Aspectos fundamentales de desarrollo seguro

Autor: Jaime Paqui

Mayo 2022

1 Introducción

Es importante entender que la seguridad debe ser parte del desarrollo como parte integral de las herramientas de integración, no como un componente separado, sino como parte del desarrollo. Con el objetivo de tratar de reducir los riesgos de seguridad de estos programas.

Como buena práctica se debería poner tantas ideas de negocios como sea posible en la sección de servidor/backend, evitando que alguien cambie de ingeniería y sacarle más provecho. Debemos darnos cuenta de que el atacante solo necesita un pequeño virus, una amenaza, para lograr su objetivo.

Si se ignoran los problemas de seguridad dentro de la empresa para acelerar los negocios, podemos dejar la puerta abierta para que la seguridad de la información se vea comprometida.

La gestión del sistema es una parte importante para garantizar que las cosas funcionen sin problemas. En particular, para quienes están directamente involucrados en el desarrollo, la responsabilidad es grande, porque, al final, son ellos quienes van a los requerimientos y establecen las soluciones a los problemas.

Por estas razones, nos animamos a considerar otras modificaciones en su desarrollo, para evitar eventos como cross-site scripting (XSS), código de inyección, ejecución de código malicioso.

Controles establecidos dentro del proceso deben asegurar la entrada, procesamiento y salida de datos, de manera que se pueda cumplir confidencialidad, fiabilidad y disponibilidad. Entonces de este modo, se deben establecer indicadores de seguimiento que ayuden a identificar oportunidades no autorizadas,controles de preventivos, diseñados para evitar la entrada de datos no autorizados o no válidos, y controles proactivos, que permiten probar los componentes de forma periódica.

2 Desarrollo

2.1 Controles proactivos

Ademas de los controles en uso, también es importante especificar un protocolo de prueba integral que incluya pruebas estandarizadas y usuarios finales, donde se monitorea el sistema operativo. Una forma de tener programas seguros, los administradores deben contar con el apoyo y la asistencia de una organización que escribe el código, lo que a veces no es el caso.

Todos los componentes que conforman un programa, como la interfaz de usuario, el concepto comercial, el controlador o el archivo, deben diseñarse teniendo en cuenta la seguridad. Los lenguajes y sistemas utilizados por los programadores para crear software a menudo carecen de controles importantes o son inseguros.

2.2 OWASP

OWASP nos proporciona una lista de recursos, sobre todo de herramientas y herramientas, para que nuestros proyectos online sean lo más seguros posible, durante todo el proceso de desarrollo. Para los desarrolladores y lectores de seguridad, ofrece una gran cantidad de recursos para ayudar a asegurar la vida del software.

2.2.1 Objetivos

- Ofrecer a los desarrolladores las mejores formas de hacer que el software sea lo más seguro posible.
- Proveer a los programadores y expertos en seguridad y recursos para proteger el software, especialmente en aplicaciones móviles.

El propósito del proyecto es tratar de ayudar a los desarrolladores a entender el qué, cuándo y como verificar aplicaciones en línea.

OWASP es una organización de código abierto dedicada a capacitar a las organizaciones para crear, adquirir y mantener software en el que se pueda confiar.

2.2.2 Riesgos más comunes

- Inyección: problemas de inyección, como SQL y LDAP, se dan cuando se envían datos poco confiables al interprete como por ejemplo por medio de un comando.
- Pérdida de autenticación y gestión de sesiones:Las actividades relacionadas con la validacion y la gestión de componentes a menudo se realizan de forma incorrecta, permitiendo a los atacantes exponer password, claves,etc.
- Secuencia de comandos en sitios cruzados:Errores a nivel de XSS se presentan cuando un programa recupera información que no es confiable y traslada estos datos a un navegador sin autenticarlos ni verificarlos.

- Configuración de seguridad incorrecta:Para tener una seguridad alta se necesita de una configuración segura, que esté definida e implementada en la aplicación, el enlace, el servidor operativo, el servidor web, la BD. Todas estas preferencias deben definirse, establecerse y mantenerse porque a menudo no se protegen automáticamente.
- Exposición de datos sensibles:La mayoría de aplicaciones en línea no respaldan por completo los datos cifrados, como por ejemplo los números de las tarjetas de crédito. En su mayoría los piratas informáticos pueden robar o alterar la información para cometer fraude, robo de datos u otros delitos. El cifrado de datos requiere medidas de seguridad complementaria, como el cifrado de datos, y un cuidado especial en el intercambio de datos con los exploradores de internet.
- Falsificación de peticiones en sitios cruzados:Los ataques CSRF estos lo que hacen es forzar al explorador de internet legítimo a enviar una solicitud HTTP falsa, incluido el componente de usuario y cualquier otro tipo de datos, adjuntando a una app web débil.
- Utilización de componentes con vulnerabilidades:Algunas de las funciones, como frameworks y otros, es común que siempre funcionen con todos las licencias. En caso de un ataque a una área vulnerable, esto podría provocar la invasión del servidor o el extravío de la información y datos.
- Re-direcciones y reenvío no validados:Las aplicaciones de Internet suelen enviar y reenviar a los usuarios unas u otras páginas y hacen uso de datos que no son confiables para identificar la página a la que se dirigen. Sin la confirmación adecuada, los piratas informáticos pueden dirigir a los afectados a un sitio web fraudulento o a un programa de malware, o utilizar el correo electrónico para acceder a páginas no autorizadas.

Un servicio interesante proporcionado por OWASP es el (ZAP) de código abierto el cual nos permite analizar todos los envíos y recibos al navegar por una página web.

Esta herramienta admite análisis tanto predeterminado como manual. El método manual es más simple y menos costoso porque, en el caso de que se necesite estar en línea e ingresar a un login, solo busca hasta ese punto. En el caso de usar el método predeterminado, todo lo que tiene que hacer es ingresar la URL que desea analizar en la sección url para atacar y hacer clic en el apartado atacar.

Podemos también decir que el método manual,ejerce como un proyecto para recibir solicitudes realizadas en el explorador web, por lo que puede analizar cualquier URL para capturar contenido en formularios de usuario y secciones de navegación.

2.3 OWASP Mobile Security Project

Tiene como objetivo concienciar sobre la seguridad en el software web, para identificar otros riesgos a los que se afrontan las organizaciones. La lista de tecnología móvil de OWASP se centra en la seguridad del software para dispositivos móviles.El propósito es establecer amenazas de seguridad y proporcionar autoridades para reducir el riesgo o de esta manera tener el software móvil seguro. El proyecto OWASP Mobile Security proporciona a los fabricantes y equipos de seguridad las herramientas para diseñar y mantener teléfonos seguros.Por otra parte le brinda la oportunidad de compartir riesgos de seguridad prácticos y proporcionar guías de desarrollo para reducir la vulnerabilidad o la violación.

2.3.1 Top 10 de los controles de seguridad para evaluar en aplicaciones móviles

- Ml-lmproper Platform Usage:Se preocupa por el mal uso de los dispositivos portátiles.
- M2-Insecure Data Storage:Protege el Storage de los datos inseguros y la pérdida innecesaria de datos.
- M3-Insecure Communication:Este grupo está preocupado por el mal establecimiento de conexiones. Las comunicaciones a las que se envíen datos de los usuarios deberán realizarse a través de un proceso seguro.
- M4-Insecure Authentication:El equipo tiene la idea de asegurar a los usuarios finales o una pésima administración de la sección de usuarios. Por ejemplo algunos modelos no logran identificar al usuario, por lo que pierden la pista de lo que está sucediendo o no mantienen la información del usuario relevante cuando es necesario.
- M5-Insumcient Cryptography:La criptografía se utiliza en el cifrado para proteger objetos ocultos. De echo, no siempre la criptografía es suficiente en la aplicación, por lo que se utilizan algoritmos criptográficamente dañados tales pueden ser MD5 o SHAI.
- M6-Insecure Authorization:Trata acerca de la falta de permisos en la aplicación. Es distinto a la complejidad de verificar e identificar a los usuarios.
- M7-Client Code Quality:Aquí podemos incluir otras características como desbordamiento de búfer, entre otros errores a nivel de desarrollo de código.
- M8-Code Tampering:El grupo combina parches o cambios con contenido local y memoria personalizable. El atacante puede modificar el código directamente, modificar el contenido de la lista de actualización o modificar la máquina API que utiliza el software, o modificar objetos y datos del software.
- M9-Reverse Engineering:El equipo incluye análisis binarios para encontrar

fuentes y bibliotecas. Algunas herramientas de software como IDA Pro, Hooper y otras herramientas de monitoreo permiten al atacante monitorear la actividad.

- MIO-Extraneous Functionality:Los desarrolladores a veces incorporan sistemas operativos ocultos a través de puertas traseras u otros controles internos, que no estaban destinados al sitio de producción, teniendo un ejemplo donde, un desarrollador puede incorporar por accidente una clave en la revisión de un programa o puede desactivar la autenticación de dos factores durante la prueba.

2.4 Controles proactivos OWASP

El proyecto Estándar de verificación de seguridad de aplicaciones (ASVS) de OWASP proporciona un marco para probar los sistemas de gestión de seguridad en línea y también proporciona a los programadores una lista de requisitos para un progreso de desarrollo de software seguro. Este modelo proporciona un apoyo para experimentar las pautas de seguridad en el software, así como las pautas técnicas para la seguridad local, que se utilizan para evitar problemas como (XSS).

Seguidamente, se tiene una breve discusión sobre las pautas que recomienda OWASP y que deben incluirse en cualquier desarrollo de software. Se lista los siguientes controles que se deben hacer:

- Verificación de la seguridad desde las primeras etapas de desarrollo: Una de las partes fundamentales seria hacer las pruebas de seguridad del proceso de desarrollo de software. Es importante que se verifique la seguridad lo más rápido y con la mayor frecuencia posible, ya sea por medio de pruebas manuales.

- Validación de las entradas del cliente:La falta más frecuente en la seguridad del software en línea es la ausencia de validez de los datos del cliente. Cualquier instalación del software debe ser verificada y filtrada, tanto en el frontend como backend. Muchas amenazas en línea se derivan de la falta de verificación de lo que escribe un usuario, cuando se trata de usar Internet y servicios, esto puede incluir lo siguiente:Cookies,Cabeceras HTTP, Parámetros GET y POST. Hay dos formas de verificar una contraseña, conocidas como lista negra y lista blanca.

  En cuanto a la lista negra es una lista de artículos no convencionales que buscan asegurar que lo que te han dado no contenga contenido maligno.

  En cuando a la lista blanca es una lista de funciones válidas que deberían garantizar que el usuario coincida con las entradas conocidas.

  Teniendo en cuenta la seguridad, siempre es mas factible hacer uso de las listas blancas. Si tuviéramos que usar una lista negra, tendríamos que ingresar una lista de todos los elementos no negros, que serían numerosos.

Pese a esto, tenemos conocimiento qué es aceptable y podemos enumerarlos inmediatamente, sin tener que ingresarlos a la lista mas adelante.

Las infracciones que se evitarán mediante el uso de la verificación de parámetros adecuada son:Cross-site scripting,Inyección de código SQL, Denegación de servicio en la aplicación y Desbordamientos de búfer.

- Desbordamientos del búfer: Son métodos utilizados para acceder al programa remoto ingresando el código malicioso en el programa que desea.
- Gestión de sesiones:Esta sección es uno de los componentes clave de cualquier actividad en línea, en la que se administra y mantiene el estado del usuario. Esta sección es una de las partes más importantes de cualquier evento en línea, donde se administra y almacena la interfaz de usuario.
- Implementación de controles de acceso:La inspección de acceso incluye la idea de permitir el acceso a los elementos para aquellos que tienen una licencia para usarlos.
- Implementación de controles de identidad y autenticación: La verificación es similar al proceso de probar que un individuo u organización afirma serlo, se realiza a menudo al enviar un nombre de usuario o ID. Las sesiones se almacenan en el servidor con identificación parcial,en donde es posible que se permita pasar de un lado a otro tanto para el lado del cliente como del servidor, al enviar y recibir solicitudes.

  Es esencial que el proyecto cumpla con los siguientes altos estándares: Puede determinar quién está tratando de ingresar al programa y Puede garantizar que solo los usuarios autorizados puedan validar.

  En el caso de protocolos válidos, siempre que la autenticación se realice mediante un nombre de usuario/contraseña y utilizando la autenticación de múltiples contraseñas, este método se considera uno de los más seguros en la actualidad. Los principales protocolos para la autenticación tenemos: OAuth(permite que la aplicación autentique al usuario contra el servidor, sin requerir una contraseña) y Openld (protocolo basado en HTTP que usa notificaciones para verificar al usuario.)
- Autenticación por múltiples factores:Uso de más de un agente de verificación para ingresar o procesar una transacción, mediante:Información de cuenta o contraseña y Tokens.
- Manejo de errores y excepciones:El objetivo primordial de corregir errores es proporcionar comentarios útiles de usuarios y grupos de respuesta. Es importante tener conocimiento de que una app certificada cumpla con los siguientes requisitos:No recopile ni escriba información confidencial a menos que sea necesario, Garantizar que toda la información registrada se gestione correctamente y Prevenir errores detallados.

  La edición especial le brinda la oportunidad de corregir una amplia variedad de errores que pueden ocurrir, ya sea en software o el sistema. La

inconsistencia de las excepciones, sin saberlo, puede reflejar mucha información del programa, que no debería estar disponible.

2.5 Ataques en aplicaciones web

- Vectores de ataque:Son mecanismos específicos que pueden hacer que un ataque sea efectivo.Para los vectores de entrada, podemos mencionar:Encabezados HTTP,enviar formularios a través del método POST,valores de parámetros de URL a través de GET, etc.Para los vectores de salida, podemos mencionar:HTML adquirido por el usuario, Hipervínculos y enlaces adquiridos por los usuarios.

- Cross-site scripting (XSS):La aplicación en línea está comprometida en XSS, donde el contenido que enviamos al servidor se muestra en la página de respuesta. En si, la invasión XSS funciona mediante la manipulación del motor de prueba HTML que utiliza el programa para entregar códigos incompatibles mientras todo lo que tiene que hacer es mostrar los datos.Los posibles fallos de XSS son:No permanentes y Permanentes.

- Cross-site request forgery (CSRF): El CSRF se basa en las variables estándar o predecibles, así como en el punto de partida para la navegación.El formulario de envío debe tener un token, que se completó en el formulario o al comienzo de la sección de usuario.Para mejorar la seguridad ante posibles ataques CSRF, podemos utilizar cabezeras de Referer.La única protección real contra los ataques CSRF radica en la provisión de una serie de protecciones únicas para cada solicitud.

  Recomendaciones para evitar CSRF:Salir tan pronto se use la aplicación, evitar que el navegador almacene información de cada página o servidor para recordar nuestra sección más del tiempo que estemos usando y por ultimo se recomienda hacer uso de diferentes navegadores.

- Seguridad en las redirecciones:Otros monitoreos y progresos no convencionales son posibles si la aplicación web acepta entradas no confiables, lo que puede resultar en que el programa envíe la solicitud de regreso a la URL que contiene la contraseña no confiable.Aquellos ataques de redirección se puede utilizar para crear agresivamente un enlace que pueda pasar la verificación de control del programa.

2.6 SQL Injection: parametrización de las consultas en BD

La inyección SQL es un ataque que se almacena en un programa de respaldo, ya sea en línea, móvil o computadora. El ataque de inyección SQL funciona inyectando el código SQL en palabras que cambian su concepto original.

- Problemas que pueden causar este tipo de ataques:Integridad(permite leer

la información almacenada en la BD),Confidencialidad(los archivos almacenan información confidencial) e Identificación.

- Uso de sentencias preparadas parametrizadas: En cuanto a las consultas SQL parametrizadas se crean haciendo uso de SQL estándar, pero al momento de incluir lo que están integrando los usuarios, integran construcciones de componentes, que son archivos que se colocan después.En la mayoría de las aplicaciones de desarrollo (Rails, Django, entre otros), el modelo relacional de objetos (ORM) se utiliza para la comunicación informativa y la BD.

2.7 Seguridad en AJAX

AJAX,es un método que le brinda la oportunidad de crear aplicaciones interactivas, que se ejecutan en el lado del cliente en el navegador.AJAX funciona de la siguiente manera: cuando el software se ejecuta en el explorador del cliente, la conexión se realiza en segundo plano detrás del servidor.Los beneficios que obtenemos al utilizar la tecnología AJAX en nuestro sitio son: Reducir el tráfico y Visibilidad del usuario.

3 Conclusion

- Hoy en día, la seguridad está disponible en cualquier tipo de diseño de software y, como tal, no puede dejarse como un objeto independiente sino que es transversal y multidisciplinar.
- Si los programas continúan desarrollándose con normalidad, estas diferencias serán explotadas por hackers y expertos en seguridad, que siempre van dos pasos por delante de las agencias, y se seguirán aprovechando las debilidades que podrían haberse evitado utilizando métodos como los comentados.
- Para crear programas seguros, es importante, en todas las etapas de desarrollo, tener en cuenta dos elementos clave en la creación de programas seguros como son: usuarios y atacantes.
- Sabemos que la programación impecable es una meta imposible, ya que no se han establecido suficientemente buenos hábitos y sistemas de programación para promover el desarrollo robusto de sistemas de conocimiento que se perciban técnicamente, cómo se manejen y gestionen.

References

[1] Jose Manuel Ortega Candel. *Desarrollo Seguro en ingenieria del software.* first edition, 2020.

[1]