Programación segura de aplicaciones

Jordi Serra (coordinador)
José María Palazón Romero

PID_00165011

Material docente de la UOC

Universitat Oberta de Catalunya
www.uoc.edu

Jordi Serra

José María Palazón Romero

Primera edición: febrero 2010
© José María Palazón Romero
Todos los derechos reservados
© de esta edición, FUOC, 2010
Av. Tibidabo, 39-43, 08035 Barcelona
Diseño: Manel Andreu
Realización editorial: Eureca Media, SL
Depósito legal: B-9.086-2011

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares del copyright.

Introducción

Desde hace ya muchos años, cuando doy charlas de seguridad, independientemente de cuál sea el contenido concreto de cada una de las charlas en particular siempre encuentro un buen momento para hablar de uno de los aspectos que considero más importantes en seguridad informática: la seguridad no va sobre vosotros. Un proyecto informático está compuesto por muchas partes de las que son responsables distintas personas, y cuanto más grande es el proyecto, mayor es el número de actores que participan en el mismo y más delegada y repartida está la responsabilidad, lo cual viene asociado normalmente a un mayor nivel de especialización concreta en cada una de las áreas por parte de los profesionales que se dedican a estas.

La parte negativa de esta separación es el aislamiento, y en lo que respecta a seguridad, se produce un efecto muy interesante que es el aislamiento mental. Si preguntáis a un administrador de sistemas sobre seguridad informática, sus respuestas van a estar enfocadas a evitar que el atacante consiga una cuenta de superusuario en la máquina. Esto, para él, es la brecha total y absoluta, el control total por parte del intruso del sistema, y todo lo que sea impedir que el atacante llegue hasta aquí no es excesivamente crítico. De esta manera, los administradores de sistemas virtualizan, enjaulan y afinan los permisos, todo con el objetivo de que la máquina final no quede totalmente comprometida.

Un programador os hablará de filtrado de la entrada del usuario y de almacenamiento criptográfico de información sensible, y centrará sus respuestas en la seguridad de la lógica de su aplicación.

Al responsable de contenidos todo lo anterior le da igual, porque si el programa más seguro del mundo, en el *data center* más protegido y mejor administrado, permite una política débil de contraseñas que haga que cualquiera pueda acceder fácilmente a la cuenta de uno de los editores y así cambiar el contenido de su portal, la seguridad ha sido comprometida para él en el punto más crítico.

La seguridad es un proceso general en el que deben estar involucrados todos los actores que participan en el desarrollo y despliegue de una aplicación, ya que cada uno aportará su punto de vista sobre los activos que hay que proteger.

Relacionado con lo anterior está el concepto de seguridad en profundidad, que es como se conoce a la práctica consistente en considerar cada capa de nuestra aplicación como algo que hay que proteger en sí mismo, sin contar con la protección adicional que otras capas puedan estar aportando. En un mundo ideal, la primera capa parará todos los intentos de ataques, y toda esta seguridad en profundidad puede parecer redundante. Sin embargo, el día en que un atacante consiga superar esta primera barrera, o se publique un fallo

de seguridad en esta tecnología en la que confiamos, nuestra aplicación quedará desnuda sin seguridad en profundidad, mientras que con esta el atacante tendrá unas cuantas vallas más que saltar para conseguir su objetivo.

"La seguridad es un proceso" es posiblemente la frase más repetida en textos y conferencias de seguridad, y esto es así porque se trata de algo en lo que absolutamente todos estamos de acuerdo. Todos excepto los que no se dedican a ello. El sistema operativo más seguro del mundo no sirve de nada si no se le aplican los parches de seguridad del fabricante con frecuencia. La librería criptográfica más potente se convierte en la más débil si no actualizamos y regeneramos nuestras firmas tras publicarse una vulnerabilidad que describe cómo suplantar certificados con cierta implementación. Es un proceso vivo, continuamente cambiante, en el que hay que estar siempre alerta y pendiente de nuevas técnicas y vulnerabilidades en general, y de las que afectan directamente a nuestras aplicaciones y sistemas en particular.

Hay cientos de textos sobre seguridad informática, y prácticamente todos se centran en describir cuáles son los problemas y cómo aprovechar (explotar) fallos de seguridad y vulnerabilidades o cómo escribir herramientas para esto. Si bien en esta ocasión, por supuesto, describiremos los problemas para tener un contexto de lo que estamos hablando, no vamos a profundizar en la infinidad de trucos y técnicas de explotación que existen para sacar partido de estas vulnerabilidades, sino que vamos a centrarnos sobre todo en cómo conseguir programar una aplicación segura. Qué prácticas debemos tener en cuenta para no caer en las vulnerabilidades más comunes, y qué *frameworks* y tecnologías encontramos que ya hacen por nosotros las tareas más delicadas, como la autenticación, la autorización, el mantenimiento de sesiones o la criptografía. Como regla general, podemos decir que si en algún momento nos encontramos programando nuestra propia manera de hacer alguna de estas cuatro tareas citadas, casi con un 100% de seguridad estaremos cometiendo un error.

Normalmente, estos textos dejan un pequeño apartado o tabla resumen al final de cada capítulo sobre las maneras de evitar estos problemas como desarrollador, y suelen quedarse muy cortos con las recomendaciones, e incluso a veces proponen soluciones que ni siquiera son completas para evitar los ataques que se acaban de describir.

En este texto tomaremos un enfoque contrario. Es esencial describir las vulnerabilidades y, en algún caso, con objeto de ilustrar la gravedad del impacto de alguna de estas, veremos con algo más de detalle cómo un atacante las aprovecharía de distintas maneras. Sin embargo, en todos los capítulos centraremos la atención en cómo hacer las cosas correctamente, cómo programar de manera adecuada para que las vulnerabilidades descritas no aparezcan en nuestro código o minimizar su impacto si acabaran apareciendo.

Además, dedicaremos dos módulos completos a infraestructuras de autenticación y autorización en las plataformas de desarrollo web más comunes, así como infraestructuras delegadas siguiendo los estándares actuales de la industria. Utilizando estos mecanismos y siguiendo estos estándares, evitaremos tener que desarrollar de manera personalizada precisamente las áreas en las que el resto del presente texto basa la mayoría de las vulnerabilidades expuestas.

Agradecimientos

Las siguientes personas han sido colaboradoras esenciales tanto en su aportación de contenidos como en asesoramiento para hacer que este texto comprenda contenidos de calidad avalados por expertos de seguridad de cada una de las distintas áreas:

- **Chema Alonso**. Ingeniero superior en Informática por la Universidad Rey Juan Carlos e ingeniero técnico en Sistemas por la Universidad Politécnica de Madrid. Ha trabajado como consultor de seguridad durante los últimos seis años y ha sido reconocido como *Microsoft Most Valuable Professional* (MVP) desde el 2005 hasta la actualidad. Es ponente regular en conferencias de seguridad y escribe mensualmente en varias revistas técnicas españolas. En la actualidad trabaja en su tesis de doctorado sobre técnicas de inyección a ciegas. Recientemente ha expuesto en BH Europe 2008, Defcon 16 y 17, Toorcon X, DeepSec2k8, HackCon#4 y SchmooCon 2k9.

- **Elena Galván**. Ingeniera informática por la Universidad Politécnica de Cataluña (UPC) en el 2008. En este año también llevó a cabo el máster en Tecnologías de seguridad informática de esCERT-UPC y TB-Security y el máster en Auditoría y protección de datos de la Universidad Autónoma de Barcelona. Del 2005 al 2006 fue administradora de sistemas en Power-Data y en el año 2006 pasó a ser directora técnica del CERT de la UPC (esCERT), y ha colaborado en distintos proyectos de seguridad: GIDRE, Seguridad2020 y RAFFI. Además de impartir distintos cursos para el personal de la UPC, en el 2009 fue profesora en la Facultad de Informática de Barcelona (UPC), donde impartió la asignatura de Seguridad en sistemas informáticos, y actualmente es tutora del máster de Software libre de la Universitat Oberta de Catalunya.

- **Daniel García**. Ingeniero técnico en Informática de Sistemas por la Universidad de Sevilla en el 2005. Durante cuatro años, actuó como ingeniero de sistemas en Sadiel, S.A. hasta el 2005, cuando decidió dar un giro en su carrera profesional y dedicarse al sector de identidad digital como desarrollador de PAPI, la infraestructura de autenticación y autorización desarrollada por RedIRIS, y en el área de PKI (infraestructura de clave pública), desplegando y gestionando dos PKI, ambas participes de proyectos internacionales como pkIRISGrid y eduGAINSCA.

- **Elena Lozano**. Ingeniera informática por la Universidad de Sevilla (España), obtuvo en el 2007 el título de ingeniera técnica informática de gestión en esta universidad. En el 2007 trabajó con la FECYT en el proyecto "Portal de Acceso a la Web of Knowledge". Durante el año 2008, se unió como organizadora al equipo humano responsable de la segunda edición del Concurso Universitario de Software Libre.

- **Alejandro Martín**. Máster en Sistemas informáticos por la Universidad Rey Juan Carlos e ingeniero informático por la Universidad de Salamanca. Actualmente es el desarrollador jefe de Informática64, donde entre otros proyectos ha dirigido MetaShield Protector. Es ponente habitual de conferencias de seguridad y ha participado, entre otras, en las FIST, Asegúr@IT y la campaña *Hands On Lab*.

- **Diego Pérez**. Ingeniero superior en Informática, actualmente trabaja en Yahoo! como ingeniero de *software*, donde es responsable de la adaptación de diferentes productos para el mercado europeo. Anteriormente ha trabajado para empresas como Telefónica o Ericsson I+D en la arquitectura y el desarrollo de soluciones de autenticación, autorización y acceso remoto.

- **Cándido Rodríguez**. A día de hoy inmerso en el programa de doctorado en Informática Industrial en la Universidad de Sevilla (España), obtuvo en el 2004 su título de ingeniero informático en esta universidad. Desde el 2001 al 2005 desarrolló su andadura profesional en el Departamento de I+D de Optima Technologies. En el 2006 se incorporó a RedIRIS, donde se ha dedicado principalmente a la identidad digital y ha sido el responsable de la infraestructura de autenticación y autorización del proyecto perfSONAR.

Contenidos