Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Shared Source no es Open Source (9369 lectures)

Per **Benjamí Villoslada**, ***Benjami*** (http://weblog.bitassa.net)

Creado el 13/02/2003 21:43 modificado el 04/03/2003 04:08

Existen grandes diferencias entre la licencia Shared Source que ahora ofrece Microsoft y el Open Source de toda la vida.

***Actualización 04-03-2003**: Sendmail tiene un problema de seguridad potencialmente tan peligroso como el del gusano SQL Slammer para SQL Server.*

La necesidad de una nota de prensa aclarándolo surgió en la lista socios[1] de Hispalinux[2], al ver que algunos medios de comunicación estaban poniendo todas las aperturas de código en el mismo saco.

En un recuadro final, usamos el virus SQL Slammer y la vunerabilidad del CVS como un ejemplo de los resultados de cada modelo frente a los problemas graves --se tomaron estos ejemplos porque eran las vulnerabilidades importantes del momento.

Encontraréis el artículo en Hispalinux.net[3] --y también aquí, haciendo de "mirror": en PDF[4] y a partir de la siguiente página de este artículo.

Más información y debate en la reseña de Barrapunto[5].

Actualización 04-03-2003: Sendmail tiene un problema de seguridad *potencialmente* tan peligroso como el del gusano SQL Slammer para SQL Server.

En el artículo incial intenté ilustrar las diferencias de los sistemas de desarrollo a través del caso reciente de dos agujeros de seguridad en ambos modelos: SQL Server en el cerrado y CVS en el abierto. Pero podían parecer distintos, porque el primero afectó al trafico mundial en la Red. En cambio, el CVS, aunque también era grave se trataba de algo más "de informáticos", sin tanta repercusión mediática. No afectó a todos los internautas.

Pero acabo de abrir Google News y veo esta página[6]. La primera noticia es sobre un agujero de seguridad en un programa abierto muy conocido, el Sendmail. Es reciente, Google está indexando artículos que tienen pocas horas (tal como puede verse en la captura[6] antes citada) Leo la noticia en el web de Forbes[7] y es preocpante:

«The flaw in Sendmail also makes it vulnerable to high volumes of data traffic, which could allow a malicious worm program to propagate and slow down global Web traffic, much like the "SQL Slammer" attack that slowed Internet traffic worldwide in late January.»

El problema puede convertirse en algo parecido a SQL Slammer.

Pero casi simultáneamente, en el "Panel" de KDE puedo ver el globo rojo con una admiración. Es el programa "rhn-applet-gui", el actualitzador automático de Red Hat 8. Hago clic en él y bingo, ya está disponible la actualización de Sendmail[8].

Cualquier software tiene problemas de seguridad. La diferencia está en cómo se tratan. Hay modelos de desarrollo que favorecen que se traten de una forma muy diferente. Vive le difference.

El código abierto no es un producto

Tras el anuncio de la apertura del código de algunos productos de Microsoft, Hispalinux recuerda que la esencia del movimiento de código abierto va mucho más lejos.

El código abierto no es sólo un producto opcional que se puede solicitar al fabricante, sino un modelo de desarrollo y generación de conocimiento en el que todos participan: investigadores, industria y usuarios.

Los gobiernos y empresas están adoptando software abierto por dos razones básicas:

1. Saber todos los detalles del código que administra su información.
2. Que de la industria local pueda participar en el desarrollo de productos informáticos.

En una Sociedad de la Información donde todos los sistemas están interconectados entre sí, es de suma importancia conocer hasta el último detalle del flujo de la información para garantizar que existe ninguna filtración programada expresamente --son los mecanismos conocidos como "troyanos".

También es importante examinar la consistencia de todos los mecanismos para prever y solucionar los posibles errores de funcionamiento. De otra manera, los errores fácilmente se convierten en agujeros de seguridad que los atacantes pueden explotar para conseguir información reservada.

Cuando el código está abierto, la industria local puede examinarlo para mejorarlo y adaptarlo a las necesidades de cada caso. Para crear y modificar código no hacen falta grandes infraestructuras fabriles. Con un ordenador y el conocimiento necesario, cualquier técnico preparado tiene todo lo que hace falta para crear nuevos productos o mejorar los que ya existen.

Así, para que la industria local pueda participar, basta con tener acceso al código además del permiso legal para poderlo modificar y ampliar. El resultado son nuevos productos o nuevas versiones de programas que ya existen.

La oferta de Microsoft

La oferta de Microsoft permite examinar el código (punto 1) pero no participar en la modificación, mejora y ampliación (punto 2). Los programas siguen siendo de su propiedad y sólo ellos pueden lanzar nuevos productos y versiones que los mejoren. Todo el negocio sigue estando en sus manos.

La industria local puede examinar la calidad de lo que compra, pero toda su influencia se limita a la relación cliente-proveedor de toda la vida: Reclamar calidad y saber lo que se compra.

Esta característica, mostrar el código, ya está presente en prácticamente todos los productos de consumo: Es posible desmontar el motor de un vehículo y examinar todas las piezas; ningún alimento puede estar en las estanterías de un comercio si no exhibe los ingredientes en el envoltorio. Cada componente debe estar aprobado y homologado por las autoridades locales.

Microsoft ahora muestra su código, pero también prohíbe usarlo para cualquier cosa que vaya más allá del simple hecho de contemplarlo; esto es, leer la etiqueta o abrir el motor. Este es todo el avance propuesto, algo que ya teníamos para el resto de productos de consumo. No hay ventajas, y lo que Microsoft ahora ofrece ya debía estar disponible desde el principio.

La oferta del modelo de código abierto

Los productos para la Sociedad de la Información son algo esencialmente distinto a lo que podíamos fabricar y consumir hasta ahora. Para desarrollarlos basta con tener conocimiento y ordenadores de bajo coste. No hacen falta grandes instalaciones industriales ni máquinas complicadas, ni una compleja logística de distribución, ni procedimientos de fabricación y métodos que a diferencia de los ingredientes --el resultado a la venta-- sí pueden estar más o menos ocultos. Todo el conocimiento necesario se imparte en las universidades locales y consiste en lenguajes de programación estándar. La distribución se hace a través de la Red.

En Hispalinux recordamos que el auténtico modelo de desarrollo basado en código abierto explota las ventajas de esta

nueva materia prima: El conocimiento expresado en lenguajes de programación universales y difundido en Internet.

También queremos llamar la atención sobre lo absurdo que resulta ocultar el conocimiento en el Siglo XXI.

La Humanidad ha avanzado a base de descifrar el código de todo lo que nos rodea. Desde el porqué de las estaciones hasta la tabla periódica de elementos pasando por el genoma humano --todavía pendiente de descodificación. Saber el código de las cosas nos ha permitido mejorar y adaptar las cosas a nuestras necesidades. Cuando conozcamos nuestro genoma podremos curar algunas enfermedades que ahora son mortales.

Atávicamente, la ocultación de código sólo ha servido para favorecer a una minoría privilegiada. Así, en la sociedad primitiva, las religiones iniciáticas estaban controladas por unos cuantos elegidos, los sacerdotes, que guardaban las revelaciones como un tesoro. Eran una fuente de poder y dominio que ejercían a través de lo que llamaban "magia", algo inexplicable para los no iniciados --porque desconocían el código esencial. Los elegidos debían realizar duras pruebas rituales para acceder a los misterios mayores --el código.

Descifrar el código sirvió para ver procesos naturales donde antes había "magia". Los sacerdotes se oponían tanto como podían a la investigación, porque esto significaba perder poder.

Nuestra sociedad ya no está basada en el temor a los desconocido la investigación no está penalizada. Al contrario, las patentes se concibieron para estimular la investigación y hacer público el código, puesto que es imposible patentar nada sin mostrar la fórmula y los ingredientes.

Microsoft hasta ahora ha actuado como aquellos sacerdotes primitivos, ocultando el código y prohibiendo el acceso a él. Le ha dado un gran poder, convirtiéndola en una de las empresas más influyentes del planeta.

Afortunadamente, hemos avanzado un poco y ahora es posible ver el código. Pero sigue siendo una exclusiva para elegidos: gobiernos y grandes empresas que deberán pasar por "duras pruebas iniciáticas".

El paso es comparable con el que ya se consiguió a través de las patentes. Pero en Hisapalinux proponemos ir más lejos. Proponemos la descodificación y generación de conocimiento total, sin reservas, que practican los informáticos que adoptan el modelo de desarrollo GNU.

Nuestro modelo no se basa en patentes, porque no tienen sentido en una industria basada en código abierto. Nuestro objetivo es el reaprovechamiento y la máxima difusión del código, cosa que no sería posible si estuviese patentado. Cuantos más usuarios utilicen el código, más fácil será vender otros servicios asociados: Instalación, adaptación, mejora, mantenimiento y formación. Toda la protección esperada se basa en la legislación que defiende los derechos de autor (esto es, la visibilidad) y la calidad de los servicios ofrecidos --más detalles en http://proinnova.hispalinux.es[9]

Nuestra oferta es esencialmente diferente a la de Microsoft. Creemos que sólo así es posible ofrecer de manera completa lo que los nuevos usuarios de código abierto reclaman y que la fórmula propuesta por Microsoft sólo satisface parcialmente:

1. Saber todos los detalles del código que administra su información.
2. Que de la industria local pueda participar en el desarrollo de productos informáticos.

En definitiva, proponemos la fórmula más eficaz y Universal para desarrollar la Sociedad de la Información y el Conocimiento.

Hispalinux Benjamí Villoslada © 2003
Se permite la copia y distribución por cualquier medio siempre que sea literal y se incluya esta nota.

Vulnerabilidades en Microsoft SQL Server y CVS

Estos días nos ocupan dos vulnerabilidades importantes, una en un producto propietario de Microsoft (Microsoft, SQL Serve) y otra en la herramienta CVS basada en código abierto. Son dos programas presentes en prácticamente

todos los servidores que usan los respectivos sistemas. Si la respuesta no es rápida, las consecuencias pueden ser graves. En el caso de SQL Server, el colapso de Internet. Un CVS vulnerable puede dar lugar a muchos programas de código abierto con troyanos en su interior, y las consecuencias funestas se manifestarían en una gran cantidad de productos --prácticamente todo el código abierto se desarrolla usando CVS.

En el caso de SQL Server, se trata de un agujero de seguridad descubierto seis meses atrás. A pesar de existir una solución y varios avisos, muchos servidores Windows todavía son vulnerables. Por este motivo Internet ha sufrido, entre los días 25 y 28-01-2003, el mayor colapso de los últimos 18 meses --el anterior fue provocado por otro agujero en un producto de Microsoft, el IIS. Las pérdidas son grandes, porque se dan en una Sociedad de la Información cada día más basada en el uso intensivo de la Red.

En cuanto a CVS, la vulnerabilidad se publicó el día 22-01-2003, pero el día 17 ya estaba disponible la solución. La mayoría de sistemas GNU/Linux ya estaban actualizados el día 21.

La diferencia en la respuesta está provocada por detalles clave favorecidos en cada modelo de desarrollo:

En el modelo propietario, sólo Microsoft puede preparar el parche y publicarlo. La mayoría de usuarios no pueden participar en la solución, así que es habitual que los sistemas estén administrados por usuarios con menos conocimientos, porque no tienen porqué trabajar con el código. De hecho, según Microsoft, una de las ventajas clave --nada trivial-- de sus productos está en que es posible contratar personal menos cualificado para administrarlos, reduciendo así el TCO (Total Cost Ownership). La industria local competente no está tan potenciada.

De hecho, algunos administradores competentes e informados que habían instalado el parche de julio de 2002 también eran vulnerables por incompatibilidades con otros parches posteriores. Ha hecho falta un ataque masivo --el de estos días-- para ver una solución definitiva en la sede de Microsoft --el Servicepack 3, del 17-01-2003. Las primeras soluciones provisionales para administradores de sistemas Windows vinieron del trabajo que hicieron los administradores de sistemas de código abierto, porque veían que sus servidores tampoco funcionaban correctamente debido a la cantidad de tráfico basura que generaba la vulnerabilidad de SQL Server. (Más detalles en el artículo "Lecciones del gusano MS-SQL" - http://www.hispasec.com/unaaldia.asp?id=1555[10])

En el modelo abierto, los administradores de servidores suelen trabajar con código y el asunto les interesa. Así, están al día de los problemas y participan en solucionarlos. Las mejoras y parches forman parte de su trabajo, y los consumen como algo normal en cuanto aparecen, porque están en contacto con los foros que los descubren, discuten y elaboran. Por este motivo también son más consistentes, no dependen de la pericia de pocos técnicos con acceso a un código restringido. La actualización es prácticamente inmediata y por lo tanto no son tan frecuentes --prácticamente nulas-- las situaciones de caos en la Red.

Microsoft

- Avisos del CERT:

24-07-2002:
http://www.kb.cert.org/vuls/id/370308[11]
http://www.kb.cert.org/vuls/id/399260[12]
http://www.kb.cert.org/vuls/id/484891[13]
25-01-2003:
http://www.cert.org/advisories/CA-2003-04.html[14]

Alerta de Microsoft, el 25-01-2003:
http://www.microsoft.com/technet/security/virus/alerts/slammer.asp[15]

Parche del 24-02-2003:
http://www.microsoft.com/technet/security/bulletin/MS02-039.asp[16]
Parche acumulativo (Servicepack 3) del 17-01-2003:
http://www.microsoft.com/sql/downloads/2000/sp3.asp[17]

Red Hat 8

- Aviso del CERT, 22-01-2003:
http://www.cert.org/advisories/CA-2003-02.html[18]

- Parche para Red Hat 8, "cvs-1.11.2-8", del 17-01-2003:
http://rhn.redhat.com/errata/RHSA-2003-012.html[19]

Un servidor de tantos, actualizado la madrugada del 21-01-2003:

```
$ rpm -qi cvs-1.11.2-8

Name        : cvs                           Relocations: /usr
Version     : 1.11.2                             Vendor: Red Hat, Inc.
Release     : 8                               Build Date: Thu 16 Jan 2003
08:36:08 PM CET
Install date: Tue 21 Jan 2003 05:53:26 AM CET      Build Host:
sylvester.devel.redhat.com
Group       : Development/Tools             Source RPM: cvs-1.11.2-8.src.rpm
Size        : 2778219                          License: GPL
Signature   : DSA/SHA1, Fri 17 Jan 2003 07:19:07 AM CET, Key ID
219180cddb42a60e
Packager    : Red Hat, Inc.
```