TALENTO TECH

TIC

REGIÓN 3

CAUCA – NARIÑO

LECCIÓN 1 – UNIDAD 3

Universidad Tecnológica de Pereira

faceIT

Open web application security Project (Owasp)

Es una organización internacional sin ánimo de lucro dedicada a la seguridad de las aplicaciones en la web. Dentro de sus principios se cuenta con que todos los materiales sean fácilmente accesibles en su sitio web para que cualquier persona pueda entender y mejorar la seguridad de sus propias páginas y aplicaciones web. Entre los materiales que la organización ofreces está el proyecto OWASP top 10.

El proyecto OWASP 10. Mantiene un listado de las 10 vulnerabilidades más graves de las aplicaciones web. También ayuda a los desarrolladores haciendo disponibles guias, recursos multimedia, cursos y los procedimientos y controles necesarios para mitigar esos riesgos.

Owasp es una comunidad abierta para que las personas se puedan unir y contribuir. Desde sus proyectos y con sus recursos. Al colaborar. La Comunidad puede aprovechar el conocimiento colectivo y la experiencia de sus miembros para que las soluciones estén actualizadas y sean oportunas.

La lista de los 10 riesgos es de seguridad en las aplicaciones sirve como guía para los profesionales en el desarrollo web y en la seguridad informática, así como los administradores de redes y de sistemas corporativos ayudándoles a mitigar los riesgos de seguridad críticos.

Para el año 2021 los días riesgos son: (tomado de la guia OWASP, de acceso libre)

1.**Pérdida de control de acceso.** Esta vulnerabilidad se produce cuando la aplicación insuficiente de la autorización y los controles de acceso permite que los atacantes puedan acceder a funciones o datos no autorizados. Esto puede deberse a referencias directas inseguras a objetos (IDOR), que pueden surgir cuando una aplicación no puede validar o autorizar la entrada del usuario que se utiliza como referencia directa a un objeto interno. También puede producirse debido a la falta de controles de acceso a nivel de función, cuando la aplicación solo valida los controles de acceso en la fase inicial de autenticación o autorización, pero no aplica esos controles de manera consistente en todas las funciones u operaciones de la aplicación. Un cortafuegos de aplicaciones web (WAF) puede ayudar a protegerse contra estos ataques mediante la supervisión y la aplicación de controles de acceso para impedir el acceso no autorizado a objetos o recursos confidenciales.

1.Fallas criptográficas. Este riesgo se produce debido a la protección inadecuada de los datos confidenciales durante el tránsito y en reposo. Las fallas criptográficas pueden dar lugar a infracciones de datos, acceso no autorizado a información confidencial y el incumplimiento de las normas de privacidad de datos, como el Reglamento General de Protección de Datos de la UE (RGPD) y estándares financieros como las Normas de Seguridad de Datos PCI (PCI DSS). Estos fallos pueden deberse a un almacenamiento criptográfico no seguro, al almacenamiento de datos en texto sin formato o a una gestión de claves no segura. El riesgo también puede derivar de la fuga de información, que puede provenir de la generación de números aleatorios o claves débiles, o de fallos en los protocolos criptográficos.

2.Ataques de inyección. Los fallos de inyección se producen cuando los atacantes insertan datos no fiables u hostiles en los lenguajes de consulta o comandos, o cuando la aplicación no valida, filtra o sanea los datos proporcionados por el usuario, lo que lleva a la ejecución accidental de comandos maliciosos. Esta categoría de riesgo abarca los ataques de inyección de NoSQL, comandos del sistema operativo, LDAP y SQL, además de los de cross-site scripting (XSS), en los que los atacantes inyectan scripts maliciosos del lado del cliente, como JavaScript, en páginas web que ven otros usuarios. Esto puede resultar en el robo de información confidencial, como credenciales de inicio de sesión, datos personales o cookies de sesión. Un WAF puede ayudar a detectar y bloquear los intentos de inyectar código malicioso al inspeccionar y filtrar las solicitudes entrantes, incluido el XSS reflejado (no persistente), almacenado (persistente) y basado en el módulo de objeto de documento (DOM), lo que les impide acceder a la aplicación.

4.**Diseño inseguro.** Esta es una categoría muy amplia que representa distintas debilidades expresadas como fallos de la arquitectura, o bien falta o ineficacia de los controles de seguridad. Estos fallos pueden ocurrir si una aplicación se ha diseñado para depender de procesos poco seguros de por sí, o cuando no se implementan los controles de seguridad necesarios para defenderse frente a ataques específicos. Estos riesgos pueden reducirse mediante un mayor uso del modelado de amenazas, patrones de diseño seguros y arquitecturas de referencia.

5.Configuración de seguridad incorrecta. La falta de refuerzo de la seguridad en los marcos de aplicaciones web, las plataformas, los servidores o los controles de seguridad puede dar lugar al acceso no autorizado, la exposición de información confidencial u otras vulnerabilidades de seguridad. Los riesgos debidos a configuraciones de seguridad erróneas también pueden deberse a permisos configurados incorrectamente en los servicios en la nube, o a la instalación o habilitación de funciones innecesarias, como puertos, servicios, cuentas o privilegios no utilizados. La configuración errónea tanto de las aplicaciones web como de las API es un riesgo importante, ya que los principales proveedores de nube tienen distintas posiciones de seguridad de forma predeterminada y la arquitectura está cada vez más descentralizada y distribuida en una estructura multinube.

6.Componentes vulnerables y desactualizados. El uso de componentes obsoletos, sin parches o vulnerables (como bibliotecas, marcos o complementos) puede exponer las aplicaciones a fallos de seguridad conocidos, lo que aumenta el riesgo de explotaciones. Estos riesgos pueden estar causados por software no compatible o no actualizado, como el del sistema operativo (SO), el servidor web o de aplicaciones, el sistema de gestión de bases de datos (DBMS), las aplicaciones, las API y todos los componentes, entornos de ejecución y bibliotecas. Estas amenazas son especialmente peligrosas cuando las organizaciones no cuentan con medidas oportunas basadas en los riesgos para corregir o actualizar la plataforma, los marcos y las dependencias subyacentes de un sistema, lo que deja a este expuesto de forma innecesaria a riesgos conocidos durante días o semanas. Las cadenas de suministro de software complejas y la automatización a través de canalizaciones CI/CD aumentan el riesgo de introducir software vulnerable en la pila de TI. Un WAF puede servir como solución provisional crítica para protegerse frente a la explotación de vulnerabilidades.

8.Fallas en el software y en la integridad de los datos. Estas vulnerabilidades se producen debido a que la infraestructura y el código de las aplicaciones no pueden proteger frente a las infracciones de integridad de los datos y el software. Esto puede ocurrir cuando una aplicación se basa en complementos, bibliotecas o módulos de fuentes, repositorios y CDN que no son de confianza. También puede ocurrir durante las actualizaciones de software, modificaciones de datos confidenciales y cambios en las canalizaciones CI/CD que no se validan. Es posible que los atacantes puedan cargar sus propias actualizaciones para distribuirlas y ejecutarlas en todas las instalaciones. La deserialización poco segura (en la que una aplicación toma datos serializados no fiables y los consume sin asegurarse de que sean válidos) también forma parte de esta categoría de riesgo, lo que permite ataques como la ejecución remota de código (RCE) y la escalada de privilegios.

9.Fallas en el registro y monitoreo de la seguridad. Un acceso y supervisión inadecuados pueden obstaculizar la detección y la respuesta a tiempo a los incidentes de seguridad, lo que dificulta la identificación y la mitigación de ataques o actividades no autorizadas. Esto puede significar que los eventos auditables, como inicios de sesión, inicios de sesión fallidos y transacciones de alto valor, no se identifican ni se registran y, además, las aplicaciones no detectan ataques activos en tiempo real.

10.Server-side request forgery (SSRF). Estas vulnerabilidades se producen cuando una aplicación no valida ni sanea una dirección URL introducida por un usuario antes de extraer datos de un recurso remoto. Los atacantes pueden usar estos fallos para forzar a las aplicaciones a que accedan a destinos web maliciosos, incluso aunque estén protegidas por un cortafuegos u otro tipo de defensa. Estos ataques también pueden producirse si el recurso objetivo tiene relaciones de confianza con otros sistemas, como un servicio de metadatos en la nube o las API de back-end, lo que permite a un atacante realizar solicitudes a esos servicios de confianza y extraer información confidencial o realizar acciones no autorizadas. Para ayudar a mitigar el riesgo de SSRF, diseñe sistemas para el acceso con privilegios mínimos y use un WAF para definir explícitamente los parámetros del identificador uniforme de recursos (URI) en su política de seguridad y permitir o denegar el acceso a los hosts que pueden acceder a ellos.

La corporación OWASP no sólo tiene su proyecto top 10, sino que patrocina más de 293 proyectos enfocados en la mejora de la seguridad informática, por ejemplo, se tienen los proyectos OWASP Amass, que ha desarrollado una herramienta para que los profesionales en seguridad informática puedan mapear las redes y validar qué conexiones y puertos están expuestos de forma ágil. Entre otros proyectos se encuentran guias y documentos que establecen metodologías de pruebas de seguridad, metodologías de identificación de fallas y de desarrollo seguro en entornos web.