Departamento de Ciencias de la Computación
Universidad de las Fuerzas Armadas - ESPE

Taller de Investigación

Tema: OWASP API Security Top 10 - 2023

Nombres:
Yeshua Amador Chiliquinga Amaya
Cesar Ignacio Loor Mercado
Jose Migel Sanmartin Galan

Carrera / Asignatura: Ingeniería de Software / Ingeniería de Seguridad de Software
NRC: 2540
Nombre del profesor: Walter Fuertes, PhD

Fecha de presentación: 2 de diciembre del 2024

Índice

1 Introducción

1.1 Objetivo

El objetivo de este trabajo es presentar el OWASP API Security Top 10 - 2023, una lista de las 10 vulnerabilidades más críticas en APIs, y analizar cada una de ellas, proporcionando ejemplos y recomendaciones para prevenir y mitigarlas.

1.2 Objetivos Específicos

- Presentar el OWASP API Security Top 10 - 2023 y sus categorías.
- Analizar cada una de las vulnerabilidades, proporcionando ejemplos y recomendaciones para prevenir y mitigarlas.
- Discutir las herramientas OWASP que se pueden utilizar para resolver vulnerabilidades en APIs.
- Proporcionar recomendaciones generales para implementar seguridad en APIs.

2 Desarrollo

2.1 ¿Qué es?

En ciberseguridad, **OWASP** (no OSWAP) es el acrónimo de *Open Web Application Security Project*, una comunidad en línea sin fines de lucro que se dedica a mejorar la seguridad del software. OWASP proporciona recursos, herramientas, estándares y documentación abierta para ayudar a desarrolladores, empresas y profesionales de seguridad a proteger las aplicaciones web contra vulnerabilidades comunes.

El proyecto más conocido de OWASP es el **OWASP Top Ten**, una lista que identifica las diez vulnerabilidades más críticas en aplicaciones web, como inyección SQL, fallos de autenticación y cross-site scripting (XSS). OWASP es fundamental para orientar las mejores prácticas en el desarrollo seguro y la evaluación de riesgos en aplicaciones web.

Características de OWASP

- **Comunidad abierta y colaborativa:** OWASP es un proyecto sin fines de lucro basado en contribuciones voluntarias de expertos en se-

guridad de todo el mundo.

- **Recursos accesibles y gratuitos:** Proporciona guías, herramientas, estándares y documentación para mejorar la seguridad en el desarrollo de aplicaciones, todo disponible libremente.
- **OWASP Top Ten:** Lista de las diez vulnerabilidades más críticas en aplicaciones web, usada globalmente como referencia para pruebas y desarrollo seguro.
- **Enfoque en aplicaciones web:** OWASP se especializa en la seguridad de software web, incluyendo aplicaciones móviles y APIs.
- **Promoción de buenas prácticas:** Facilita metodologías para diseñar, desarrollar y mantener aplicaciones con un enfoque en seguridad desde el inicio.
- **Herramientas y proyectos diversos:** Mantiene numerosos proyectos como OWASP ZAP (herramienta de análisis de seguridad), Dependency-Check, entre otros.
- **Actualización constante:** Los recursos y listas se actualizan regularmente para reflejar nuevas amenazas y técnicas de ataque.

3 OWASP API Security Top 10 - 2023

El Open Web Application Security Project (OWASP) es una organización sin fines de lucro que se dedica a mejorar la seguridad de las aplicaciones web y las API. Una de sus contribuciones más reconocidas es el "OWASP Top 10", una lista que identifica las diez vulnerabilidades de seguridad más críticas en aplicaciones web. Este informe se actualiza regularmente para reflejar las amenazas más relevantes.

En 2019, OWASP reconoció la creciente importancia de las API y desarrolló una lista específica de vulnerabilidades para este tipo de servicios. La versión 2023 del OWASP API Security Top 10 identifica las principales amenazas que enfrentan las API modernas.

3.1 1. API1:2023 – Autorización de Nivel de Objeto Rota (BOLA)

- **Descripción:** Ocurre cuando una API no verifica adecuadamente si un usuario tiene permiso para acceder a un objeto específico, permitiendo

que atacantes manipulen identificadores de objetos para acceder a datos de otros usuarios.

- **Impacto:** Acceso no autorizado a información confidencial y posibilidad de realizar acciones no autorizadas, como modificar o eliminar datos.
- **Ejemplo:** Un atacante autenticado como el usuario A accede o modifica los datos del usuario B mediante la manipulación del ID del objeto.
- **Causas Raíz y Prevención:**
  - Falta de controles de acceso a nivel de objeto.
  - Implementar políticas de acceso estrictas y controles en la capa lógica de la aplicación.
  - Realizar pruebas automatizadas para detectar fallas de autorización.

3.2 2. API2:2023 – Autenticación Rota

- **Descripción:** Implementaciones deficientes de autenticación, como contraseñas débiles o manejo inseguro de tokens.
- **Impacto:** Compromiso de cuentas de usuario, robo de datos y transacciones no autorizadas.
- **Ejemplo:** Uso de credenciales robadas o filtradas para acceder a una API.
- **Causas Raíz y Prevención:**
  - Requisitos de contraseñas débiles.
  - Implementar autenticación multifactor y políticas de contraseñas robustas.
  - Monitorear intentos de acceso y aplicar bloqueos por tasa.

3.3 3. API3:2023 – Autorización de Nivel de Propiedad de Objeto Rota

- **Descripción:** Ocurre cuando no se aplican controles adecuados en propiedades específicas de un objeto.

- **Impacto:** Escaladas de privilegios o pérdida de integridad de datos.
- **Ejemplo:** Cambiar el plan de suscripción modificando la propiedad `"account-type"`.
- **Causas Raíz y Prevención:**
  - Falta de validación de autorización a nivel de propiedad.
  - Implementar controles detallados y validaciones estrictas por propiedad.

3.4 4. API4:2023 – Consumo de Recursos sin Restricciones

- **Descripción:** Falta de limitaciones en el uso de recursos puede ser explotado para agotar el sistema.
- **Impacto:** Denegación de servicio (DoS), costos operativos altos y degradación del rendimiento.
- **Ejemplo:** Un atacante envía múltiples solicitudes que consumen muchos recursos.
- **Causas Raíz y Prevención:**
  - Ausencia de límites de tasa y cuotas.
  - Implementar mecanismos de limitación y monitoreo.
  - Usar gateways de API para control de acceso.

3.5 5. API5:2023 – Autorización de Nivel de Función Rota

- **Descripción:** Controles de acceso inadecuados en funciones específicas.
- **Impacto:** Acceso a funciones administrativas o sensibles sin autorización.
- **Ejemplo:** Un usuario accede a un endpoint de administrador sin permisos.
- **Causas Raíz y Prevención:**
  - Falta o mal implementación de controles de acceso.
  - Definir roles y permisos claramente y validarlos por función.

3.6 6. API6:2023 – Falsificación de Solicitudes del Lado del Servidor (SSRF)

- **Descripción:** La API permite que el servidor realice solicitudes a destinos maliciosos.
- **Impacto:** Acceso a recursos internos, escaneo de puertos y exfiltración de datos.
- **Ejemplo:** Solicitud maliciosa que accede a recursos internos del servidor.
- **Causas Raíz y Prevención:**
  - Falta de validación de URLs proporcionadas.
  - Usar listas blancas y validar estrictamente las URLs.

3.7 7. API7:2023 – Configuración de Seguridad Incorrecta

- **Descripción:** Configuraciones inseguras o por defecto en la API o servidores.
- **Impacto:** Exposición de datos sensibles, acceso no autorizado y explotación de vulnerabilidades.
- **Ejemplo:** API que expone información de depuración o usa credenciales por defecto.
- **Causas Raíz y Prevención:**
  - Uso de configuraciones inseguras o por defecto.
  - Realizar auditorías, aplicar principios de configuración segura.

3.8 8. API8:2023 – Falta de Protección Frente a Amenazas Automatizadas

- **Descripción:** Ausencia de protección contra bots o ataques automatizados.
- **Impacto:** Explotación de flujos de negocio, fraude y agotamiento de recursos.

- **Ejemplo:** Bots compran entradas para revenderlas, afectando usuarios reales.
- **Causas Raíz y Prevención:**
  - No detección de tráfico automatizado.
  - Implementar detección de bots, CAPTCHA y límites de velocidad.

3.9 9. API9:2023 – Gestión Inadecuada del Inventario

- **Descripción:** Falta de control y documentación sobre versiones y endpoints de las APIs.
- **Impacto:** Exposición de datos por uso de versiones obsoletas o endpoints no documentados.
- **Ejemplo:** API antigua expone datos por falta de mantenimiento.
- **Causas Raíz y Prevención:**
  - Ausencia de políticas de versionado y retiro.
  - Gestionar activos y auditar periódicamente.

3.10 10. API10:2023 – Consumo No Seguro de APIs

- **Descripción:** API que consume servicios de terceros sin validar respuestas.
- **Impacto:** Violación de privacidad, robo de datos y control de cuentas.
- **Ejemplo:** Redirección a sitios maliciosos por una API comprometida.
- **Causas Raíz y Prevención:**
  - Validación insuficiente de respuestas de terceros.
  - Evaluar seguridad de APIs externas, validar y cifrar toda la comunicación.

3.11 Herramientas OWASP para Resolver Vulnerabilidades

OWASP ofrece diversas herramientas y recursos para mitigar estas vulnerabilidades:

- **OWASP Testing Guide:** Guía detallada para realizar pruebas de seguridad en aplicaciones web y API, que ayuda a identificar vulnerabilidades específicas.
- **OWASP Cheat Sheet Series:** Hojas informativas con recomendaciones prácticas y ejemplos de código para evitar vulnerabilidades comunes.
- **OWASP ZAP (Zed Attack Proxy):** Herramienta gratuita y de código abierto para pruebas de penetración y detección de vulnerabilidades en aplicaciones web y API.
- **OWASP Dependency-Check:** Analiza dependencias y bibliotecas para identificar componentes vulnerables y ayudar en la gestión de riesgos.
- **OWASP ModSecurity Core Rule Set:** Conjunto de reglas para firewalls de aplicaciones web que ayuda a bloquear ataques comunes y proteger las API.

Estas herramientas permiten a los desarrolladores y equipos de seguridad implementar medidas efectivas para prevenir y corregir fallas en sus aplicaciones. La combinación de pruebas automatizadas, análisis de código y protección en tiempo real es esencial para mantener la seguridad de las API modernas.

4 Conclusiones

La seguridad de las API es una tarea compleja que requiere una aproximación integral. La implementación de controles de acceso estrictos, el uso de herramientas de análisis de código y la actualización continua de dependencias y bibliotecas son fundamentales. Sin embargo, la seguridad de las API no se puede lograr solo con herramientas, también se requiere educación y concienciación sobre la importancia de la seguridad.

La creciente complejidad de las API y la cantidad de datos que manejan las hace especialmente vulnerables a ataques. La implementación de múltiples

capas de seguridad es esencial para proteger las API. La educación y concienciación sobre seguridad de API es necesaria para que los desarrolladores y los equipos de seguridad puedan implementar medidas efectivas para prevenir y corregir fallas en sus aplicaciones.

5 Recomendaciones

Para mejorar la seguridad de las API, se recomienda:

Implementar controles de acceso estrictos en todas las API. Utilizar herramientas de análisis estático y dinámico de código. Mantener un inventario actualizado de APIs y sus versiones. Implementar monitoreo y logging continuo de las API. Realizar pruebas de seguridad periódicas. Mantener actualizadas todas las dependencias y bibliotecas.

6 Referencias

- Arsys. (n.d.). OWASP: ¿Qué es y cómo usar esta metodología? `https://www.arsys.es/blog/owasp`
- Guía del Desarrollador OWASP — Fundamentos de Seguridad — OWASP Foundation. (n.d.). `https://owasp.org/www-project-developer-guide/release-es/fundamentos/fundamentos_seguridad/`
- Stone, N. (2024, July 5). Cómo aprender OWASP e impulsar la seguridad de sus aplicaciones - Parasoft. Parasoft. `https://es.parasoft.com/blog/getting-started-with-owasp/`
- `https://owasp.org/`
- GTI. (s. f.). Revisión del Top 10 de OWASP API 2023. `https://gti.co.cr/tendencias/revision-del-top-10-de-owasp-api-2023`