Técnico en

DESARROLLO DE SOFTWARE

Seguridad Informática

Instituto Von Neumann

Seguridad en el desarrollo de software

Unidad IV

Seguridad en el desarrollo de software

La seguridad en el desarrollo de software tiene varias aristas de aplicación. Es importante mencionar que actualmente el desarrollo de software difícilmente está basado en una sola tecnología y que en la mayoría de los casos siempre se utilizan recursos locales, web y bases de datos.

Pero, aun así, es posible asignar cada desarrollo en una esfera específica o al menos es posible clasificar individualmente cada parte a su categoría.

La seguridad en aplicaciones locales tales como de escritorio, en teléfonos móviles, relojes inteligentes, gafas de realidad aumentada y cualquier otro dispositivo que funcione con recursos locales.

Luego tenemos las aplicaciones basadas en tecnología web. En este caso se encuentran los sitios web, aplicaciones con sincronización multidispositivo, respaldos automatizados almacenados en la nube, interfaces de aplicaciones (API) tanto en tecnología REST y SOAP.

Por último, tenemos las bases de datos en donde se encuentran las bases de datos SQL (transaccionales) tales como MySQL, MSSQL y ORACLE, y las NoSQL (Not Only SQL) tales como Cassandra DB, MongoDB, Redis, y Neo4j. Pero también se

pueden mencionar directorios de archivos administrados tanto locales como en la nube tales como S3 de Amazon.

Guías generales

Existen guías generales que pueden ser aplicadas en las tres categorías mencionadas anteriormente. Estas guías pueden apoyar al desarrollador para aplicar prácticas de seguridad mínimas pero necesarias en el software.

Es importante mencionar que estas prácticas no garantizan una protección al 100% dado que tal objetivo no es realista. Pero la falta de dichas prácticas de seguridad ofrece a los atacantes una mayor superficie de ataque y por lo tanto el riesgo aumenta considerablemente. Dado que estas prácticas pueden ser útiles para detener los ataques más básicos.

Autenticación

Es el proceso de verificar que un usuario sea quien dice ser. Existen 3 factores:

- ¿Qué tienes?: tokens, correo, códigos QR
- ¿Qué sabes?: contraseña, pin
- ¿Quién eres?: huella digital, iris, identificación facial

Autenticación de dos factores (A2F)

La A2F es la autenticación utilizando 2 de los métodos mencionados.

Ejemplo:

- Usuario y contraseña (¿Qué sabes?)
- Token en aplicación de autenticación (¿Qué tienes?)

Aplicaciones de autenticación

1. Google Authenticator App
2. Duo Mobile
3. Microsoft Authenticator

4. SMS
5. Correo electrónico
6. Facebook Authenticator

A2F, A3F, A4F o A5F

1. Es posible aumentar la cantidad de factores.
2. Se debe mantener el balance entre seguridad y accesibilidad del usuario.
3. A mayor riesgo, más capas de seguridad para reducirlo.

Factores adicionales

1. Ubicación: a través de IP, información del sistema o GPS.
2. Tiempo: Hora de acceso restringidas, operaciones habilitadas en horario específico.
3. Comportamiento: Utilizando Machine Learning, perfila el comportamiento del usuario. Buscar comportamientos anómalos.

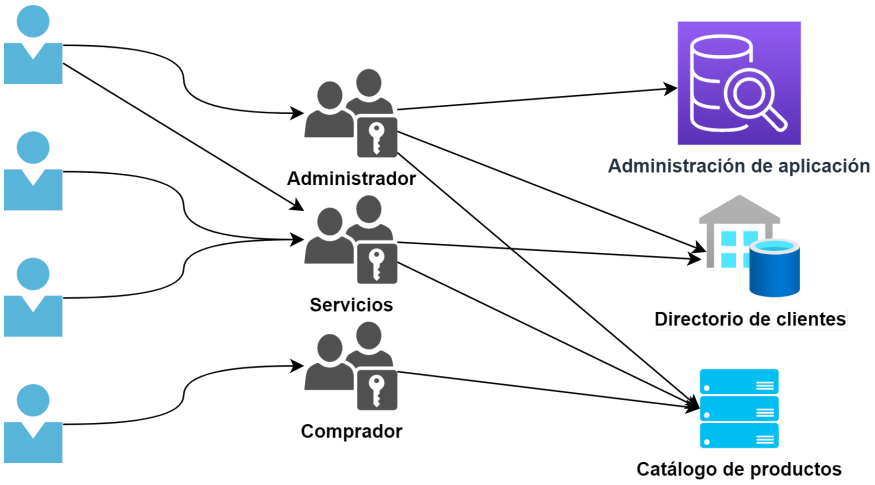
Autorización (RBAC y UBAC)

Role-based Access (RBAC):

El acceso a recursos y funcionalidades de la aplicación es brindado por el nivel de acceso del ROL que el usuario posee.

User-based Access (UBAC):

Los permisos son manejados a nivel de cada usuario.



Manejo de tipos de datos

El manejo de tipos de datos es importante, dado que cuando se solicitan información de los usuarios se debe limitar qué forma tiene esta información. Por ejemplo, si se solicita un número de teléfono se debe validar que el dato ingresado tenga el formato de un número de teléfono. De igual manera se puede mencionar los correos electrónicos.

Whitelist y Blacklist

De igual manera cuando se solicitan los datos, no solo basta con validar que tengan un formato específico, también es importante restringir algunos caracteres que presentan una amenaza para la seguridad. Esto toma más relevancia en campos de texto libre, tales como cajas de comentarios, pero no se debe olvidar también aplicarlo en todos los campos de los formularios.

Se tiene las whitelist, que permiten filtrar los datos para que todo lo que se encuentre en la lista sea aceptado. Y se tienen las blacklist que, al contrario, eliminan todo lo que está en la lista y dejan pasar todo el resto. Como se puede observar, las listas blancas son mucho más restrictivas que las negras.

Validación de autenticación en datos sensibles

En una aplicación, cuando el usuario quiere ver, modificar o eliminar alguna información sensible, es de vital importancia generar una re-autenticación. Esto es de mayor importancia cuando se trabaja con sesiones que no caducan o tienen un tiempo de vida extendido. Por ejemplo, cuando el usuario quiere modificar la contraseña.

Frameworks con seguridad integrada

Existen muchos Frameworks que fueron desarrollados con muchas de las buenas prácticas de seguridad integradas. Es aconsejable utilizar este tipo de Frameworks que dan herramientas al desarrollador de software para que, de una manera más simple, pueda aplicar las buenas prácticas de seguridad aceptadas por la industria. Qué Framework elegir depende mucho de qué tipo de aplicación se esté trabajando.

Seguridad en Aplicaciones locales

Las siguientes prácticas están enfocadas directamente al desarrollo de aplicaciones locales, pero es posible que algunas puedan ser aplicadas también en las otras categorías, dependiendo del contexto.

Limpieza de archivos temporales

Las aplicaciones generan mucha información en base a las acciones de los usuarios. También se escriben logs y ciertos archivos temporales como los Thumbnail, que son las versiones en vista previa de los archivos. Toda esta información está almacenada localmente en diversas carpetas, según el sistema operativo y su versión.

Claro que el usuario puede hacer limpieza de dichos archivos, pero es importante que las aplicaciones tengan un ciclo de limpieza en el que eliminen dichos archivos temporales generados por la aplicación.

Permisos de aplicación

Cuando se desarrolla una aplicación, usualmente los SDK permiten manejar qué tipos de permisos necesita la aplicación para ejecutarse sin inconvenientes. Lo importante en este concepto es poder desarrollar con el mínimo de permisos posibles, e informar al usuario por qué se necesita cada uno de estos permisos.

Privacidad visual, datos sensibles ocultos

Usualmente los usuarios tienden a utilizar las aplicaciones en entornos en donde no existe la privacidad y las personas alrededor puede ver que información está manejando. Por tal motivo es importante que los datos sensibles puedan ser ocultados por el usuario cuando este lo desee. Por ejemplo, en una aplicación bancaria, permite al usuario ocultar datos tales como números de cuenta, montos y direcciones.

Copias de seguridad locales y en línea

Como ya se ha comentado anteriormente, la seguridad no es un objetivo que se pueda alcanzar al 100%. Por tal motivo, las copias de seguridad son un factor clave para recuperar un sistema dañado. Se pueden tener dos estrategias dependiendo de la localidad en donde se almacenen las copias de seguridad locales y en línea.

Cada una es importante dependiendo que tipo de información se maneje y los requisitos que tenga el negocio sobre el manejo de la información.

Seguridad en Aplicaciones web

Las siguientes prácticas están enfocadas directamente al desarrollo de aplicaciones web, pero es posible que algunas puedan ser aplicadas también en las otras categorías, dependiendo del contexto.

Cookies

Las cookies son herramientas de mucha utilidad para los desarrolladores, dado que permiten dar al usuario un sentimiento de comodidad y configurabilidad. Dado que es posible almacenar los gustos de los usuarios para poder desplegar de la misma manera los sitios web. Pero, es importante no almacenar información sensible en las cookies, dado que pueden ser robadas.

HTTPS

Este es un tema clave para la seguridad web, es imperativo utilizar los certificados web para los sitios que se desarrollen. Mucho más aún si los datos que se van a manejar son de características sensibles. Y si se van a manejar tarjetas de créditos se debe utilizar PCI DSS (Payment Card Industry Data Security Standard).

Verificación de Frameworks utilizados

Los Framework son una herramienta de mucha utilidad, tal como se explicó anteriormente. Pero es importante seleccionar muy bien que Framework se utilizara para el desarrollo y asegurarse de estar al pendiente en las noticias de seguridad informática sobre vulnerabilidades que se hayan encontrado.

Seguridad en Bases de datos

Las siguientes prácticas están enfocadas directamente a la seguridad en las bases de datos.

Cifrado de datos sensibles

Cuando se almacenan datos de entidades en las bases de datos, a menudo se almacenan como texto plano o números. Usualmente los datos no tienen una capa de cifrado. Pero es importante que la información sensible sea almacenada en forma cifrada, de esta manera, aunque el sistema sufra un robo de información los datos especialmente sensibles tendrán una capa extra de protección.

Se debe mencionar que, al igual que las demás prácticas de seguridad, se debe evaluar qué información amerita ser cifrada, dado que el proceso de encriptar y desencriptar consume recursos del sistema y aumenta el tiempo de espera de los usuarios. Por tal motivo, aplicarlo a todos los datos de la base de datos, es en general una mala práctica.

Acceso controlado

Los DBMS (Database Management Systems) tienen integrado en las herramientas de administración un módulo de autenticación y autorización, usualmente basados en RBAC. Es importante que hacer uso de dicha herramienta y dar acceso solo a los recursos necesarios a las entidades necesarias. Y que exista un responsable DBA (Database Administrator) a cargo de estos permisos.

Respaldos (en frío y en caliente)

Los respaldos o copias son un componente clave, tal como se ha descrito en las prácticas de desarrollo local. En la categoría de bases de datos toman una mayor importancia dado que el objetivo principal es el almacenamiento de los

datos. Por tal motivo es importante remarcar el uso. En las bases de datos existen dos tipos de respaldos principales, divididos por la estrategia y el momento en que se ejecutan.

Los respaldos en frío son los que se ejecutan cuando se apaga el servidor y ningún usuario tiene acceso a los recursos. La copia se ejecuta y no tiene inconsistencias.

Los respaldos en caliente se ejecutan mientras el servidor está en funcionamiento y por tanto los usuarios pueden estar activos. Esta estrategia puede generar inconsistencias en los datos y puede generar lentitud en el servidor dado que utiliza recursos.

Protección contra comandos (drop y delete)

En sistemas como MySQL viene implementado por defecto la protección contra el uso de comandos drop y delete. Y requieren permisos especiales para su ejecución.

Referencias:

- http://windows.microsoft.com/es-XL/windows/frequently-asked-questions-about-malicious-software
- https://www.microsoft.com/es-xl/security/pc-security/protect-pc.aspx