Podemos clasificar los cortafuegos en dos tipos genéricos:

Filtros de paquetes (“packet filters”)

> En los que se controla el envío, recepción y retransmisión de los paquetes TCP/IP que atraviesan el cortafuegos. Dentro de este tipo se encontrarían los filtros de paquete con inspección de estados que incorporarían el conocimiento del estado de las comunicaciones al filtrado de éstas.

Pasarelas de aplicación (“application proxies”)

> En este esquema la comunicación no se realiza nunca directamente contra la aplicación (servidor web, correo electrónico, ...). La conexión del sistema remoto se realiza contra la pasarela instalada en el cortafuegos. Esta comprueba la operación que se desea realizar sobre la aplicación, permitiendo de esta forma un control del origen, destino y contenido de la comunicación.

5.2.1. Envoltorios TCP (TCP Wrappers)

Estos filtros se sitúan entre la red y cada una de las aplicaciones que escuchan peticiones de la red. Se denominan así porque se sitúan al nivel del protocolo de transporte de Internet, TCP (ver glosario)

En los sistemas UNIX, y GNU/Linux no es una excepción, el filtro TCP más utilizado es tcpd de Wietse Venema. Esa herramienta permite especificar desde que ordenadores se podrá conectar y desde cuales no a cada uno de los servicios de nuestro ordenador.

Lo más habitual es pensar que un ordenador de usuario final no tiene ningún servicio y por tanto esta protección no es necesaria, pero esto no es así. Como se ha dicho, por defecto hay diversos servicios de red activados. E incluso aunque se desactiven, en todo sistema operativo de red hay aplicaciones que ’abren’ servicios para sus propios propósitos.

La configuración de tcpd es muy sencilla y por ello no existe ninguna herramienta específica para realizarla. Se basa en dos archivos, `/etc/hosts.allow` y

`/etc/hosts.deny` donde se especifica desde donde se permite y desde donde no se permite respectivamente acceder a cada servicio. Además, estas definiciones pueden ser comprobadas con las herramientas tcpdchk y tcpmatch para verificar los posibles casos de uso.

Uno de los puntos débiles de tcpd y sus herramientas auxiliares es la falta de una herramienta de configuración de fácil uso y que integre todas las posibilidades que ésta ofrece. Sin embargo, con la importancia creciente que se está dando a este aspecto dentro del mundo del software libre es posible que aparezca alguna pronto.

5.2.2. Filtros de paquetes

Un filtro de paquetes puede admitir, rechazar o simplemente descartar los paquetes que le llegan con destino a una red interna o salientes desde ésta. Estas decisiones se toman en función de una serie de reglas establecidas por el administrador. Estas reglas pueden basarse, en principio, en la siguiente información contenida en un paquete:

- Direcciones origen y destino
- Puertos TCP origen y destino
- Protocolo empleado

Una generación posterior de los cortafuegos de filtrado de paquetes puede incorporar una tecnología conocida como “inspección de estados” que permite validad los paquetes en función del estado de la conexión. Es decir, los paquetes TCP podrán rechazarse si no siguen el protocolo establecido (“three-way handshake”), por ejemplo, si se envía un paquete FIN sin haber establecido una comunicación previamente. Igualmente, acercándose cada vez más a los cortafuegos de pasarela de aplicación, se podrá limitar una comunicación a nivel de aplicación si no sigue el protocolo correcto, por ejemplo, el envío de un LS en una comunicación FTP sin haber realizado previamente la autenticación con el comando USER.

GNU/Linux (y también las distintas versiones de BSD) disponen de sistemas de filtrado de paquetes integrados en el propio sistema operativo. Esta integración ofrece una gran fiabilidad a este filtrado, ya que pasa siempre por el núcleo del sistema operativo antes de llegar a ninguna aplicación en el espacio de usuarios.

Este soporte de filtrado de paquetes viene dado por:

ipfwadm

Es la versión usada por las versiones antiguas de GNU/Linux (la serie 2.0)

ipchains

Es usado por las versiones de GNU/Linux 2.2. Contiene mejoras significativas como la posibilidad de crear grupos de reglas arbitrarios. Es decir no se limita al administrador a los típicos "input", "output" y "forward".

iptables (netfilter)

Es la nueva versión incluida (y escrita completamente desde cero) en la última serie del kernel de GNU/Linux, la 2.4. Contiene varias novedades muy interesantes. La más importante de ellas es que se implementa mecanismos de inspección de estados.

También se han hecho grandes mejoras en su gestión, que ahora es a la vez más fácil y potente. A pesar de ello iptables es compatible con versiones anteriores.

ipfw

Empleado por FreeBSD

SINUS

Es una alternativa independiente a las opciones anteriores. Es un producto bastante completo que se distribuye bajo la licencia GPL y que dispone de documentación bastante buena y herramientas de configuración propias. Para obtener SINUS o conseguir más documentación sobre él puede visitarse su página oficial (http://www.ifi.unizh.ch/ikm/SINUS/firewall/).

El soporte de filtrado viene acompañado de una interfaz de administración de línea de comandos, pero además existen un número interesante de aplicaciones (gráficas en su mayoría) que facilitan su gestión. Generalmente estas aplicaciones funcionan para más de una herramientas listadas antes. Algunas de estas, pensadas para usuarios finales sin conocimientos profundos de seguridad han sido descritas en la sección anterior. Las herramientas que aquí se indican siguen siendo de gran ayuda para el administrador de red que tiene que configurar un cortafuegos por cuanto permiten una configuración más versátil de las reglas de este tipo de cortafuegos.

Generalmente el cortafuegos se sitúa en el sistema que actúa de encaminador (router) y que de todas formas es necesario, así que no es imprescindible incorporar hardware extra a la arquitectura de conexión.

5.2.2.1. gfcc

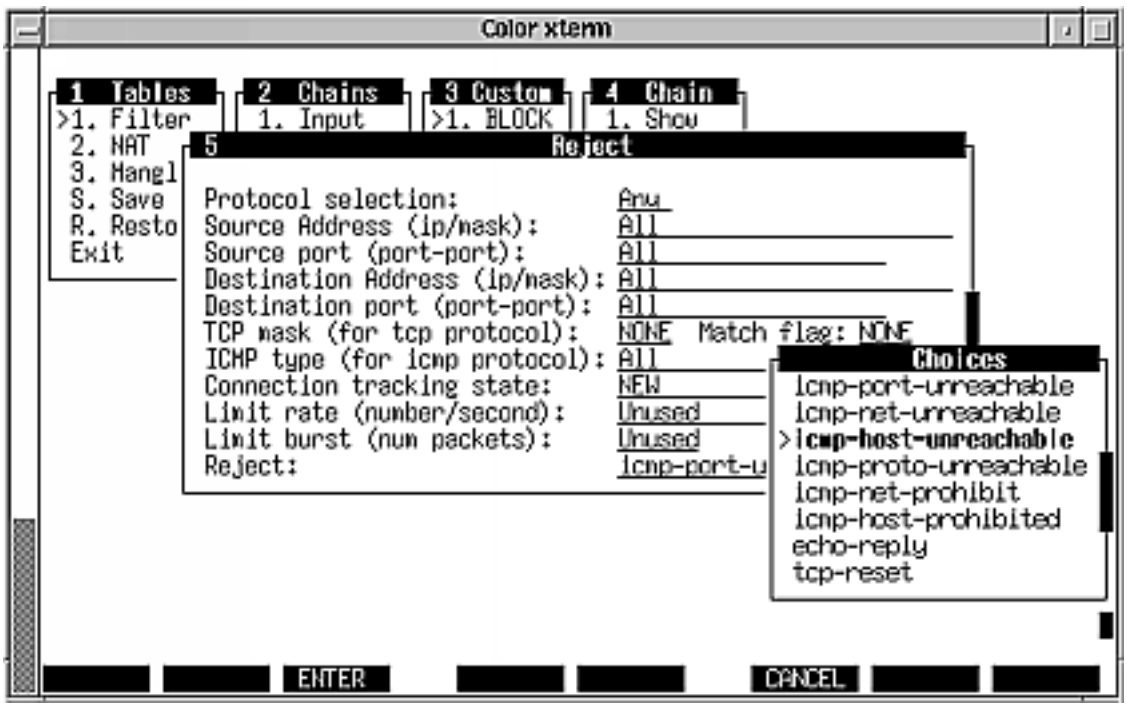
Permite configurar reglas de ipchains e ipfwadm de forma gráfica pero dejando al administrador el mismo control que la herramienta de línea de comandos. Para usar esta herramienta es necesario conocer la filosofía de funcionamiento de ipchains o ipfwadm.

5.2.2.2. ipmenu

Es una interfaz de usuario basada en la librería ncurses. En otras palabras es una interfaz textual basada en menús. A pesar de no ser gráfico es fácil de usar a la vez que potente en flexible. Está pensado para configurar iptables y permite desde una única interfaz configurar el filtrado y modificación de paquetes, la traducción de direcciones (NAT) y las características de encaminamiento (iproute2).

A continuación se muestra una captura de ipmenu. En ella se puede apreciar la gran potencia de esta herramienta.

Figura 5. Uno de los menús de ipmenu (Fuente: página oficial)



Para obtener más información sobre ipmenu se recomienda visitar su página oficial (http://users.pandora.be/stes/ipmenu.html).

5.2.2.3. Easyfw

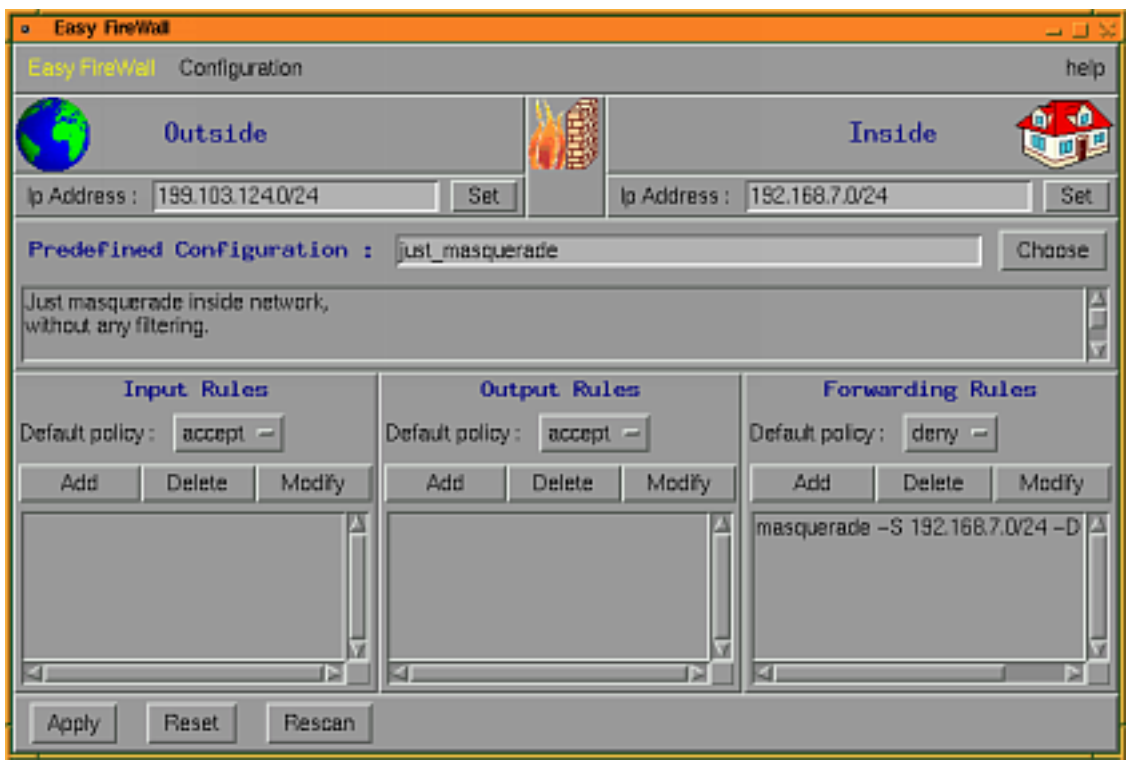
Esta es una herramienta gráfica bastante fácil de usar. Tras establecer la dirección de la interfaz interna y la dirección de la interfaz externa se pasa a configurar de forma independiente reglas para los paquetes entrantes (input rules), salientes (output rules) o que van a ser reenviados (forwarding rules).

Además dispone de la funcionalidad adicional de permitir almacenar varias configuraciones e incluso añadir comentarios a cada una para identificarlas fácilmente.

Easyfw incluye soporte para GNU/Linux únicamente, en concreto para ipfwadm e ipchains.

A continuación se incluye una captura de pantalla de la herramienta. Para obtener más información y obtener el programa se puede consultar su página web (http://www.linux-kheops.com/pub/easyfw/easyfwGB.html).

Figura 6. Pantalla principal de easyfw (Fuente: página web oficial)



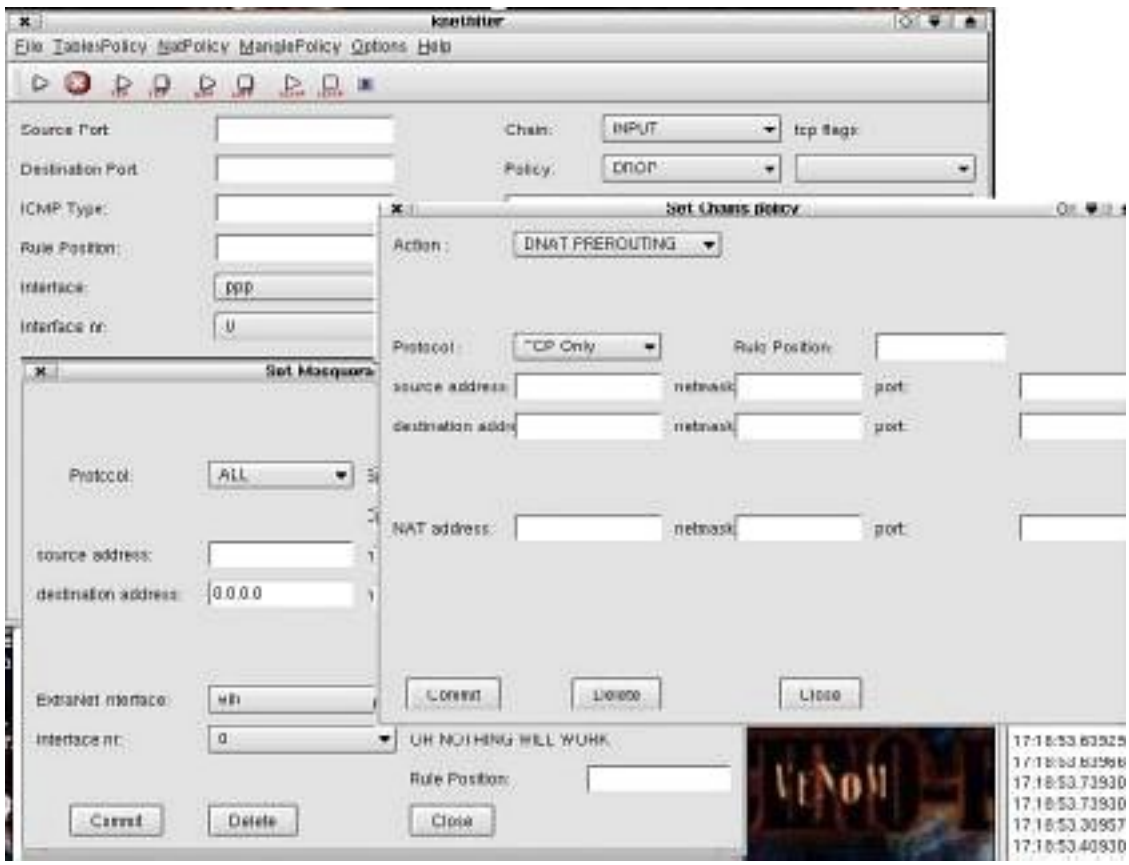
5.2.2.4. Knetfilter

Es una interfaz interfaz de usuario de configuración de cortafuegos para el escritorio KDE. Es muy completo permitiendo realizar configuraciones comunes rápidamente a la vez que configuraciones más complejas y personalizadas.

Una característica muy interesante es su posible integración con tcpdump para examinar los paquetes que transcurren por la red y con nmap para realizar un escaneo para comprobar la fiabilidad del cortafuegos una vez configurado. Más adelante se describirán con más detalle estas herramientas.

Cabe notar que knetfilter sólo se encuentra disponible para la versión 2.4 de Linux.

Figura 7. Escritorio mostrando varias ventanas de knetfilter durante la configuración de un cortafuegos.



Knetfilter puede obtenerse en http://expansa.sns.it/knetfilter/ (http://expansa.sns.it/knetfilter/). Esta página por contra no incluye mucha información sobre esta aplicación.

5.2.2.5. FERM

Esta es una herramienta de ayuda a los administradores que tienen que mantener cortafuegos muy complejos. FERM permite almacenar todas las reglas en un fichero y cargarlo empleando un comando. Este fichero de configuración tiene una sintaxis similar a un lenguaje de programación partiendo crear listas y agrupación por niveles de las reglas.

FERM se distribuye bajo la licencia GPL.

Puede obtenerse el software y más información sobre Lokkit en página oficial de la herramienta (http://www.geo.vu.nl/~koka/ferm/).

5.2.2.6. Mason

Ofrece una forma de configuración de cortafuegos de una forma enormemente original. Básicamente permite configurar un cortafuegos aprendiendo a partir de tráfico real generado en la red donde se desea que actúe el cortafuegos. La idea es instalarlo e ir generando tráfico de todos los tipos que se desea considerar a la vez que se indica si debe permitirse su paso o no.mason puede ser integrado con gfcc (ver http://www.govirtual.com.au/gfcc+mason.html) (http://www.govirtual.com.au/gfcc+mason.html).

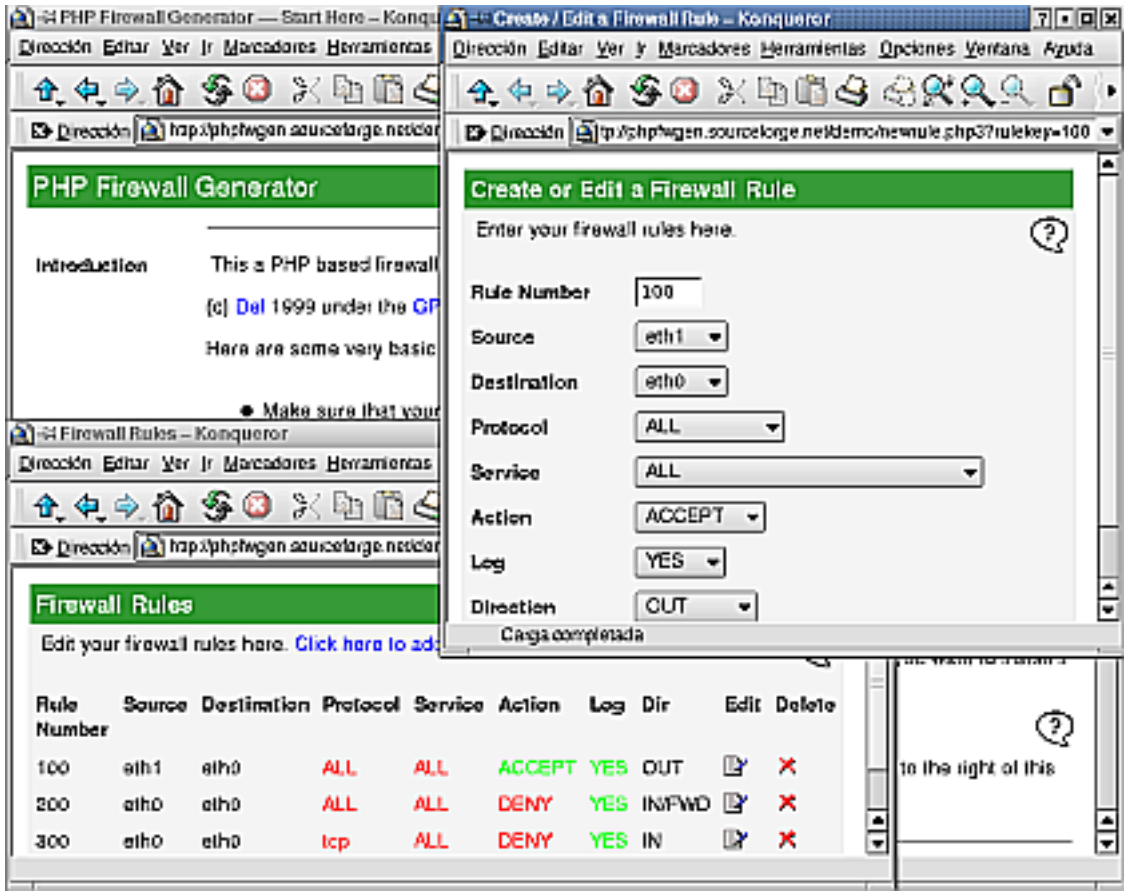
5.2.2.7. PHP firewall generator

Esta herramienta está pensado para ser instalado en un ordenador propio con soporte para PHP (ver glosario). Se accede a ella a través de un navegador y mediante una serie de formularios se configuran directamente las reglas de filtrado del cortafuegos.

Esta aplicación se distribuye bajo licencia GPL y está alojado en SourceForge y su página del proyecto puede encontrarse en la dirección http://phpfwgen.sourceforge.net (http://phpfwgen.sourceforge.net)

A continuación se muestra una captura bastante completa que incluye la ventana principal y dos ventanas auxiliares, una con una lista de reglas (abajo) y otra editando una de estas reglas (arriba a la derecha).

Figura 8. Varias ventanas configurando un cortafuegos con PHP Firewall Generator



Una última opción interesante es Firewall configuration toolkit (TCP) que puede obtenerse en fct.linuxfirewall.org (fct.linuxfirewall.org). Una vez instalado puede configurarse un cortafuegos rellenando información sobre la política de seguridad deseada a través de una interfaz web.

5.2.3. Cortafuegos propietarios

Una alternativa propietaria a los cortafuegos libres presentados es Firewall-1 de Check Point. Puede obtenerse más información de este cortafuegos en http://www.checkpoint.com/products/firewall-1/

(http://www.checkpoint.com/products/firewall-1/). Este es uno de los cortafuegos líderes del mercado, junto con el módulo de cortafuegos PIX de Cisco (http://www.cisco.com/). Las capacidades de ambos son, en principio, similares a los productos presentados previamente aunque dirigidos a una solución completa que integra muchas otras funciones que las que se podrían, en principio, atribuir a un cortafuegos de filtrado. En el caso de Firewall-1, dispone de un software de gestión y monitorización del cortafuegos (que en los productos de software libre son funciones separadas debido a la habitual arquitectura modular de UNIX), generación de redes privadas virtuales (VPN), integración con software de alta disponibilidad, y capacidades de filtrado de URLs y código malicioso, del que carecen en principio, las tecnologías anteriores (aunque pueda ser un componente proporcionado por otros proyectos de software libre su integración no es inmediata).

5.2.4. Pasarelas de aplicación (“proxies”)

Este tipo de herramientas se instalan en sistemas intermedios entre los ordenadores finales de la red interna e Internet. Cuando se desea conectar con el exterior se debe hacer a través de la pasarela, que será la que realmente se comunicará con los ordenadores externos. Cuando reciba una respuesta de estos la reenviará al ordenador interno que inició la conexión. La forma de hacer esto es específica de cada aplicación y por tanto sólo podrá usarse la pasarela con aquellas aplicaciones específicamente soportadas por esta. Las aplicaciones más usuales como la navegación por el web (HTTP) o el correo electrónico están soportadas por un gran número de pasarelas.

La gran ventaja de este esquema es que se tiene un control global sobre la seguridad y además este control se tiene a nivel individual sobre cada una de las aplicaciones. Esto permite comprender y mantener el estado en el que se encuentra una comunicación y con ello reconocer y evitar un mayor número de ataques.

Pero las pasarelas de aplicación también tienen inconvenientes. Una de ellas es precisamente que requiere una configuración específica para cada uno de los usos que se van a hacer de la red: HTTP, FTP, telnet, correo, news, etc. Además las aplicaciones finales de los usuarios deben estar preparadas para usar la pasarela como sistema intermedio para llegar a un destino. Afinando un poco más, en realidad esta última afirmación no es del todo cierta. Últimamente han aparecido las pasarelas transparentes que con algo de ayuda del sistema operativo permiten que las aplicaciones no tengan que ser modificadas.

En este tipo de cortafuegos, las implementaciones propietarias son múltiples destacando Gauntlet Firewall de PGP, y Raptor Eagle de Axent. Estos cortafuegos incorporan generalmente capacidades mixtas de proxy y de cortafuegos de filtrado

(aunque no estén diseñados específicamente para esta función), pudiéndose establecer reglas de filtrado, de traducción de direcciones, redirección de puertos... De igual forma, dado que se hace una inspección de los contenidos, pueden integrarse con otras soluciones para categorización y filtrado de URLs, análisis de código malicioso, capacidades de antivirus, etc. Existe, por ejemplo, una plataforma de e-ppliance de la compañía Nokia basada en Gauntlet para ofrecer una solución antivirus de *caja negra*. En cualquier caso, es habitual encontrar en estos cortafuegos propietarios capacidades de establecimiento de redes privadas virtuales y de alta disponibilidad. Aunque ésta última característica aún está poco desarrollada en las implementaciones propietarias de estos cortafuegos.

Las implementaciones de tecnología proxy, en el campo de software libre, están, sin embargo, muy retrasadas frente a las implementaciones propietarias. En el caso del cortafuegos Gauntlet, se disponen de proxies a nivel de aplicación para más de treinta servicios distintos (desde http hasta bases de datos Oracle, servicios de X, whois, finger, correo...). Existen proxies disponibles en software libre para los servicios de http, DNS, X, correo, FTP e IRC, pero no existe un paquete integrado que los ofrezca todos de forma conjunta con una interfaz de administración única. Además, las capacidades de control de estos proxies están lejos de igualarse a las ofrecidas por los cortafuegos de proxy propietarios.

6. Ofrecer servicios de forma segura

6.1. Introducción

Existen diversas amenazas cuando una empresa, organización o individuo decide poner a disposición de un grupo de usuarios información o servicios a través de una red de comunicaciones. Evidentemente, el ofrecer un servicio al mundo exterior significa abrir una puerta a una red que podría ser totalmente impermeable al exterior. Esto es así porque un servicio puede tener vulnerabilidades que permitan que un atacante pueda malversarlo para realizar funciones distintas a las que se le suponían en principio.

Las medidas de protección de una entidad que desee ofrecer estos servicios son, además de las indicadas previamente (sistemas de protección perimetral), aquellas dirigidas hacia asegurar que las aplicaciones finales (ofrecidas a los clientes) funcionan correctamente y carecen de fallos en su configuración, o vulnerabilidades, que puedan hacer que sean malversadas.

6.2. Aplicaciones para realizar auditorías de seguridad de sistemas.

Las herramientas de auditoría permiten detectar, de forma rutinaria, problemas de seguridad para los que pudieran existir ataques conocidos. Este tipo de programas no sustituyen al sentido común ni a la experiencia de un buen administrador, sino que suponen una ayuda para realizar algunas tareas rutinarias que pueden llevarle mucho tiempo.

Estos programas pueden operar a muchos niveles, desde la comprobación de la pertenencia de archivos a usuarios y grupos del sistema hasta pruebas sobre aplicaciones instaladas para verificar si éstas tienen agujeros conocidos. Una forma sencilla de demostrar que una aplicación es vulnerable sería, por ejemplo, comprobar la versión de ésta , y ver si se trata de una versión que tuviera un problema especialmente grave.

Dentro de las herramientas de auditorías podemos dividirlas en dos tipos: auditorías internas también conocidas como de caja blanca, o auditorías externas, también conocidas como de caja negra. Las de caja blanca se realizan con conocimiento interno del sistema, habitualmente haciendo el análisis desde dentro del mismos sistema, mientras que las de caja negra se realizan sin conocimiento previo del sistema ni investigación de su contenido. Poniendo una analogía, si uno tuviera una caja de cartón, la auditoría de caja negra serían moverlo para ver cómo suena e intentar adivinar que hay dentro y la auditoría de caja blanca sería levantar la tapa y mirar dentro.

6.2.1. Auditorías de caja negra

La herramienta libre para auditorías externas más desarrollada es Nessus (http://www.nessus.org/). Esta herramienta cuenta con más de seiscientas pruebas de seguridad y está siendo desarrollada de forma activa, por lo que el número de pruebas de vulnerabilidades aumenta a medida que salen nuevas vulnerabilidades en sistemas. Esta herramienta cuenta con un servidor (que ha de ejecutarse en un sistema GNU/Linux) y con un cliente gráfico desde la que se configuran y lanzan las pruebas, el cliente está disponible para Linux (con interfaz GTK), Java y Win32.

Existe un buen número de herramientas propietarias para auditorías externas como son: CyberCop Scanner, Retina de eEye e Internet Scanner (http://www.iss.net/) de SAFESuite. Esta última es interesante por tratarse de una herramienta que empezó siendo de libre distribución pero que cambió posteriormente a una licencia propietaria.

La funcionalidad ofrecida por Nessus y sus equivalentes propietarias puede diferir en algunos puntos, generalmente relacionados con la generación de informes (muy

elaborado en ISS y menos elaborado en Nessus), pero sin embargo todas son equiparables en cuanto a capacidad de testeo de vulnerabilidades. De hecho, en pruebas realizadas sobre ambas, Nessus queda en los primeros puestos, en lo referente a vulnerabilidades (reales, no falsos positivos) detectadas en sistema y perfectamente comparable (e incluso mejorando) a sus análogos propietarios, no en vano ha sido elegida en algunas comparativas realizadas por profesionales de la seguridad como la mejor herramienta para realizar auditorías.

Además, algunos de los problemas históricos de estas herramientas, que hacen que la compañía que las realice pierda credibilidad, es la falta de aparición de nuevas versiones a la hora de aparecer nuevas vulnerabilidades. Esto ha tenido lugar, por ejemplo, con ISS, y es fruto de los problemas que pueden afrontar las compañías para destinar recursos a investigación a desarrollo. Es por ello una gran ventaja la posibilidad de que las compañías que dependan de estas herramientas para detectar fallos recientes en sistemas de gran envergadura puedan subcontratar el mantenimiento o actualización de dichas herramientas a terceras compañías en caso de que el fabricante original no esté a la altura de sus necesidades.

6.2.2. Auditorías de caja blanca

En el caso de herramientas de auditoría de caja blanca se pueden encontrar pocas en el campo del software libre, una de las más conocidas sería Tiger, un desarrollo basado en la herramienta COPS de Dan Farmer. Esta herramienta fue desarrollada en 1993 por la Universidad de Texas y, aunque no está siendo mantenida de forma activa actualmente, los chequeos que realiza siguen siendo útiles para sistemas UNIX. En el campo de aplicaciones propietarias se podría hablar de System Scanner de SAFESuite, cuyo objetivo es el mismo, la realización de baterías de pruebas sobre sistemas (que se podrían realizar de forma manual) con una generación integrada de informes. Contrastando ambas herramientas, la funcionalidad que ofrecen es muy similar y, si bien se podría esperar que la herramienta propietaria (y supuestamente actualizada) realizara chequeos en sistemas UNIX actualizados a los avisos de vulnerabilidades enviados por fabricantes, la experiencia demuestra que no es así. Al final, ambas herramientas integran una batería de pruebas que intentan determinar configuraciones incorrectas y permisos defectuosos en sistemas UNIX, evidentemente, este tipo de problemas no han variado mucho en el tiempo y no cambian tampoco para sistemas nuevos como GNU/Linux. La gran ventaja, a la postre, de Tiger frente a System Scanner es que, tratándose de software libre, con el paso del tiempo el testigo puede ser recogido por otra persona que mantenga al día (y añada) nuevos chequeos a la batería de pruebas.

6.3. Aplicaciones de detección de intrusos

Una tarea también necesaria a la hora de exponer servicios a Internet es la posibilidad de poder reconocer cuando un sistema se ha comprometido o se están realizando intentos para lograr dicho compromiso. Estos intentos pueden ser escaneos de puertos sobre un sistema o un rango de ellos, intentos de ataques de fuerza bruta de contraseñas, subversión de aplicaciones interactivas, etc.

Para llevar a cabo esta tarea se puede hacer uso de sistemas de detección de intrusión o “Intrusion Detection Systems” (IDS). Estos sistemas se pueden dividir básicamente en dos categorías: basados en host o basados en red. Los basados en host realizan un análisis del sistema “por dentro” para determinar si un intruso ha accedido o a intentado acceder, tareas habituales de estas herramientas son el análisis de los registros del sistema, análisis de los ficheros para detectar modificaciones de los mismos (comprobando la integridad del sistema) o análisis de los procesos ejecutándose en el sistema. Los detectores de intrusos basados en red analizan el envío y recepción de paquetes a sistemas finales en busca de patrones que puedan considerarse ataques remotos contra sistemas internos. Estos últimos no necesitan saber qué servicios se están ofreciendo en los sistemas finales, siendo capaz de detectar ataques contra servicios que no se están ofreciendo.

6.3.1. Detección de intrusos a nivel de red

En el campo de la detección de intrusión a nivel de red la gran estrella en el mundo de software libre es Snort (http://www.snort.org), un sistema ligero que actúa como sniffer de red y puede cotejar el tráfico que pasa por la red con reglas predefinidas con ataques “tipo” e incluso con ataques genéricos (sobrecargas de buffer, escaneos de puertos, etc..). Las funcionalidades de detección que ofrece son exactamente las mismas que las ofrecidas por sus contrincantes propietarios, como por ejemplo RealSecure. De hecho, Snort ha sido utilizado en la realización de sistemas de detección de intrusos “appliances” (es decir, sistemas con configuración mínima que se enchufan a la red y proporcionan de forma inmediata una funcionalidad determinada) y en comparativas con productos similares de otros fabricantes (por ejemplo Nokia) lucha codo con codo por el primer puesto siendo, con mucho, el más rápido de ellos. Sin embargo la característica que se ofrece en otros IDS y no está tan avanzada en Snort es la capacidad de integrar diversas sondas en una única consola y la existencia de consolas gráficas intuitivas que puedan ofrecer, en tiempo real, los avisos a operadores. Estas carencias parecen empezar a mitigarse con la aparición de RazorBack (http://www.intersectalliance.com) y demarc (http://www.demarc.org/) que son dos interfaces a Snort, el primero es un interfaz desarrollado para GNOME (uno de los dos escritorios libres disponibles para GNU/Linux) y el siguiente un interfaz que hace uso

de una base de datos MySQL y Apache para presentar una consola del sistema de detección de intrusos vía WWW.

Otras herramientas de este grupo son:

MOM

Es muy potente y compleja y permite vigilar redes enteras. Por ello no es muy adecuado si sólo se desea vigilar un único sistema. Se basa en un proceso principal que recibe información de procesos clientes distribuidos en distintas máquinas. La última versión en el momento de escribir este documento es MOMv3 y puede obtenerse en http://www.biostat.wisc.edu/~annis/mom3/ (http://www.biostat.wisc.edu/~annis/mom3/).

"The Hammer Project"

Es un conjunto de herramientas Java que pueden distribuirse entre varios sistemas y con ello permite identificar ataques que afecten simultáneamente a varios de ellos. De otra forma sería muy difícil relacionar ataques a distintas máquinas entre sí, dado que los logs de cada una son independientes. Estas herramientas se distribuyen bajo licencia GPL y en el momento de escribir este documento puede obtenerse la versión 3.4 en fuentes o binarios en http://www.csds.uidaho.edu/~hummer/ (http://www.csds.uidaho.edu/~hummer/).

Deception Toolkit: dtk

Se trata de una herramienta que instala en el sistema servicios falsos que simulan a los reales, al tiempo que modifica las características de servicios legítimos en el sistema para engañar a posibles atacantes. Puede obtenerse esta aplicación y más información sobre ella en su web: http://all.net/dtk (http://all.net/dtk/).

tcplogd

Detecta rastreos realizados de forma 'silenciosa', es decir aquellas que usan alguna técnica especial como conexiones medio abiertas para que sean más difíciles de detectar. Algunas herramientas que realizan este tipo de rastreos son NMAP, QueSo y Saint. Puede obtenerse esta herramienta en http://kalug.lug.net/tcplogd/ (http://kalug.lug.net/tcplogd/)

Shadow

Al igual que tcplogd, Shadow detecta rastreos silenciosos. Es fácil de instalar y suele dar buenos resultados. Está formado por dos partes (al menos) un sensor situado fuera del cortafuegos y un analizador de logs que se debe instalar en una

máquina interna de la red. Puede obtenerse más información en http://www.nswc.navy.mil/ISSEC/CID/ (http://www.nswc.navy.mil/ISSEC/CID/).

HostSentry (proyecto Abacus)

Esta herramienta busca comportamientos extraños al entrar en el sistema, como hacerlo fuera de las horas de trabajo, desde lugares no habituales, etc. HostSentry se distribuye bajo licencia GPL y puede obtenerse en http://www.psionic.com/abacus/ (http://www.psionic.com/abacus/) junto con otras herramientas relacionadas englobadas dentro del proyecto Abacus.

6.3.2. Detección de intrusos basados en host

Dentro de las herramientas de detección de intrusos basadas en host podemos hablar, en primer lugar, de comprobadores de integridad. Se trata de programas que comprueba las modificaciones realizadas en ficheros y directorios en un sistema en producción. En una primera pasada se utiliza para generar información sobre éstos en una base de datos, y posteriormente podrá comprobar y avisar de cualquier diferencia (incluso borrados y añadidos). Ejecutado de manera regular permite encontrar cambios en ficheros críticos que podrían haber tenido lugar por la entrada de un "intruso". Habitualmente se marca en la base de datos tanto los permisos y usuarios de los ficheros como un código de redundancia cíclica (CRC) que permite detectar modificaciones de éstos. El prototipo de estos comprobadores de integridad es Tripwire (ftp://coast.cs.purdue.edu/pub/COAST/Tripwire/) realizado por Gene Kim y Gene Spafford que se trata de software “casi-libre”, ya que limita la posibilidad de cobrar por modificaciones realizadas al mismo. Sin embargo en el mundo del software libre existen varios sustitutos para Tripwire: integrit (http://integrit.sourceforge.net/), aide (http://www.cs.tut.fi/~rammer/aide.html), L5 (ftp://avian.org/src/hacks), y samhain. Integrit, por ejemplo, es un sistema actualizado de chequeos de integridad que incluye el uso de algoritmos nuevos de generación de hashes como SHA-1 de 160 bits, y salida en XML. Samhain tiene características que le permiten ejecutarse de forma indetectable en los sistemas (usando estenografía), puede utilizarse con registro en servidores remotos y también incorpora algoritmos novedosos para realizar los hashes como TIGER y Rijndael para el cifrado de la base de datos.

Otra herramienta más sencilla pero útil para detectar intrusiones es chkwtmp que analiza el resultado de wtmp e informa de entradas borradas.

Por último puede mencionarse chkrootkit. Esta pequeña herramienta es capaz de detectar un gran número de “rootkit” realizando comprobaciones en busca de binarios del sistema que hayan podido ser modificados. Se distribuye con las fuentes bajo una

licencia libre poco restrictiva. Puede obtenerse este software y más información sobre chkrootkit en http://www.chkrootkit.org/ (http://www.chkrootkit.org/).

6.3.3. Herramientas integradas

Al margen de las herramientas que comprueban la integridad del propio sistema, en lo que a herramientas de detección de intrusos basadas en host propiamente dichas, el software libre no ofrece, todavía, herramientas integradas comparables a las herramientas CyberCop Monitor, RealSecure o System Scanner de ISS. Estas herramientas propietarias suelen ofrecer un número elevado de características (comprobaciones de integridad, análisis de logs, etc.) para las que no existe un competidor directo que sea software libre. Sin embargo, sí se puede hablar de múltiples herramientas para detección de intrusos basadas en hosts que cumplen ciertas (pequeñas) funciones. Algunas de las más interesantes son :

Existe, incluso, programas que se comunican con el núcleo del sistema operativo para servir como sistema de detección de intrusos. Algunos son fruto de investigaciones realizadas como parte del trabajo de las universidades, como imsafe (http://imsafe.sourceforge.net/) (que analiza el comportamiento de procesos y contrasta éstos con perfiles de ataques) o Eye on Exec (http://www.cs.uni-potsdam.de/homepages/students/linuxer) (que permite también seguir la evolución de ejecución de los procesos). Más ambicioso, sin embargo, es el proyecto Linux Intrusion Detection System (http://www.lids.org/) que es una modificación al núcleo de Linux que incorpora capacidades de bloqueo de ficheros, reglas de cortafuegos, etc..

6.4. Aplicaciones finales: servidor web, correo, DNS, etc.

Evidentemente, cuando se ofrece un servicio al exterior, este servicio está soportado, además de por toda la infraestructura de comunicaciones y de un sistema operativo final, por una aplicación en sí. Son las aplicaciones las encargadas de resolver las peticiones, pudiendo ofrecer distintos grados de interactividad. En éste área, si cabe, se pueden observar más fácilmente las ventajas que el software libre ha ofrecido frente al software propietario.

Podemos tomar como ejemplo prototipo el servicio de WWW en los servidores en Internet. De un lado, en el campo del software libre, el servidor más desarrollado (aunque no es el único libre) es Apache. De otro, en el campo de software propietario, los servidores más extendidos son el servidor iPlanet de Netscape y el Internet

Information Server de Microsoft. Los hechos del último año hablan por sí sólos a la hora de comparar la seguridad de estos servidores, se han detectado importantes vulnerabilidades en los servidores de Microsoft (tanto en su versión para Windows NT 4.0 como para Windows 2000) que culminaron con la aparición de dos gusanos distintos, siendo el más malicioso “Code Red”. Estas vulnerabilidades, detectadas por compañías ajenas a Microsoft mediante ensayos en laboratorio, implicaban ejecución directa de código en el sistema operativo (vulnerabilidad UNICODE e ISAPI), el origen de éstas era, por un lado, una inadecuada programación de entrada unido a la relación directa entre servidor y sistema operativo, por otro una inadecuada programación que sufría de una sobrecarga de buffer. En el servidor iPlanet el propio fabricante también ha detectado vulnerabilidades de sobrecarga de buffer. Sin embargo, las vulnerabilidades detectadas durante el mismo tiempo en Apache han sido resultado de la auditoría de código y han sido “supuestas” por cuanto los desarrolladores sospechaban que podrían, a la larga, suponer un problema de seguridad.

Si se comparan la evolución de estos servidores se tiene tres posiciones distintas:

- Errores detectados mediante prueba y error por compañías externas.
- Errores detectados (quizás por auditorías de código o problemas observados) por la propia compañía.
- Errores detectados por auditoría de código por una comunidad abierta.

Claramente, esta tercera opción encontrará los problemas posiblemente antes de que aparezcan (aunque esto es un factor que dependerá de la calidad de desarrollo), mientras que en la primera posibilidad los problemas están encerrados en una “caja de gusanos” que puede abrirse en cualquier momento. De nuevo, la seguridad depende de un grupo de personas que activamente monitorizan el desarrollo, pero ¿de donde salen? En el software propietario serán los recursos que el fabricante dedique (o no) a esta tarea, en el software libre dependerá de los recursos puestos en marcha por los propios usuarios del código, pudiendo variar en el tiempo más fácilmente que en el anterior caso.

Dejando al margen los problemas de seguridad específicos de servidores de WWW se pueden considerar otros servicios que puedan ser interesante ofrecer. No se entrará al detalle en cada uno de ellos, sin embargo.

Existen implementaciones basadas en software libre también de todos los servicios de Internet: RADIUS, LDAP, Correo, News, DNS, FTP, etc. En algunos casos existen servicios menos desarrollados y que ofrecen menos funcionalidades frente a sus equivalentes propietarios, como es el caso de los servidores de Radius o LDAP. En otros, como es el servicio de correo y DNS, las implementaciones de referencia son

totalmente libres y son, sin lugar a duda, las más extendidas, probadas y analizadas y, en consecuencia, las más seguras.

6.5. Defensa ante ataques de denegación de servicio

Un ataque de denegación de servicio (DoS) es, en pocas palabras, cualquier ataque que impida a un servidor ofrecer cualquiera de sus servicios de forma habitual. Por ejemplo si se trata de un servidor de páginas web un ataque DoS provocaría que los visitantes no pudieran acceder a dichas páginas.

Este tipo de ataques se han hecho tristemente famosos por los casos ocurridos recientemente y que han recibido una gran atención por parte de la prensa. En particular tuvo una gran relevancia un ataque realizado a los principales portales americanos incluyendo Yahoo, la CNN, etc.

A grandes rasgos, un ataque DoS se consigue enviando una cantidad inmensa de peticiones al servidor que se desea atacar. De esta forma el servidor se desborda y no es capaz de atender las *peticiones legítimas*. Para conseguir realizar un número muy grande de peticiones los atacantes se las arreglan para que un número muy grande de ordenadores participen en el ataque. Además se suelen emplear peticiones especiales que provocan que el servidor se quede en un estado de espera o inestable con lo que se multiplica el poder dañino de cada petición.

No existen soluciones software perfectas ante este tipo de ataques. Las mejores medidas que pueden tomarse son disponer de medios para reaccionar rápidamente ante un ataque (probablemente con ayuda de operadores de la red) y emplear software que no disponga de fallos de seguridad que faciliten este tipo de ataques. En este último sentido gana enteros el software que dispone de código abierto y por tanto puede ser auditado (muchos ojos ven más de dos), si además el software es libre la posibilidad de modificación por muchas partes ha demostrado en la práctica que se reduce el tiempo que un fallo de este tipo tardará en ser arreglado.

7. Autenticidad, integridad, cifrado y disponibilidad de la información

En general, los mecanismos necesarios para soportar autenticidad e integridad de la información a todos los niveles. En el caso de la autenticación será necesario por un lado permitir que usuarios se autentiquen a sí mismos y comprueben la autenticidad de

otros, pero también debe permitirse que sistemas informáticos (servidores) se autentiquen entre sí y ante una petición de un usuario. La integridad de la información se podrá entender como la necesidad de que la información no sea modificada en tránsito, para lo que será necesario establecer los mecanismos criptográficos adecuados, o la necesidad de que la información que reside en los sistemas no se degrade o pueda ser manipulada.

7.1. Sistemas de autenticación

Los sistemas de autenticación disponibles en los sistemas de software libre son muy variados, partiendo desde la autenticación local original de UNIX basada en ficheros en el servidor local, a implementaciones de servicios de directorios como NIS, RADIUS o LDAP pasando por implementaciones de sistemas de autenticación en dominios con tokens como Kerberos. En este campo, existen implementaciones libres de todos los servidores de autenticación con funcionalidades equivalentes a sus versiones propietarias. En algunos casos las versiones disponibles en software libre soportan funcionalidades más avanzadas que las que se pueden encontrar en el software propietario.

Por ejemplo, los sistemas UNIX ya disponían de implementaciones de Kerberos (un sistema de autenticación desarrollado en el MIT) mucho antes de que éste fuera adaptado y modificado para su uso en Windows 2000 porque consistía un sistema mejor al ya implementado en Windows NT 4.0. Por otro lado, los servidores de RADIUS propietarios como NavisRadius de Lucent o SteelBelted Radius, ofrecen funcionalidades que aún no están disponibles en las versiones de Radius mantenidas por la comunidad del software libre: Cistron y Livingston, pero que sí que lo están en la versión en fase de desarrollo (*beta*) de FreeRadius, un servidor de Radius mejorado basado en Cistro.

7.2. Firma digital

Cuando se habla de firma digital en este contexto nos referimos a la capacidad de aplicar mecanismos criptográficos para asegurar la identidad del autor del documento y, si fuera necesario, asegurar que un documento no pueda ser leído más que por un grupo cerrado de usuarios. Estos mecanismos de firma digital, basados en sistemas de claves públicas y claves privadas, tuvieron su impacto en la comunidad Internet con la distribución del software PGP (Pretty Good Privacy).

Sin embargo, tras la distribución, inicialmente sin restricciones, de este software, la compañía que posteriormente se creo alrededor de este producto ha ido limitando la

posibilidad de distribución del mismo hasta el punto de que, hoy por hoy, este producto no se distribuye con código fuente. Aún así, PGP se ha convertido en el abanderado de las comunicaciones cifradas a través de correo electrónico.

¿Existe una alternativa a este software imprescindible en el mundo del software libre? La respuesta es GNU Privacy Guard, también conocido como GNUPG o, simplemente, GPG. Que implementa las mismas características que PGP, aunque no se incluyen algunos algoritmos de generación de hashes y cifrado debido a problemas de patentes (pero es posible cargarlos mediante extensiones). Sí se incluyen, desde sus versiones iniciales, para cifrado: 3DES, Blowfish, Twofish y CAST5, así como distintos algoritmos para hashes como MD5 y SHA1, se han incluido nuevos algoritmos a lo largo del desarrollo de este producto. Igualmente, GPG es compatible con el estándar OpenPGP (RFC2440).

En cuanto a capacidades de integración y usabilidad, GPG está integrado en la mayoría de lectores de correo modernos desarrollados en software libre, y dispone de interfaces gráficos (como GPGP) para facilitar la generación de claves y el tratamiento de los anillos de claves en general.

7.3. Autoridades de certificación

El mundo de las autoridades de certificación ha sufrido en los últimos tiempos un cierto auge por la necesidad de desarrollar infraestructuras de clave pública para, sobre estas infraestructuras, desarrollar nuevos servicios: single sign-on, servidores de web seguros, redes privadas virtuales... En este área, fabricantes como Baltimore y Verisign se han posicionado rápidamente ofreciendo soluciones propietarias para el desarrollo de estas infraestructuras de clave pública, y sus soluciones son las usadas, mayoritariamente, para estos despliegues.

El software libre, en estos aspectos está algo menos desarrollado que estas iniciativas propietarias. Existe una alternativa para la generación de certificados digitales, que es OpenSSL (http://www.openssl.org). Este software es, de hecho, utilizado por muchos fabricantes de "cajas negras" de red, que utilizan habitualmente una interfaz de administración vía HTTP para la rápida configuración de estos elementos que implementan funciones de aplicación basadas en hardware (típicamente cortafuegos, balanceadores de carga, sistemas de caché). Estas mismas interfaces, para poder ofrecer una conexión segura, generan certificados autofirmados que les permiten cifrar la comunicación con un navegador.

En cualquier caso, el despliegue de una infraestructura utilizando OpenSSL es laborioso por cuanto no dispone de todas las capacidades de su equivalentes

propietarios. Actualmente, si bien existe el proyecto OpenCA (http://www.openca.org) para ofrecer un sistema de desarrollo de infraestructuras de clave pública basado en software libre, estas iniciativas son aún inmaduras y no han llegado aún a un nivel suficiente para ser usadas de forma intensiva.

7.4. Comunicaciones cifradas

En el caso de comunicaciones cifradas entre sistemas podemos hablar de implementaciones basadas en software libre diversas para según qué tipo de comunicaciones sean consideradas.

Si se consideran comunicaciones directas entre dos sistemas, con capacidades de cifrado de tráfico, los proyectos OpenSSH y OpenSSL ofrecen las mismas capacidades que sus equivalentes propietarios (SSH de F-Secure y SSH Technologies y las diversas implementaciones de SSL existentes). En este sentido, es posible, con por ejemplo OpenSSL, tunelizar conexiones de servicios determinados (telnet, ftp, pop3) para asegurar que la información intercambiada entre éstos no pueda ser interceptada y descifrada en tránsito.

En implementaciones de creación de túneles para la generación de redes privadas virtuales se puede hablar básicamente de una serie de tecnologías: IPsec y túneles punto a punto (que suponen una extensión a la tecnología IP para establecer circuitos cifrados extremo a extremo). La diferencia del primero, y el hecho de diferenciarlo de los túneles, es que la implementación de éste es obligatoria en IPv6 y tiene unas funcionalidades, a priori, mucho más potentes que las disponibles en la creación de túneles entre sistemas finales. Si bien el software libre tiene herramientas desarrolladas equivalentes a las propietarias (y en algunos casos mejores) para la generación de túneles punto a punto, no existe aún una implementación completamente desarrollada de la tecnología IPsec y parte del problema reside en la necesidad de desarrollar, conjuntamente con la implementación, una infraestructura de clave pública (que es la que utiliza IPsec como base para el establecimiento de las conexiones cifradas).

En esta última área es importante destacar el proyecto FreeSwan (http://www.freeswan.org), que ofrece una implementación (no totalmente a prueba de fallos) del protocolo IPsec para GNU/Linux. Sin embargo, esta tecnología está por detrás de la tecnología propietaria desarrollada para IPsec por otros fabricantes. Esto se debe en gran medida a que la evolución natural de los sistemas cortafuegos (basados en sistemas propietarios ya asentados) ha ido dirigida a implementar funcionalidades de VPN para facilitar la conexión remota de usuarios corporativos. Así, la mayoría de las soluciones propietarias de cortafuegos de filtrado de paquetes disponen de versiones con soporte de VPNs y clientes específicos.

Sin embargo, FreeSwan sí que ha sido utilizado como implementación de referencia para la elaboración de algunos de los RFCs relacionados con IPsec y participa en los foros de estandarización que los fabricantes están obligados a llevar a cabo.

7.5. Sistemas de alta disponibilidad

Una funcionalidad muy apreciada en los sistemas, tanto en sistemas intermedios de comunicaciones como en sistemas finales, es su capacidad de poder mantener una alta disponibilidad basada en una redundancia de equipos. Es decir, que se pueda dimensionar el sistema sin puntos únicos de fallo de forma que, por ejemplo, en caso de fallo de un servidor, sea otro servidor el que ofrezca los servicios que prestaba aquél. Está siendo, cada vez más común, la utilización de sistemas redundantes que podrán estar configurados en modo activo-pasivo (también conocido como *failover* y *hot-standby*), es decir, que un sistemas realiza las funciones y, en caso de fallo, un sistema de backup las asume. O que podrán estarlo en activo-activo, en el que una serie de sistemas, por ejemplo A o B, llevan a cabo servicios distintos A' y B' respectivamente pero que, en caso de fallo de un sistema (por ejemplo B), el otro (A) asume todos los servicios del cluster (A' y B').

La implementación de estos mecanismos de alta disponibilidad se viene haciendo de dos maneras: mediante técnicas de rutado virtual (protocolo VRRP) implementados a nivel de sistema operativo o mediante mecanismos software implementados por encima del sistema operativo que detectan los fallos y toman las medidas necesarias.

Desgraciadamente, las implementaciones en la actualidad de mecanismos de alta disponibilidad como el proyecto linux-ha (http://www.linux-ha.org) o el software de alta disponibilidad Piranha (ftp://ha.redhat.com/pub/ha), están muy por debajo de los mecanismos de alta disponibilidad implementados en productos propietarios. En el área de alta disponibilidad con rutado virtual, los sistemas Nokia basados en IPSO (una versión propietaria basada en FreeBSD) están muy por delante. Igualmente, existe una serie de software propietario para implementar alta disponibilidad de servicios a alto nivel (servicios de correo, bases de datos, correo, cortafuegos, etc.) basados en aplicaciones que monitorizan a los sistemas, como puedan ser Qalix, Legato Full Time Cluster o StoneBeat, que son, hoy por hoy, muy superiores a los proyectos, aún en desarrollo de software libre.

8. Conclusiones

Realizado todo este análisis de productos y tecnologías de seguridad, cuyo objetivo es dar una visión general de las capacidades en este área del software libre, es necesaria hacerse la pregunta final: ¿es mejor utilizar software libre para los productos de seguridad?

La respuesta es ... depende. Que, aunque pueda parecer una respuesta ambigua, está en realidad suficientemente fundamentada. Se ha podido ver, a lo largo de todo este desarrollo, las distintas capacidades, ventajas y desventajas del software libre frente al software propietario. Igualmente, se ha podido ver que, en determinadas áreas, hoy por hoy, no es viable basar una solución de seguridad en software libre y va a ser necesario acudir a soluciones propietarias por no estar el primero aún suficientemente desarrollado.

Sin embargo, sí que es posible adivinar que el software libre está, en determinadas áreas, compitiendo codo con codo con las soluciones propietarias existentes. La situación ha ido cambiando a medida que las distintas soluciones desarrolladas se han demostrado competitivas y han ido siendo aceptadas. Es por esto que es de esperar que, en aquellas áreas en las que el software libre aún no alcanza al software propietario, la situación llegue a igualarse (e incluso invertirse) pasado un cierto tiempo.

Es importante observar que, en las áreas investigadas no se produce un efecto tangencial. Es decir, como quiera que una determinada solución no llega al nivel de otra implementación, y como el desarrollo en ambas a lo largo del tiempo es inicialmente el mismo, la solución más pobre nunca puede llegar a alcanzar a la más desarrollada porque ésta está siempre en cabeza. Sin embargo se han visto áreas en las que el software libre iguala al software propietario y otras en las que incluso lo supera.

En la tabla adjunta se muestra, de forma resumida, las distintas áreas estudiadas en este trabajo y la valoración que pueden recibir las soluciones dividiendo en software libre y software propietario. La calificación se ha hecho de una forma, en gran medida, subjetiva, basándose en la apreciación de los autores. Para esta calificación se ha utilizado una nota expresada de la A (mejor) a C (pero). Una A significa que un área está muy desarrollada, una B que implementa la funcionalidad suficiente para ser operativa (pero no capacidades que la puedan convertir en una tecnología plenamente desarrollada) y una C que aún está en desarrollo. Para tener una mayor flexibilidad en la calificación se han añadido ’+’ y ’-’ indicando una mejora, o degradación, dentro de una misma calificación.

Por otro lado, independientemente del ritmo de crecimiento del software, del lado de la seguridad, las ventajas ofrecidas por el software libre son evidentes frente a las alternativas propietarias. Máxime en determinados entornos en los que una persona no

se puede "fiar" de aquella compañía que le vende la solución o no puede depender de la seguridad "garantizada" por un determinado producto que no tiene forma de demostrar.

Por tanto, si bien el software libre en la actualidad tiene una cobertura desigual de las distintas necesidades de seguridad de una empresa o corporación, éste es, definitivamente, una apuesta de futuro provechosa en aquellas áreas aún no desarrolladas y una oportunidad real e inmediata en las demás áreas para utilizar soluciones equivalentes a las propietarias con:

- un menor coste
- unas mayores garantías de seguridad, debido a la posibilidad de auditar el código en uso
- una mayor flexibilidad en la adaptación e integración, gracias a la posibilidad de modificar dicho código
- la posibilidad del mantenimiento asegurado de una solución de seguridad con independencia del origen del producto en sí.

Tabla 1. Comparativa de la situación actual del software libre en el área de la seguridad

Área	Situación sw libre	Situación sw propietario
Sistema operativos	A	A
Cortafuegos personales	B+	A
Cortafuegos de filtrado	A	A+
Cortafuegos de aplicación	C	A
Herramientas de Auditoría Externa	A+	A
Herramientas de Auditoría Interna	B	B
Detección de intrusos	A	A
Sistemas de autenticación	A	A
Firma digital	A	A
Autoridades de certificación	C+	A
Comunicaciones cifradas	B+	A
Alta disponibilidad	C+	A+

9. Bibliografía

9.1. Libros

Linux Máxima Seguridad

Autor: Anónimo, Editorial: Prentice Hall

Un libro muy completo sobre la seguridad en Linux en general. Quizá se queda un poco corto en el tratamiento de herramientas, sobretodo las orientadas a usuarios que no sean administradores.

9.2. Documentos y tutoriales

Linux Security-HOWTO

Documento que indica a un administrador como asegurar su sistema GNU/Linux. Puede obtenerse en http://www.linuxdoc.org/HOWTO/Security-HOWTO.html (http://www.linuxdoc.org/HOWTO/Security-HOWTO.html).

Linux Administrator's Security Guide

Una guía completa (aunque algo obsoleta ya, no ha sido actualizada desde 1999) que trata todos los aspectos relacionados con la seguridad en Linux, desde la seguridad en el núcleo a implementaciones en VPN. Puede obtenerse en http://www.securityportal.com/lasg/ (http://www.securityportal.com/lasg/). El autor está manteniendo como versión actualizada de esta guía la "Linux Security Knowledge Base" disponible en http://www.securityportal.com/lksb/ (http://www.securityportal.com/lksb/).

Linux IPCHAINS-HOWTO

Excelente documento que forma parte del "Linux Documentation Project" e incluye gran cantidad de información útil para los administradores de cortafuegos. Puede obtenerse en http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html (http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html)

Linux Gets Stateful Firewalling

Excelente artículo que describe las novedades en cuanto a filtrado de paquetes de la versión 2.4 del núcleo de Linux. Puede obtenerse en http://www.securityportal.com/cover/coverstory20010122.html (http://www.securityportal.com/cover/coverstory20010122.html).

9.3. Sitios web de seguridad y software libre

Free Software Foundation: www.fsf.org

Página web de la Fundación del Software Libre.

GNU www.gnu.org

Servidor principal del proyecto GNU, que ofrece los componentes básicos del sistema operativo GNU/Linux así como las herramientas de compilación.

Linux security: www.linuxsecurity.com

Servicio dedicado a todos los aspectos de seguridad en el mundo GNU/Linux, con avisos y noticias, herramientas, etc.

Security Portal: www.securityportal.com

Servidor con artículos relacionados con la seguridad en general, noticias e informes sobre tecnologías.

Security Focus: http://www.securityfocus.com

Servidor orientado a la seguridad en todos los sistemas operativos y sus aspectos. Hospeda la base de datos de vulnerabilidades Bugtraq y su lista de correo, uno de los medios que fomenta la “popularización” de los problemas de seguridad que afectan a todos los sistemas.

Linux firewall and security site: http://www.linux-firewall-tools.com/linux/

Información general sobre cortafuegos en Linux y herramientas para su control y configuración.