Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Diseño Orientado a Objetos

Erick Alejandro Campos Rivero

1506449

Tipos de aplicaciones web y vulnerabilidades

Prof.: Miguel Ángel Salazar Santillán

¿Qué es una aplicación web?

Las aplicaciones web reciben este nombre porque se ejecutan en el internet. Es decir que los datos o los archivos en los que trabajas son procesados y almacenados dentro de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en tu computador.

El concepto de aplicaciones web está relacionado con el almacenamiento en la nube. Toda la información se guarda de forma permanente en grandes servidores de internet y nos envían a nuestros dispositivos o equipos los datos que requerimos en ese momento, quedando una copia temporal dentro de nuestro equipo.

Ventajas de las aplicaciones Web

- Muchas aplicaciones web son gratuitas.
- Puedes acceder a tu información en cualquier lugar y momento.
- No dependes de tu computador o de algún equipo específico ya que el contenido está almacenado en la web.
- Muchas de las aplicaciones web permiten que varias personas trabajen simultáneamente en ellas.
- Los documentos y archivos no se te van a perder ni borrar a menos que tú así lo quieras.

Tipos de Desarrollo App Web

1. Aplicaciones web estáticas

Este tipo de web App muestran muy poca información y no suelen variar mucho (aunque pueden mostrar en alguna parte de los mismos objetos en movimiento, como por ejemplo (banners, GIF animados, vídeos, etc.) Por regla general suelen estar desarrolladas en lenguaje HTML y CSS y pueden ser creadas en **plataformas de desarrollo** como por ejemplo **AppYourSelf** o **Monincube**.

2. Aplicaciones web dinámicas

Una aplicación web dinámica es mucho más compleja de crear y desarrollar a nivel técnico que una App web estática, ya que utilizan bases de datos para cargar la información para que los contenidos de la App Web se vayan cargando y actualizando cada vez que el usuario accede a la misma. Las Apps web dinámicas cuentan por lo general con un panel de administración (llamado CMS) desde dónde los administradores pueden corregir, modificar y cambiar los contenidos, ya sean textos o imágenes.

3. **Portales para Aplicaciones Web**

Un portal móvil App Web, es un sitio o página web para dispositivos móviles, es decir, es muy similar a un sitio web normal, pero diseñado para las pantallas de este tipo de dispositivos que son más pequeñas, de tal forma que los contenidos se optimizan para ajustarse a estos requisitos. Esto facilita la navegación y lectura a través de dispositivos móviles ya que de lo contrario un sitio Web normal sería muy incómodo de visitar y visualizar.

4. Tienda online para aplicaciones Web

El desarrollo es similar al de un sitio web orientado al e-commerce. Es decir, una App Web basada en una tienda online, o también denominada M-Commerce o comercio móvil ya que lleva todas las transacciones a nivel de poder ser ejecutadas desde cualquier dispositivo móvil. El desarrollo de una tienda online App Web es más complicado que el de una App Web estática o una App Web Dinámica, porque debe contar con una pasarela de pagos electrónicos a través de tarjeta de crédito, PayPal, u otro método de pago, además de tirar de bases de datos. Este tipo de Apps también cuenta con un CMS o panel de gestión que el desarrollador App también deberá crear, desde el cual, se pueden subir los productos, actualizarlos o eliminarlos, gestionar los pedidos y pagos, etc.

5. **Aplicaciones web animadas**

Es una de las tecnologías más usadas por diseñadores, creativos y desarrolladores App debido a que permite presentar los contenidos de la App Web con efectos animados de todo tipo y diseños muy creativos y modernos. Utiliza tecnología Flash para las animaciones. Tienen un inconveniente bastante importante, y es que su posicionamiento Seo es mucho más complicado ya que este tipo de tecnología no es la más adecuada para ello. Los motores de búsqueda no Indexan este tipo de formatos correctamente.

VULNERABILIDADES DE UNA APLICACIÓN WEB

Los sitios web pueden ser blanco de ataques cibernéticos, como por ejemplo hackers, extorsionadores y se puede sufrir el robo de información de los usuarios: pérdidas financieras, convertir el sitio en descarga de programas maliciosos.

1- ***Inyección**:* Existen distintos tipos de inyecciones: SQL, LDAP, XPath, XSLT. HTML, XML. Ocurre cuando datos son proporcionados por el usuario y se envían y son interpretados como parte de una orden y consulta. Se interrumpe el intérprete para que se ejecuten comandos mal intencionado proporcionando datos modificados. Pueden crear, modificar o borrar información de una aplicación.

2- **XSS - *Cross-site scripting:*** Permite a una tercera persona inyectar en páginas web visitadas por el usuario código JavaScript o en otro lenguaje similar. Es posible encontrar una vulnerabilidad de Cross-Site Scripting en aplicaciones que tengan entre sus funciones presentar la información en un navegador web u otro contenedor de páginas web. Puede ser utilizado para robar información delicada, secuestrar sesiones de usuario, y comprometer el navegador, subyugando la integridad del sistema. **Directa**: este tipo de XSS comúnmente filtrado, y consiste en insertar código HTML peligroso en sitios que lo permitan; incluyendo así etiquetas como <script> o <iframe>. **Indirecta**: este tipo de XSS consiste en modificar valores que la aplicación web utiliza para pasar variables entre dos páginas, sin usar sesiones y sucede cuando hay un mensaje o una ruta en la URL del navegador, en una cookie, o cualquier otra cabecera HTTP.

3- **Exposición de datos sensibles**: Muchas aplicaciones web no protegen adecuadamente datos sensibles tales como números de tarjetas de crédito o credenciales de autenticación. Los atacantes pueden robar o modificar tales datos para llevar a cabo fraudes, robos de identidad u otros delitos. Los datos sensibles requieren de métodos de protección adicionales tales como el cifrado de datos, así como también de precauciones especiales en un intercambio de datos con el navegador.

4- **Pérdida de Autenticación y Gestión de Sesiones**: Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, claves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.

5- **Referencia Directa Insegura a Objetos**: Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder datos no autorizados.

¿CÓMO SE PUEDEN PREVENIR?

***1 Inyección*:**

La opción preferida es usar una API segura la cual evite el uso de intérpretes por completo o provea una interface parametrizada. Ser cuidadoso con las APIs, como los procedimientos almacenados que son parametrizados, pero que aún pueden introducir inyecciones en el motor del intérprete. Si una API parametrizada no está disponible, se debe codificar cuidadosamente los caracteres especiales, usando la sintaxis de escape específica del intérprete.

2 XSS - Cross-site scripting:

Prevenir XSS requiere mantener los datos no confiables separados del contenido activo del navegador. La opción preferida es codificar los datos no confiables basados en el contexto HTML (cuerpo, atributo, JavaScript, CSS, o URL) donde serán ubicados.

3 Exposición de datos sensibles

Considerar las amenazas de las cuáles protegerá los datos (ej.: atacante interno, usuario externo), asegúrese de cifrar los datos sensibles almacenados o en tráfico de manera de defenderse de estas amenazas. No almacenar datos sensibles innecesariamente y asegurarse de aplicar algoritmos de cifrado fuertes y estándar así como claves fuertes y gestionarlas de formas segura. Asegurarse que las claves se almacenan con un algoritmo especialmente diseñado para protegerlas. Deshabilitar el autocompletar en los formularios que recolectan datos sensibles, también el cacheado de páginas que contengan datos sensibles.

4 Pérdida de Autenticación y Gestión de Sesiones

Cumplir con todos los requisitos de autenticación y gestión de sesiones definidos en el Application Security Verification Standard (ASVS) de OWASP. Tener un interfaz simple para los desarrolladores. Se debe realizar un gran esfuerzo en evitar vulnerabilidades de XSS que podrían ser utilizadas para robar ID de sesión.

5 Referencia Directa Insegura a Objetos:

Utilizar referencias indirectas por usuario o sesión y comprobar el acceso.