

¿Oscuridad o transparencia?

Todas las mañanas, para matar el tiempo que consume el largo camino a la facultad, me entretengo ojeando la prensa gratuita (aquí en Madrid es habitual acumular hasta tres y cuatro periódicos al entrar al metro). Una mañana de hace unas cuantas semanas leí una noticia que, de no estar totalmente seguro que no era 28 de Diciembre, me habría parecido una inocentada: una conocida marca de automóviles ha diseñado “el primer coche sólo para mujeres” que además ha sido diseñado por un equipo formado íntegramente por mujeres.

Podéis hacer una visita a nuestro bien amado google y buscar el término “Volvo YCC” (no había manera de no mencionar la marca... publicidad gratis xD) para saber cuál es la idea que esas mujeres han tenido de lo que ha de ser un “coche para mujeres”. Seguramente, si eres mujer, desearías ver las cabezas de estas “diseñadoras” (<http://imgserv.ya.com/images/9/8/982e8235131ca7bi3.jpg>) en una pica... no te culpo, si yo fuera mujer desearía lo mismo.

Ahora (y antes de iniciar otra guerra de sexos), vamos a imaginar que alguien compra ese coche. Para dejar de martirizar a las féminas (que bastante tienen con ver su género como apellido de ese pseudocoche), imaginemos que yo mismo me lo compro. Sé que imaginar que tengo el dinero para comprar un coche -más aún de esa marca- es un gran esfuerzo, pero confío en que tengáis la suficiente imaginación. ;-)

Bien, pongamos que me he comprado mi nuevo y flamante YCC y decido salir a dar una vuelta a probarlo. Todo va de maravilla hasta que, de buenas a primeras el coche se para y me deja tirado. ¿Qué ha pasado? Bien, no perdamos la calma, vamos a abrir el capó a ver si veo algo a simple vista... ¿Qué!? ¡Este coche no tiene capó! (Nota: no, no es broma, ese coche NO tiene capó). Yo no soy un hacha de la mecánica, pero algo sé, y en cualquier caso tengo amigos que sí son bastante buenos con la mecánica... pero no podré hacer nada ni pedir a ningún amigo que lo haga. La única solución es llevar el coche al taller, donde sabrán cómo quitar la pieza completa que recubre el motor.

En ese momento empieza a funcionar la parte más pragmática de mi cerebro: ¿qué necesidad real había de negar el acceso al capó? ¿no sería mejor proporcionar el acceso al mismo y que cada usuario decida por sí mismo si desea abrirlo e intentar repararlo, o bien llevarlo a un taller y confiar en profesionales? Así se ha hecho toda la vida y no nos ha ido mal. El que quería aprender mecánica lo hacía y el que no, o buscaba un taller económico o le pedía un favor a un amigo que entendiera y luego le invitaba a algo (o no, que caraduras los han habido siempre :-P).

Traslademos ahora esta pequeña parábola al campo del software. Como hace mucho que dejamos de creer en los cuentos de hadas, sabemos que no existe el software perfecto, y que todos los programas -en mayor o menor medida- tienen fallos. Pues bien, en el mundo de la seguridad informática existen dos tendencias a la hora de manejar las situaciones críticas que suponen la aparición de bugs en el software.

La primera tendencia es la de ofrecer toda la información posible sobre el fallo: en qué consiste, cómo explotarlo, cómo solucionarlo... esta tendencia de “capó abierto” permite al usuario decidir por sí mismo si desea corregir el problema o esperar a que los profesionales (los responsables del software) lo corrijan. A esta tendencia se le conoce en este mundillo como “full disclosure” (completamente destapado).

La otra tendencia consiste en intentar que sea conocida la mínima información posible sobre el fallo. Generalmente se saca un parche para un fallo que ni siquiera ha sido anunciado con antelación. A esta tendencia se le conoce como “security through obscurity” (seguridad a través de oscuridad) y viene a ser el equivalente a no poner capó a nuestro coche.

Podéis encontrar más información sobre ambas tendencias en la wikipedia:

http://en.wikipedia.org/wiki/Security_through_obscurity

http://en.wikipedia.org/wiki/Full_disclosure

Dejando de lado las implicaciones legales de este asunto (un tema del que ya habla AZIMUT en su artículo de opinión) podemos observar que en la comunidad internacional dentro del campo de la seguridad informática, tanto empresas como particulares se decantan por una u otra tendencia (raramente por un término medio).

Generalmente, las grandes empresas de software prefieren que la seguridad de sus productos recaiga en la seguridad por oscuridad, mientras que los expertos independientes (los “hackers”, si es que esa palabra significa algo hoy en día) suelen preferir el full disclosure.

Pero técnicamente, ¿qué diferencias reales hay entre una y otra tendencia?

La política de seguridad por oscuridad significa en la práctica que dependemos totalmente de la empresa o particular que ha programado el software para poder corregir fallos. Y esto estaría bien si esos fallos se siguieran corrigiendo eternamente y se hiciera de forma eficiente, pero... ¿es así?

Hace unos meses Microsoft, el mayor gigante del software hoy en día, anunció que retiraba el soporte de Windows NT 4.0. En la práctica significa que no se van a desarrollar mejoras ni correcciones para el mismo nunca más... y que al primer fallo grave que se descubra, todo aquel con ese sistema estará completamente desprotegido. Dado que el entorno NT casi siempre se ha elegido como solución en empresas o entornos de producción, no creo que nadie en esas circunstancias se arriesgue a un fallo de semejante magnitud. ¿Solución? Actualizar, previo paso por caja para renovar tooodas las licencias que tuvieras.

En Microsoft los coches no tienen capó. Y cuando ellos deciden que debes cambiar de coche (sin importar que siga funcionando o no), en el taller te responden que no van a reparar más coches de ese modelo. Como el “plan renove” pero al revés y poniendo tú la pasta.

Pero voy más allá: ¿de qué me serviría conocer todos los detalles sobre un fallo de diseño en la junta de la trócola del modelo X del último coche de Microsoft? Aunque ese modelo de coche tuviera capó, al abrirlo descubriría que no tengo ni idea de cómo ha sido diseñado ese motor ni, por supuesto, de dónde está la junta de la trócola ni de cómo interactúa con su entorno para trabajar.

No sé si habréis leído los boletines de seguridad de Microsoft, pero a mí es que me da la risa. Parecen haber sido redactados con la intención de que no los comprendan ni sus ingenieros, y al final lo único que se entiende es “Descargar el parche”. Pues vale. Por no mencionar las descripciones de las actualizaciones que podemos encontrar en Windows Update... a mí a veces me da una sensación terrible de *dejà vu* y no sé si estoy viendo la misma actualización por enésima vez o si las descripciones se asignan por el método de “ctrl+c; ctrl+v”.

Por ello, creo que aún cuando las grandes corporaciones del software optaran por el full disclosure, éste no tendría sentido de ser sin tener acceso al código fuente del software en cuestión. ¡Y ojo! Que no estoy abogando por el software libre (que también, los que me conocen saben que soy firme defensor del mismo), simplemente pido derecho a saber en qué estoy poniendo mi confianza.

No sé vosotros, pero si el fabricante de mi coche de vez en cuando enviara cartas a sus clientes diciendo:

“Cuidado, se ha descubierto un fallo de diseño en su automóvil que puede causar que explote sin previo aviso.”

Yo, cada vez que escuchara un ruido extraño, por más que el mecánico insistiera en que no es nada, no me arriesgaría a ir a ningún sitio con ese riesgo con ruedas. No hablemos de hacer viajes de 500 kilómetros con pasajeros.

Últimamente hago bastantes compras online, y consulto los movimientos de mis cuentas desde la página web de las entidades bancarias correspondientes, por falta de tiempo más que nada (ya se sabe, la vida a la que nos obliga el *stress*). Y por el mismo motivo por el que no usaría ese coche sin capó, no uso software “sin capó” del que no me fío.

Es un hecho que las vulnerabilidades se descubren, por más que las empresas de software pretendan que no sea así. En muchas ocasiones son avisadas de esos fallos de seguridad y hacen caso omiso de los avisos. Recuerdo el caso, hará quizá un año o más, de una grave vulnerabilidad en hotmail (aquí es donde todos los *script-kiddies* agudizan sus cinco sentidos xD) que permitía resetear el password de cualquier usuario: el webmaster de "Zone H" avisó en repetidas ocasiones al personal de MSN y no le hicieron caso. Al final, cansado de la situación, decidió publicar el fallo, lo que obligó al personal de MSN a trabajar a destajo para corregirlo en una noche y evitar el desastre que se avecinaba (lo siento, *script-kiddies*, otra vez será...).

¿No sería mejor haber agradecido la información y haber solucionado con tiempo (semanas) el fallo que haberlo hecho de prisa y corriendo en una noche? En el colegio, todos hacíamos los deberes el día antes, pero creo que una empresa como Microsoft no puede permitirse hacer los deberes el día antes. Eso ya no es "security through obscurity"... eso es "security through idiocy".

Ahora imaginemos que se descubre una vulnerabilidad crítica en el software X y que el descubridor tiene otras intenciones bastante más dañinas. Nadie (ni el desarrollador) conoce el fallo, pero todos empiezan a observar los ataques y sus consecuencias. Con el código fuente disponible, y una política de full disclosure, es posible que cualquiera encuentre el fallo auditando el código y proponga una solución, mientras que si el código no está disponible, se hace bastante más complicado el encontrar el punto exacto del fallo y mucho más complicado solucionarlo... eso sin tener en cuenta que el realizar ingeniería inversa sobre código propietario es delito. Y si tenemos que esperar a que el desarrollador parchee, y tiene la misma prisa que demuestran a veces empresas como Microsoft... estamos apañados.

Es por ello que mi filosofía del software pasa por el full disclosure y la publicación del código fuente. Y de nuevo repito: la disponibilidad del código fuente no implica necesariamente software libre ni open source. Tenemos casos de software muy famoso como PGP (<http://www.pgp.com/>) donde el código fuente está disponible para descarga y revisión (<http://www.pgp.com/downloads/sourcecode/>) pero NO es libre ni abierto.

Es por ello que yo sí me fío de PGP, por ejemplo. No pecaré de soberbia diciendo que he revisado el código fuente de PGP, entre otras cosas porque no veo necesario revisarlo. Tampoco he revisado línea a línea el código de, por ejemplo, phpBB (el sistema de, entre otros, los foros de Hack x Crack)... pero porque no necesito hacerlo.

Sé perfectamente que hay gente que sí audita esos código línea por línea, y que en el momento en que se encuentra una vulnerabilidad, se publica toda la información relativa a la misma, así como soluciones temporales. A la experiencia me remito:

Hace un par de semanas se descubrió una vulnerabilidad muy grave en phpBB hasta la versión 2.0.12 que permitía comprometer totalmente cualquier foro. Gracias al full disclosure pudimos cerrar el foro a tiempo para evitar males mayores (y gracias a la gente de elhacker.net ;-P) y en cuanto hubo parche oficial, solucionar el problema.

Pero voy más allá... ayer mismo se empezó a mencionar en ciertos círculos una nueva vulnerabilidad en phpBB hasta la última versión (2.0.13) que permite hacer más de una maldad. Aún no hay parche oficial ni versión 2.0.14, pero gracias al full disclosure, he podido aplicar yo mismo una modificación al código del foro que nos pone a salvo del fallo. ¿Os imagináis si phpBB fuera código cerrado y propiedad de una gran empresa del software? Mejor ni pensarlo.

Por desgracia, como poderoso caballero es Don Dinero, las corporaciones del software influyen a gobiernos para que el full disclosure sea castigado (caso Guillermito) o para perjudicar abiertamente al software libre (aprobación de las patentes de software en el Parlamento Europeo).

Parece ser que, a golpe de talonario y mordaza, desean imponernos software "sin capó". Puede parecer que al final tenía razón George Orwell, y solamente erró la fecha... pero como no soy alguien pesimista, prefiero pensar que no es así.

Si observamos las tendencias del software de estos últimos años, vemos que cada día el software libre arrebató más trozos del pastel a las grandes empresas: Linux a Windows, Firefox a Internet Explorer, OpenOffice.org a Microsoft Office... y también hay casos de pasteles que el software libre se come casi completamente: MySQL, Apache...

También vemos que el full disclosure cada día está más vivo, y que la comunidad hacker se encarga de alimentarlo día a día con conocimiento... a pesar de todo. Como reza la firma de un amigo y compañero del foro... "Si no se vive como se piensa, se acabará pensando como se vive".

Así pues, cuando me compre un coche, lo pediré con capó. Y si me lo dan sin capó, ya me encargaré yo de ponerlo, os lo aseguro.

Distribución de este documento

Este documento se distribuye en formato PDF realizado bajo OpenOffice.org 2.0.

Ficheros a distribuir:

Nombre: "osc_trans.pdf"

Descripción: Documento principal.

Nombre: "osc_trans.pdf.sig"

Descripción: Firma digital PGP del fichero "osc_trans.pdf" realizada por el autor.

Nombre: "hash.txt"

Descripción: Contiene las cadenas hash MD5 y SHA-1 de los ficheros "osc_trans.pdf" y "osc_trans.pdf.sig".

Datos adicionales:

Las cadenas de hash MD5 y SHA-1 pueden resultar útiles para comprobar la integridad del fichero descargado, pero no son garantía de la inalterabilidad del documento, pues éste puede haber sido alterado junto a las cadenas de hash.

Para comprobar la completa autenticidad e inalterabilidad del fichero, es necesario utilizar el sistema OpenPGP para validar el fichero .sig (MIME/PGP) de firma. Cualquier modificación no autorizada del documento hará que la firma del mismo no sea válida, y ésta es imposible de falsificar.

Death Master

Autentificación:

-----BEGIN PGP MESSAGE-----

```
qANQR1DBwUwDJoT5ygJgu7ABD/9LFubOKAuJG595Nf7+qErBnJVPE6AS7kDgZr2w
E8xIBfACqTzXcYQfNBRLrTlKpZ8ejCFsRIZtVQb1CZVrmmOzi9ofWGrGug+hLmr
vh0OCobzG10bdbgwzQh/ClayV2MAy0c6R7N5dQRbloZb3PR4WYYPtJ3ehcvRygb3
k0BWT1NnuOi2t/r/W3wTUYYNsXcd2Qkvf5DW+DrMKocJlakPf7nZUj7ergS/H121
yehih09c58d03LL/R/WzoKxMVlQ3Yt9muk7wdZzc9NWJsl3uzco/py4v7rPY8Ai4
5ec7nJ4/JKCDN3JG9o+fyzSpvCh53zXSSFDKjiTy3cojxuFhBRGKz1wlXWn1kbTT
5MKCMpt4QdeL/THcVe90A0Ceyn8CwGITPStfs+o3+040VVnBuMdidkx73mqJbDC
PRdxJXby/APpslu0irKF+yFXJulpZOsvU4apGbREWsc8KT8X1J2wTLEJlgByTXe2
QBh9PT90hhHvJailxXasNJCnkoZhGfOkmV0XPm3sw9JHCmCmF0dH7K2cwlAy7Y
5a9loda4uX7EG5f1FPNKBdWaxUJa8gOYEQqkN5uMXDKp8AQKHgC/3buXuQ3/mOCV
6Ci1ljbhRKHiKYypysv+9sO/uz6oDej+9lIUJNqeF9vzjbH9EKuOy387qG3nlqS
48r6+9LC4wFP4UqaB9aBMTNEIzpCwc5AZMDCWJbO59+BbD+qLr5gjqmL7QXcac+g
oCQDAmj/2k9aw7xYOhwZg8nT4USkpHVY0kM6ipx5DfJfAbdza7NMh4XL0WNasd+L
rna+GoPjhclE6/eDvrzgzgKNUg0QrsVZqvIGDAAQQAxCeQ3t+EFhCs6cD3IAbDp6
2taTnvXqBFIDFNy4YpKfPYeu9i+e6GkbT1loN1qhn/JRL5a95/rj/hBVp7T8wiW
dxbs6eQBGFnOzJLqo8DEpmZU0IRNnzf6hInK+ZgvtW5jMSLXdIZODxspIDelzXJk
6Nc4tjo8KKX+vaG8osySg0YY6Azm4wW2uGM2mlLtGBfmQ9zeW/sKHEWT1C3eZk2
nbPxOKa67AfQlppN/8tWL6mXjRs/rVILfhDrCAXQ+xdkBDeuvxEpliXyLj+lrwN4
LtEVd23Tgi4ido412BzCucgkoVkf66asYk68CQxqgbm5S1RE00LBcyDPbaM/ofy/
GLqwGdP+4oh1LodeJelpvKfYtoGpv4PcX8Uaa90v8edWQw1dP41ZZhwcl4V99iot
f95lcvAlh6gNZWOYARGelAnBFWCFiZ4mlNME13dh27mBzB42RVjAdnuq6QS2l68l
d71+5q9koYzi9yqgbtjzgoJcgWYx9ZHH+RjMghQVwyknQq/+slTgSIWwhMa/X40dQ
uQ477+HmmdibpHrxGYq7jhWhXy10Xs2fMQD2SU4nyv6iOtXi+zfapsbYrvDjohkG
Z6vK0rVse5LseQdx7HRz6bfoLCh4FO8SDyLXK7RpaNOt0bGMA7MhpCwkmI45NrrR
l+F2tVgKMIqNaEVlpVeB7uIDVSMX8Y/czJU4tcRLDk+0eotRcri5OGWDP6bxWaHh
yv9SBCryFggUFYtWOr0ReQlX1/wUICho7nhSRdoprwwQK1eCGCC2/A8XIVXss8it
cOo8EJxzM0wx5Jbp55yHt6t0CzajL25W/oirRFxiBr8Z7NpalChEcEFvB6Gp96IG
xEKmm6VJsBbxPCBi8A00cu3tM67IFR0AF7JbeZgF8DrX+X8mwvSdH0MOio6tp7m
olo688e6dfCnVic4nTVhXE4qvwfJf/crGs/SVbFAp7gDMnalzziKZacYjUCZtNur
dO6C+wHf1n7CFtrB8NJm6h4s4V0eW1eFGqTl7Y2QmZGPIkbGX6O69m8Si0Wwv1U+
FbCW39BDaGUIBSEnLBP4Ph9JCeZUx4ss6KY=
```

=jgS1

-----END PGP MESSAGE-----

Licencia

¿Oscuridad o transparencia? – <http://www.death-master.tk/> – Versión 1.0

Este documento ha sido liberado por su autor bajo la licencia GNU Free Documentation License (GFDL), y su utilización, copia o reproducción queda sujeta a los términos de la citada licencia, que puede ser consultada en el siguiente sitio web:

- **GNU Free Documentation License:** <http://www.gnu.org/copyleft/fdl.html>
GFDL Version 1.2, November 2002
Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc.

Copyright (c) 2005 Death Master

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being “Distribución de este documento” and “Licencia”, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Cualquier copia, modificación, distribución o utilización en general de este documento debe respetar la autoría original del mismo, correspondiente a **Death Master**.

Darkness or transparency? – <http://www.death-master.tk/> – Version 1.0

This document has been freed by its author under the license GNU Free Documentation License (GFDL), and its use, copy or reproduction is subject to the terms of the mentioned license that can be consulted in the following website:

- **GNU Free Documentation License:** <http://www.gnu.org/copyleft/fdl.html>
GFDL Version 1.2, November 2002
Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc.

Copyright (c) 2005 Death Master

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being “Distribución de este documento” and “Licencia”, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Any copy, modification, distribution or general purpose use of this document should respect the original responsibility of it, corresponding to **Death Master**.



*** End Of File ***