



¿QUÉ  
PREOCUPA?

## LA RESPONSABILIDAD Y LA GARANTÍA SOBRE EL SOFTWARE Y SU SEGURIDAD

El pasado 26 de abril tuve la fortuna de asistir a un interesante debate sobre la conveniencia o no de utilizar software de seguridad de código abierto en la empresa, debate que posteriormente se amplió al software de código abierto en general, todo ello en el marco de la XVII edición del Congreso español de Seguridad de la Información (Securmática).

Fueron muchos los argumentos mostrados a favor y en contra del software libre y el debate, como era de esperar, no finalizó con un ganador ni con un resultado definitivo puesto que, a mi entender, no se puede decir que el uso del software libre en la empresa sea del todo bueno o del todo malo, todo depende de las circunstancias concretas de uso, incluso tratándose de software de seguridad.

### Seguridad y confianza

En resumidas cuentas y según mi criterio, no se trataba de si el software es de código abierto o cerrado o de si está creado por una

software pueda ser considerado como confiable y, por tanto, seguro.

### Cuando falla la seguridad: garantía y responsabilidad

Además, para que la confianza y la cobertura sean aún mayores y podamos empezar a hablar realmente de seguridad, se necesita una "red de funambulista" que, en el caso de que todo lo anterior falle y se produzca un incidente debido a un fallo ocasional del software, tengamos una cobertura o garantía y responsabilidad de alguien que nos cubra y repare los daños que nuestra empresa haya sufrido.

Ya sé que es muy difícil hablar de responsabilidad en el software, cuando las cláusulas de utilización de los productos de los principales fabricantes multinacionales de hoy en día manifiestan claramente esa falta de responsabilidad y de garantía. Dichos fabricantes argumentan que el software moderno es muy complejo, que muchos de sus productos tienen millones de

se pueden y se deberían proporcionar ciertas garantías y responsabilidades. Al final, se trata de un producto comercial y cuesta creer que no exista garantía o responsabilidad sobre la seguridad del mismo. Ello denota que el estado de madurez de la industria del software y de su seguridad no es tan avanzado como podríamos pensar. Como ningún fabricante se responsabiliza sobre el software, ello ya no es una necesidad competitiva. ¿Lo será cuando el software libre incremente su avance?

### Y los gobiernos y reguladores, ¿qué dicen de todo esto?

En este contexto, uno podría pensar: cuando la industria no se autorregula es cuando deben aparecer las regulaciones de los gobiernos pero, en este caso, tampoco los gobiernos y los reguladores habituales obligan a la existencia de responsabilidades y garantías sobre el software desarrollado, ni tan siquiera en relación con la seguridad del mismo. Eso sí, las regulaciones y leyes que obligan a las corporaciones y empresas de todos los sectores a securizar sus sistemas de información son amplias y variadas, todos las conocemos, pero parece paradójico pensar que se pueda obligar a las empresas usuarias a "securizar" sus sistemas de información y, en cambio, no se obligue a los fabricantes de productos de seguridad a responsabilizarse y garantizar que sus productos sean seguros.

### El dilema: ¿avanzo rápido o avanzo seguro?

Al final, yo creo que en todo esto aparece el siguiente dilema: ¿hago las cosas bien y perfectamente seguras, con lo cual mis productos y los de mi competencia van a tardar más tiempo en aparecer en el mercado y, tratándose de productos de software, ello va a repercutir en ralentizar la llamada "sociedad de la información" o, por el contrario, continuamos la carrera actual, en la que lanzamos los productos sin demasiada certeza de seguridad ni responsabilidad y, vamos resolviendo a base de "parches" los problemas que otros nos descubren? Al fin y al cabo, hasta ahora todavía no ha habido grandes catástrofes debidas a agujeros de seguridad del software no tapados a tiempo. No obstante, ¿esto será siempre así? No es bueno confiar en la suerte. Al menos, no entre los profesionales de la seguridad. ■

Proporcionando ciertas garantías y responsabilidades sobre sus productos empresariales, los fabricantes de software podrían crear confianza en sus clientes y, con ello, seguridad...

determinada compañía o por una "comunidad" libre, sino de la confianza y de la garantía que un determinado producto nos dé y del nivel de servicio asociado al mismo.

No debemos olvidar que estamos tratando de productos software para uso empresarial, entendiendo como tal un entorno en el que la fiabilidad del software se convierte en un aspecto crítico. En este caso, confianza es sinónimo de seguridad y, por tanto, una característica que en todas sus dimensiones resulta elemental cuando se trata de software de seguridad: confianza en que dicho software no tiene "agujeros" de seguridad; confianza en que actúa como se espera que lo haga en todas las circunstancias que se le exigen; confianza en que el producto software no contiene líneas de código indeseado introducido por alguno de sus creadores o de sus manipuladores, es decir, que no lleva incluido un "caballo de troya" o alguna "puerta trasera"; confianza en la continuidad de quien desarrolla el producto y en que sus diferentes evoluciones seguirán el camino lógico que se espera de dicho producto y que éste cumplirá todos los estándares y regulaciones a que se ve sometida una empresa; confianza en que, si aparece algún defecto en dicho software, alguien garantizará una solución en el plazo acorde con el riesgo que el defecto comporte.

Seguro que a los lectores se les ocurren otros muchos requerimientos para que un producto

líneas de código y que no se puede garantizar en ellas la inexistencia de fallos, que las matemáticas no demuestran que se pueda crear software libre de fallos, etc. Sin embargo, supongo que estará de acuerdo conmigo en que no podemos hablar de confianza si no tenemos tal garantía. Y al final, ya sea el software de código abierto o cerrado, para mí la garantía es lo que cuenta.

Estoy de acuerdo en que hoy en día es difícil garantizar la ausencia de fallos en un producto software y que los fabricantes tienen parte de razón pero, bajo ese argumento, tenemos la excusa perfecta para evitar todo tipo de responsabilidades y obligaciones, tal y como sucede actualmente. Es aquí donde creo que los fabricantes de software, a diferencia de los creadores de software libre, podrían tener una ventaja. Proporcionando ciertas garantías y responsabilidades sobre sus productos empresariales, los fabricantes de software podrían crear confianza en sus clientes y, con ello, seguridad.. De lo contrario, los creadores de software libre irán ganando mercado a los de software propietario, pues a paridad de funcionalidades y calidad, y en ausencia de garantías por ambas partes, la conveniencia de su precio determinará una inclinación del mercado a su favor en detrimento del software propietario.

Y es que, sin llegar a la garantía y responsabilidad total del producto software creado,



Xavier Serrano Cossío  
Responsable de Seguridad  
Tecnológica

GRUPO BANCO SABADELL