

El Software Libre y la seguridad informática

LINK: <http://es.tldp.org/Informes/informe-seguridad-SL/informe-seguridad-SL-html/sw-libre.html>

¿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 "dominio público" (aquel para 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](#), 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](#) y la licencia [BSD](#).

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 distribuirlos 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, sí 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 sí 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 sugeridas que se podrán incorporar rápidamente en el código auditado.

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).

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 comenzando a tener soluciones. El caso de 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 indiquen 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 enfocarse en el negocio orientado a 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.