Grupo 4

Paola Lara
María Mahecha
Gustavo Ladino
Yaneth Rodríguez

Docente:
JAIDER OSPINA NAVAS

Investigación:

Análisis de Vulnerabilidades en el OWASP Top 10: Métodos de Explotación y Prevención

OWASP (Open Web Application Security Project) es una organización sin ánimo de lucro cuyo propósito fundamental es el mejoramiento de la seguridad del software. Se dedica a incrementar la visibilidad de la seguridad del software, facilitando así que las organizaciones puedan tomar decisiones fundamentadas respecto a los riesgos de seguridad. El OWASP TOP 10, publica una lista que recoge los problemas más frecuentes y críticos detectados en las aplicaciones web.

Los 10 principales riesgos de seguridad de las aplicaciones web

1. Broken Access Control: Controles de acceso deficientes.

- Naturaleza: Fallos en los controles de acceso que permiten a los usuarios acceder a recursos no autorizados.
- Causas: Implementaciones incorrectas de controles, falta de verificación de permisos, y lógica de negocio deficiente.
- Impacto Potencial: Acceso no autorizado a datos sensibles, modificación o eliminación de información crítica.
- Métodos de Explotación:
  - Manipulación de solicitudes HTTP (modificación de parámetros).
  - Ejemplo: Un atacante accede a la cuenta de otro usuario al cambiar el ID de usuario en la URL.
- Mejores Prácticas:
  - Implementar control de acceso basado en roles (RBAC).
  - Validar permisos en el servidor y no confiar en la lógica del cliente.
  - Realizar auditorías de seguridad regularmente.

2. Cryptographic Failures: Fallos en la implementación de la criptografía.

- Naturaleza: Uso inadecuado de criptografía que compromete la protección de datos sensibles.
- Causas: Algoritmos obsoletos, falta de cifrado en la transmisión de datos, o almacenamiento inseguro de claves.
- Impacto Potencial: Exposición de información sensible, como contraseñas y datos personales.

- Métodos de Explotación:
    - Intercepción de datos no cifrados usando herramientas como Wireshark.
    - Ejemplo: En 2016, el ataque a LinkedIn expuso contraseñas por hashing débil.
- Mejores Prácticas:
    - Usar algoritmos de cifrado modernos (AES, RSA).
    - Implementar TLS para la transmisión de datos.
    - Almacenar contraseñas usando técnicas de hashing seguras (bcrypt, Argon2).

3. Injection: Inyección de código.

Naturaleza: Permite a un atacante insertar código malicioso en consultas o comandos que la aplicación ejecuta.

- Causas: Falta de validación de entradas y uso de consultas SQL no parametrizadas.
- Impacto Potencial: Acceso no autorizado a bases de datos, ejecución de comandos no autorizados, fuga de información.
- Métodos de Explotación:
    - Inyección SQL, por ejemplo: '; DROP TABLE users; --.
    - Ejemplo: En 2017, un ataque de inyección SQL expuso datos de estudiantes en una universidad.
- Mejores Prácticas:
    - Usar consultas parametrizadas o un ORM.
    - Validar y sanitizar entradas de usuario.
    - Implementar un WAF (Web Application Firewall).

4. Insecure Design: Falta de buena arquitectura de seguridad.

- Naturaleza: Falta de una buena arquitectura de seguridad en el diseño de la aplicación.
- Causas: No realizar análisis de amenazas durante el diseño, omitir controles de seguridad.
- Impacto Potencial: Vulnerabilidades introducidas que pueden ser explotadas.
- Métodos de Explotación:
    - Explotación de flujos de trabajo inadecuados para obtener acceso no autorizado.
    - Ejemplo: Sistemas de pago mal diseñados que permiten transacciones no autorizadas.
- Mejores Prácticas:
    - Realizar análisis de amenazas en la fase de diseño.
    - Aplicar principios de diseño seguro desde el inicio.
    - Involucrar expertos en seguridad en el desarrollo.
- 5. Security Mis configuration: Configuraciones de seguridad incorrectas.
- Naturaleza: Configuraciones de seguridad incorrectas en la infraestructura de la aplicación.
- Causas: Uso de configuraciones predeterminadas inseguras, falta de actualizaciones de seguridad.
- Impacto Potencial: Exposición de datos sensibles, vulnerabilidades que pueden ser explotadas fácilmente.
- Métodos de Explotación:
    - Acceso a datos sensibles a través de configuraciones inseguras.
    - Ejemplo: Exposición de un servidor Elasticsearch sin protección.
- Mejores Prácticas:

    - Realizar auditorías de configuración.
    - Desactivar servicios no utilizados.
    - Aplicar configuraciones de seguridad recomendadas.

6. Vulnerable and Outdated Components: Uso de software desactualizado.
    - Naturaleza: Uso de software desactualizado que contiene vulnerabilidades conocidas.
    - Causas: Falta de mantenimiento y actualización de componentes de software.
    - Impacto Potencial: Explotación de vulnerabilidades conocidas que comprometen la aplicación.
    - Métodos de Explotación:
        - Uso de herramientas como Metasploit para atacar componentes desactualizados.
        - Ejemplo: El ataque a Equifax fue resultado de no aplicar un parche de seguridad.
    - Mejores Prácticas:
        - Mantener un inventario de componentes y sus versiones.
        - Aplicar actualizaciones de seguridad regularmente.
        - Usar herramientas de análisis de vulnerabilidades.
7. Identification and Authentication Failures: Errores en autenticación y gestión de sesiones.
    - Naturaleza: Fallos en la autenticación y gestión de sesiones que permiten a atacantes suplantar identidades.
    - Causas: Uso de contraseñas débiles, falta de autenticación multifactor, mala gestión de sesiones.
    - Impacto Potencial: Suplantación de identidad y acceso no autorizado a datos.
    - Métodos de Explotación:
        - Ataques de fuerza bruta para adivinar contraseñas.
        - Ejemplo: Acceso no autorizado a cuentas de Twitter mediante ingeniería social.
    - Mejores Prácticas:
        - Implementar autenticación multifactor (MFA).
        - Forzar políticas de contraseñas fuertes.
        - Limitar intentos de inicio de sesión.
8. Software and Data Integrity Failures: Fallas en la validación de integridad de software.
    - Naturaleza: Falta de mecanismos para validar la integridad del software y datos.
    - Causas: Procesos inseguros para actualizaciones y falta de controles de integridad.
    - Impacto Potencial: Inyección de malware y corrupción de datos.
    - Métodos de Explotación:
        - Inyección de malware en actualizaciones no verificadas.
        - Ejemplo: Actualizaciones inseguras que permiten ejecutar código malicioso.
    - Mejores Prácticas:
        - Usar firmas digitales y hashes para validar integridad.
        - Monitorear actualizaciones de software.
9. Security Logging and Monitoring Failures: Falta de registro y monitoreo de eventos de seguridad.
    - Naturaleza: Inadecuada gestión de logs y falta de monitoreo de eventos de seguridad.
    - Causas: No registrar eventos relevantes y falta de alertas para actividades sospechosas.
    - Impacto Potencial: Dificultades para detectar ataques y responder a incidentes.
    - Métodos de Explotación:

    - Dificultades en la detección de ataques prolongados sin registros adecuados.
    - Ejemplo: Ataques sin ser detectados debido a la falta de monitoreo.
- Mejores Prácticas:
    - Implementar registros detallados de eventos.
    - Establecer alertas para actividades sospechosas.
    - Realizar auditorías de seguridad periódicas.

10. Server-Side Request Forgery (SSRF): Solicitudes enviadas desde el servidor a recursos.

- Naturaleza: Permite que un atacante envíe solicitudes desde el servidor a recursos internos o externos.
- Causas: No validar adecuadamente las URLs y datos de entrada.
- Impacto Potencial: Acceso no autorizado a servicios internos y divulgación de información sensible.
- Métodos de Explotación:
    - Envío de solicitudes a servicios internos al manipular parámetros de entrada.
    - Ejemplo: Explotación de errores de validación que permiten acceder a recursos internos.
- Mejores Prácticas:
    - Restringir acceso a recursos internos.
    - Validar y sanitizar entradas de URL.
    - Usar controles de seguridad en la red.