CRUZ GUTIERREZ JEMIMAH 2-B INS.COMPUTACIONALES

¿Qué es OWASP TOP TEN?
El TOP 10 representa los 10 riesgos y amenazas más prominentes y peligrosos para las aplicaciones. OWASP se refiere al Top 10 como un "documento de concientización" y recomienda que todas las organizaciones incorporen el informe en sus procesos para mitigar los riesgos de seguridad. Una cosa importante para recordar es que no es un estándar. Las organizaciones pueden definir la matriz en función de su propio entorno. El informe es elaborado por un equipo de expertos en seguridad de todo el mundo que analizan datos provenientes de varias organizaciones para posteriormente publicarlo.

1. Autenticación rota.

Las llamadas de administración de sesión y autenticación implementadas incorrectamente pueden ser un gran riesgo para la seguridad. Si los atacantes notan estas vulnerabilidades, pueden asumir fácilmente las identidades de los usuarios legítimos.
Ejemplo: una aplicación permite cambiar una clave principal, y cuando esta clave se cambia al registro de otro usuario, la cuenta de ese usuario se puede ver o modificar.
Solución: una solución de prueba de seguridad de aplicaciones interactivas (IAST), como Seeker ® , puede ayudarlo a detectar sin esfuerzo la falsificación de solicitudes entre sitios o el almacenamiento inseguro de sus datos confidenciales. También identifica cualquier lógica incorrecta o faltante que se utilice para manejar tokens web JSON. Las pruebas de penetración pueden servir como complemento manual de las actividades de IAST, lo que ayuda a detectar controles de acceso no deseados. Se pueden justificar cambios en la arquitectura y el diseño para crear límites de confianza para el acceso a los datos.

2. Fallas criptográficas (A02:2021).

Las fallas criptográficas ocurren cuando se comprometen datos importantes almacenados o transmitidos (como un número de seguro social).
Ejemplo: una institución financiera no protege adecuadamente sus datos confidenciales y se convierte en un blanco fácil para el fraude con tarjetas de crédito y el robo de identidad.
Solución: los verificadores de Seeker pueden escanear tanto la fuerza de cifrado inadecuada como las claves criptográficas débiles o codificadas, y luego identificar cualquier algoritmo criptográfico roto o riesgoso. El módulo de criptografía Black Duck® muestra los métodos criptográficos utilizados en el software de código abierto (OSS) para que puedan evaluarse más a fondo en cuanto a su solidez. Tanto las pruebas de seguridad de aplicaciones estáticas (SAST) de Coverity® como el análisis de composición de software (SCA) de Black Duck tienen verificadores que pueden proporcionar una instantánea de "punto en el tiempo" a nivel de código y componente. Sin embargo, complementar con IAST es fundamental para proporcionar monitoreo y verificación continuos para garantizar que los datos confidenciales no se filtren durante las pruebas integradas con otros componentes de software internos y externos.

3. Inyección (A03:2021).

Esencialmente, una inyección de código ocurre cuando un atacante envía datos no válidos a una aplicación web para hacer que la aplicación haga algo para lo que no fue diseñada.
Ejemplo: una aplicación utiliza datos que no son de confianza al construir una llamada SQL vulnerable.
Solución: la inclusión de las herramientas SAST e IAST en su canalización de integración continua/entrega continua (CI/CD) ayuda a identificar fallas de inyección tanto a nivel de código estático como dinámicamente durante las pruebas de tiempo de ejecución de la aplicación. Las herramientas modernas de prueba de seguridad de aplicaciones (AST) como Seeker pueden ayudar a proteger la aplicación de software durante las diversas etapas de prueba y verificar una variedad de ataques de inyección (además de las inyecciones de SQL). Por ejemplo, puede identificar inyecciones de NoSQL, inyecciones de comandos, inyecciones de LDAP, inyecciones de plantillas e inyecciones de registros. Seeker es la primera herramienta que proporciona un nuevo verificador dedicado diseñado para detectar específicamente las vulnerabilidades de Log4Shell., determine cómo está configurado Log4J, pruebe cómo se comporta realmente y valide (o invalide) esos hallazgos con su motor de verificación activa patentado.

4. Diseño Inseguro (A04:2021).

El diseño inseguro es una nueva categoría para 2021 que se enfoca en los riesgos relacionados con fallas de diseño. A medida que las organizaciones continúan "girando hacia la izquierda", el modelado de amenazas, los patrones y principios de diseño seguro y las arquitecturas de referencia no son suficientes.
Ejemplo: una cadena de cines que permite descuentos en reservas de grupos requiere un depósito para grupos de más de 15 personas. Los atacantes amenazan con modelar este flujo para ver si pueden reservar cientos de asientos en varios cines de la cadena, causando así miles de dólares en ingresos perdidos.
Solución: Seeker IAST detecta vulnerabilidades y expone todas las llamadas a funciones, servicios y API entrantes y salientes en aplicaciones web, en la nube y basadas en microservicios altamente complejas. Al proporcionar un mapa visual del flujo de datos y los puntos finales involucrados, se aclara cualquier debilidad en el diseñode la aplicación, lo que ayuda en los esfuerzos de modelado de amenazas y pruebas de penetración.

5. Configuración incorrecta de seguridad (A05:2021).

La antigua categoría de entidades externas ahora forma parte de esta categoría de riesgo, que asciende desde el puesto número 6. Las configuraciones incorrectas de seguridad son debilidades de diseño o configuración que resultan de un error o deficiencia de configuración.
Ejemplo: una cuenta predeterminada y su contraseña original aún están habilitadas, lo que hace que el sistema sea vulnerable a la explotación.
Solución: Soluciones como Coverity SAST incluyen un verificador que identifica la exposición de información disponible a través de un mensaje de error. Las herramientas dinámicas como Seeker IAST pueden detectar la divulgación de información y las configuraciones de encabezado HTTP inapropiadas durante las pruebas de tiempo de ejecución de la aplicación.

6. Componentes vulnerables y obsoletos (A06:2021).

Esta categoría se relaciona con componentes que plantean riesgos de seguridad conocidos y potenciales, en lugar de solo los primeros. Los componentes con vulnerabilidades conocidas, como CVE, deben identificarse y parchearse, mientras que los componentes obsoletos o maliciosos deben evaluarse para determinar su viabilidad y el riesgo que pueden presentar.

Ejemplo: debido al volumen de componentes utilizados en el desarrollo, es posible que un equipo de desarrollo no conozca o no comprenda todos los componentes utilizados en su aplicación, y algunos de esos componentes pueden estar desactualizados y, por lo tanto, vulnerables a ataques.Solución: las herramientas de análisis de composición de software (SCA) como Black Duck se pueden usar junto con el análisis estático e IAST para identificar y detectar componentes obsoletos e inseguros en una aplicación. IAST y SCA funcionan bien juntos, brindando información sobre cómo se están utilizando realmente los componentes vulnerables u obsoletos. Seeker IAST y Black Duck SCA juntos van más allá de identificar un componente vulnerable, descubriendo detalles como si ese componente está cargado actualmente por una aplicación bajo prueba. Además, métricas como la actividad del desarrollador, la reputación del colaborador y el historial de versiones pueden dar a los usuarios una idea del riesgo potencial que puede representar un componente obsoleto o malicioso.

7. Fallas de Identificación y Autenticación (A07:2021).

Ahora incluye CWE relacionados con fallas de identificación. Específicamente, las funciones relacionadas con la autenticación y la administración de sesiones, cuando se implementan incorrectamente, permiten a los atacantes comprometer contraseñas, palabras clave y sesiones, lo que puede conducir al robo de la identidad del usuario y más.

Ejemplo: una aplicación web permite el uso de contraseñas débiles o fáciles de adivinar (es decir, "contraseña1").

Solución: la autenticación multifactor puede ayudar a reducir el riesgo de cuentas comprometidas, y el análisis estático automatizado es muy útil para encontrar tales fallas, mientras que el análisis estático manual puede agregar fuerza al evaluar esquemas de autenticación personalizados. Coverity SAST incluye un verificador que identifica específicamente las vulnerabilidades de autenticación rotas. Seeker

8. Fallas de integridad de software y datos (A08:2021).
Esta es una nueva categoría para 2021 que se enfoca en actualizaciones de software, datos críticos y canalizaciones de CI/CD utilizadas sin verificar la integridad. También ahora incluida en esta entrada, la deserialización insegura es una falla de deserialización que permite a un atacante ejecutar código en el sistema de forma remota.
Ejemplo: una aplicación de serializa los objetos hostiles proporcionados por el atacante, lo que la expone a la vulnerabilidad.
Solución: las herramientas de seguridad de aplicaciones ayudan a detectar fallas de deserialización y las pruebas de penetración pueden validar el problema. Seeker IAST también puede verificar la deserialización insegura y ayudar a detectar redireccionamientos inseguros o cualquier manipulación de los algoritmos de acceso al token.

9. Registro de seguridad y fallas de monitoreo (A09:2021).
Conocida anteriormente como registro y supervisión insuficientes, esta entrada subió del número 10 y se amplió para incluir más tipos de fallas. El registro y la supervisión son actividades que deben realizarse en un sitio web con frecuencia; de no hacerlo, el sitio queda vulnerable a actividades más graves y comprometedoras.
Ejemplo: los eventos que se pueden auditar, como inicios de sesión, inicios de sesión fallidos y otras actividades importantes, no se registran, lo que genera una aplicación vulnerable.
Solución: después de realizar las pruebas de penetración, los desarrolladores pueden estudiar los registros de prueba para identificar posibles deficiencias y vulnerabilidades. Coverity SAST y Seeker IAST pueden ayudar a identificar excepciones de seguridad no registradas.

10. Falsificación de solicitud del lado del servidor (A10:2021).
Una nueva categoría este año, una falsificación de solicitud del lado del servidor (SSRF) puede ocurrir cuando una aplicación web obtiene un recurso remoto sin validar la URL proporcionada por el usuario. Esto permite que un atacante haga que la aplicación envíe una solicitud manipulada a un destino inesperado, incluso cuando el sistema está protegido por un firewall, una VPN o una lista de control de acceso a la red adicional. La gravedad y la incidencia de los ataques SSRF están aumentando debido a los servicios en la nube y la mayor complejidad de las arquitecturas.

Ejemplo: si una arquitectura de red no está segmentada, los atacantes pueden usar los resultados de la conexión o el tiempo transcurrido para conectarse o rechazar las conexiones de carga útil de SSRF para mapear las redes internas y determinar si los puertos están abiertos o cerrados en los servidores internos.