Seguridad informática y software libre.

Estructura de Hispalinux

Jorge Ferrer
Hispalinux

Javier Fernández-Sanguino
Hispalinux

En este documento se estudiarán las ventajas ofrecidas por el software libre en el área de la seguridad informática, comparando éstas con las ofrecidas, hoy en día, por el software propietario.

Tabla de contenidos

Seguridad informática y software libre.

por Estructura de Hispalinux

por Jorge Ferrer

por Javier Fernández-Sanguino

v 1.0 para el documento general

En este documento se estudiarán las ventajas ofrecidas por el software libre en el área de la seguridad informática, comparando éstas con las ofrecidas, hoy en día, por el software propietario.

1. Introducción

1.1. 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, etc.) 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 de una forma inmediata como si se tratara, en lugar de un banco, de 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.

1.2. ¿Por qué 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.

1.2.1. 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.

1.2.2. 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:

- Introducción de virus o troyanos por la descarga y ejecución de ficheros en servidores, en principio, confiables, por parte del usuario. El efecto de distribución puede ser, incluso, involuntaria si se hace uso de sistemas de archivos compartidos. En el caso de los virus el efecto destructivo se hará patente más pronto o más tarde. La instalación de troyanos puede, sin embargo, pasar desapercibida.
- Difusión de virus por correo electrónico. Lograda gracias a la malversación por parte del virus del programa utilizado como lector de correo (que lo ejecuta automáticamente sin intervención del usuario) o porque el usuario activa el virus inadvertidamente creyendo que se trata de otra cosa. Su efecto pernicioso es, además del destructivo habitual de un virus, la distribución a las direcciones conocidas convertiendo su propagación en exponencial.
- 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 sistemas.

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.

1.2.3. 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.

1.3. 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 achacarse, 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. El Software Libre y la seguridad informática

2.1. ¿Qué es el Software Libre?

Para entender la situación de este tipo de software con respecto a su uso en seguridad informática es imprescindible describir, en primer lugar, a qué se refiere este documento cuando hace referencia a “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, éste no tiene por qué perder sus derechos sobre el mismo. No se incluye, por tanto, en esta definición software en el “domino público” (aquél para en el que el autor ha cedido todos sus derechos).

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. 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 totalidad, 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 del código del program.
- 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 que puede sobrevenir en la realización del software (presentado anteriormente), el software libre protege a sus usuarios con una serie de mecanismos determinados. Entre estos:

- 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.

Las auditorías de código son, por tanto, posibles o no en determinados sistemas operativos en función de la publicidad dada a su código fuente. Sin embargo, no basta con decir qué se puede hacer una auditoría del código, 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 realizado por terceras entidades. 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 sugerir implementaciones de las modificaciones sugeridades que se podrán incorporarán rápidamente en el código auditado.

2.3. Desventajas del software propietario

En primer lugar, es necesario aclarar que 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.

Con respecto a la seguridad, las mismas garantías que ofrece el software libre en el mundo de la seguridad son problemas que se le pueden achacar al software propietario. 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 cumplan 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 notar 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).

2.4. 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 están comenzado a soluciones. El caso la difusión de troyanos se limita mediante el uso de técnicas de firma digital para garantizar la inviolabilidad del código o binarios transmitidos. Es frecuente que algunos autores de software libre al distribuir el código indique también información (sumas MD5 firmadas) que permitan garantizar la integridad del código descargado. Asimismo, las distribuciones del sistema operativo, como Debian o RedHat, han incorporado a lo largo del año 2001 soluciones de firma digital para la distribución de código fuente y binario de forma que el usuario pueda garantizar la integridad del mismo tras una descarga.

De igual forma, los problemas de evolución futura empiezan a quedar resueltos con un cambio de paradigma por parte de las compañías de software. Se trata del cambio de un modelo de negocio en el software que pasa a enfocar el negocio orientado a el cobro de la realización de servicios en lugar del cobro a la utilización de productos. Ya se observan, en el mundo de software libre, compañías que contratan a personal cualificado para hacer mejoras sobre proyectos libres para cubrir sus propios intereses y ofrecen soporte de productos de software libre. Estas compañías, a diferencia de la orientación propietaria previamente presentada, siguen haciendo públicas las modificaciones realizadas al código fuente.

3. Conclusiones

Es posible hacer un análisis de los distintos productos y tecnologías de seguridad disponibles actualmente. Este análisis, realizado en “El sistema operativo GNU/Linux y sus herramientas libres en el mundo de la seguridad: estudio del estado del arte”, se muestra en la tabla adjunta.

En ésta se muestra, de forma resumida, las distintas áreas estudiadas en el estudio indicado 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 (peor). 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.

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
Aplicaciones finales	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+

En vista de estos resultados ¿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 presentado, en este documento, las distintas capacidades, ventajas y desventajas del software libre frente al software propietario. En la tabla, sin embargo, se puede 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 estar el primero en una etapa aún inmadura de desarrollo.

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 por el público general. Esta aceptación ha dado lugar a un desarrollo exponencial en el que se pueda 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.

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.

Además de todas estas considearaciones *Si Vd. no puede ver ni auditar el código de sus herramientas propietarias de analisis ¿como puede estar seguro de que su software de seguridad es seguro?, ¿puede estar Vd. seguro de que no hacen mas de lo que dicen hacer?*

4. Bibliografía

4.1. Libros y artículos

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.

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

Autores: Jorge Ferrer y Javier Fernández-Sanguino

Un análisis completo de las herramientas disponibles en el mundo del software libre en el ámbito de la seguridad informática. Publicado en el Congreso de Hispalinux 2001 http://congreso.hispalinux.es (http://congreso.hispalinux.es).

4.2. 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/).

4.3. 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 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.