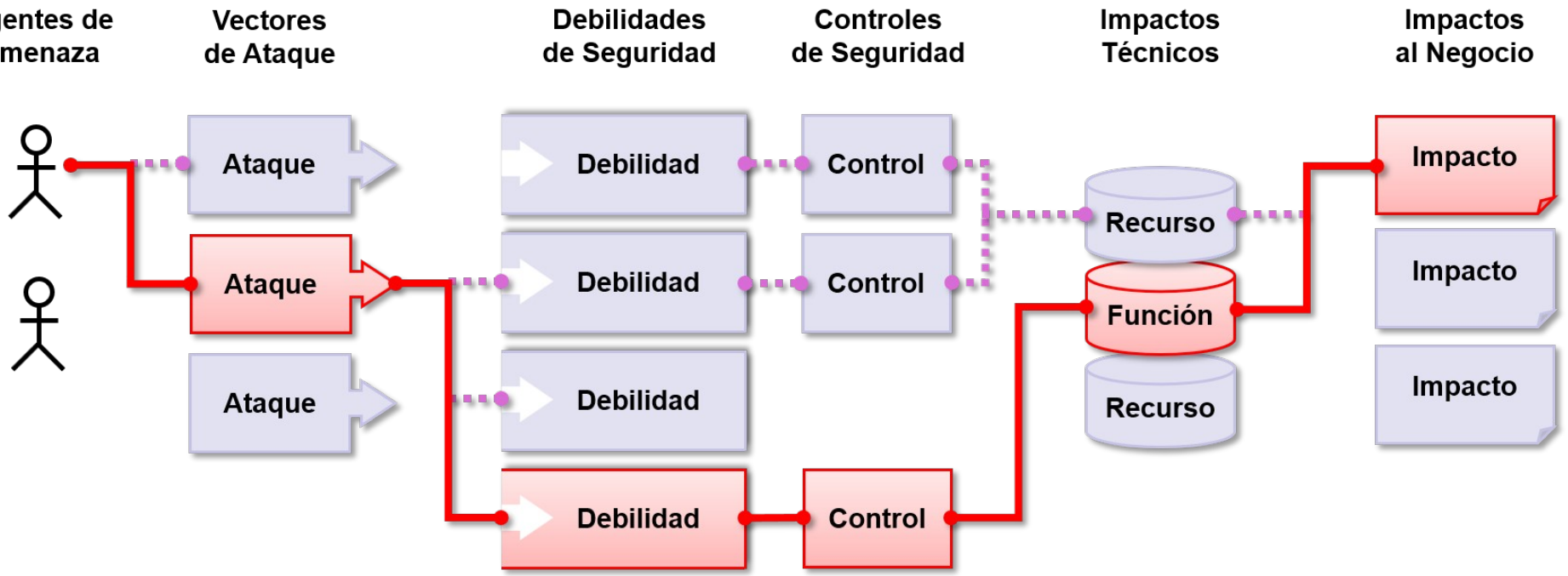
GRUPO DE SEGURIDAD INFORMÁTICA

Fundamentos de la Seguridad Informática

Seguridad en aplicaciones: OWASP Top Ten

Modelo de riesgo usado por OWASP



GRUPO DE SEGURIDAD INFORMÁTICA

Top 1-5

A1:2017 **Inyección**	Las fallas de inyección, como SQL, NoSQL, OS o LDAP ocurren cuando se envían datos no confiables a un intérprete, como parte de un comando o consulta. Los datos dañinos del atacante pueden engañar al intérprete para que ejecute comandos involuntarios o acceda a los datos sin la debida autorización.
A2:2017 **Pérdida de Autenticación**	Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son implementadas incorrectamente, permitiendo a los atacantes comprometer usuarios y contraseñas, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios (temporal o permanentemente).
A3:201 **Exposición de datos sensibles**	Muchas aplicaciones web y APIs no protegen adecuadamente datos sensibles, tales como información financiera, de salud o Información Personalmente Identificable (PII). Los atacantes pueden robar o modificar estos datos protegidos inadecuadamente para llevar a cabo fraudes con tarjetas de crédito, robos de identidad u otros delitos. Los datos sensibles requieren métodos de protección adicionales, como el cifrado en almacenamiento y tránsito.
A4:2017 **Entidades Externas XML (XXE)**	Muchos procesadores XML antiguos o mal configurados evalúan referencias a entidades externas en documentos XML. Las entidades externas pueden utilizarse para revelar archivos internos mediante la URI o archivos internos en servidores no actualizados, escanear puertos de la LAN, ejecutar código de forma remota y realizar ataques de denegación de servicio (DoS).
A5:2017 **Pérdida de Control de Acceso**	Las restricciones sobre lo que los usuarios autenticados pueden hacer no se aplican correctamente. Los atacantes pueden explotar estos defectos para acceder, de forma no autorizada, a funcionalidades y/o datos, cuentas de otros usuarios, ver archivos sensibles, modificar datos, cambiar derechos de acceso y permisos, etc.

GRUPO DE SEGURIDAD INFORMÁTICA

Top 6-10

A6:2017

Configuración de Seguridad Incorrecta

La configuración de seguridad incorrecta es un problema muy común y se debe en parte a establecer la configuración de forma manual, *ad hoc* o por omisión *(*o directamente por la falta de configuración). Son ejemplos: *S3 buckets* abiertos, cabeceras HTTP mal configuradas, mensajes de error con contenido sensible, falta de parches y actualizaciones, *frameworks,* dependencias y componentes desactualizados, etc.

A7:2017

Secuencia de Comandos en Sitios Cruzados (XSS)

Los XSS ocurren cuando una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada; o actualiza una página web existente con datos suministrados por el usuario utilizando una API que ejecuta *JavaScript* en el navegador. Permiten ejecutar comandos en el navegador de la víctima y el atacante puede secuestrar una sesión, modificar (*defacement*) los sitios web, o redireccionar al usuario hacia un sitio malicioso.

A8:2017

Deserialización Insegura

Estos defectos ocurren cuando una aplicación recibe objetos serializados dañinos y estos objetos pueden ser manipulados o borrados por el atacante para realizar ataques de repetición, inyecciones o elevar sus privilegios de ejecución. En el peor de los casos, la deserialización insegura puede conducir a la ejecución remota de código en el servidor.

A9:2017

Componentes con vulnerabilidades conocidas

Los componentes como bibliotecas, *frameworks* y otros módulos se ejecutan con los mismos privilegios que la aplicación. Si se explota un componente vulnerable, el ataque puede provocar una pérdida de datos o tomar el control del servidor. Las aplicaciones y API que utilizan componentes con vulnerabilidades conocidas pueden debilitar las defensas de las aplicaciones y permitir diversos ataques e impactos.

A10:2017

Registro y Monitoreo Insuficientes

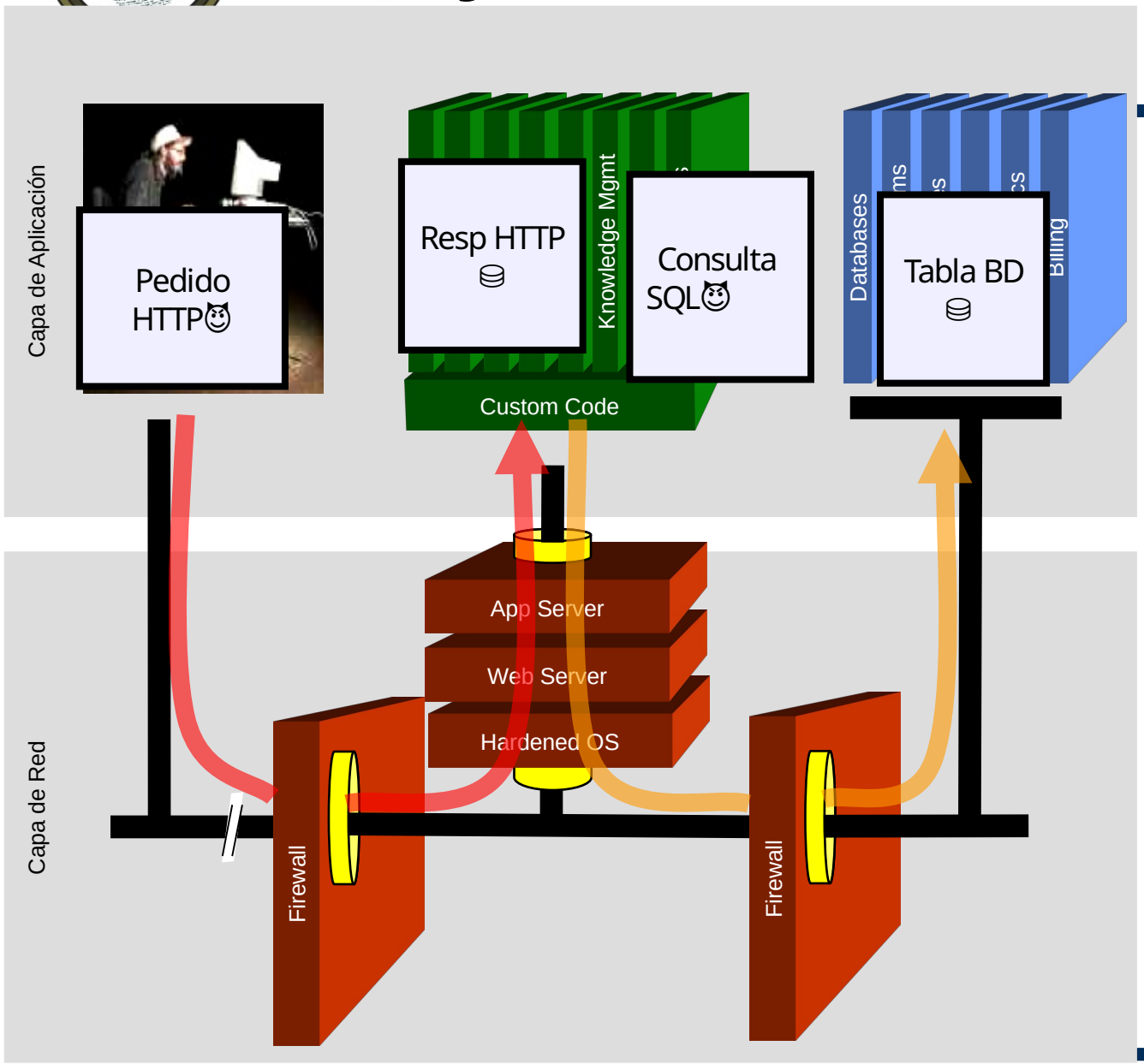
El registro y monitoreo insuficiente, junto a la falta de respuesta ante incidentes permiten a los atacantes mantener el ataque en el tiempo, pivotear a otros sistemas y manipular, extraer o destruir datos. Los estudios muestran que el tiempo de detección de una brecha de seguridad es mayor a 200 días, siendo típicamente detectado por terceros en lugar de por procesos internos

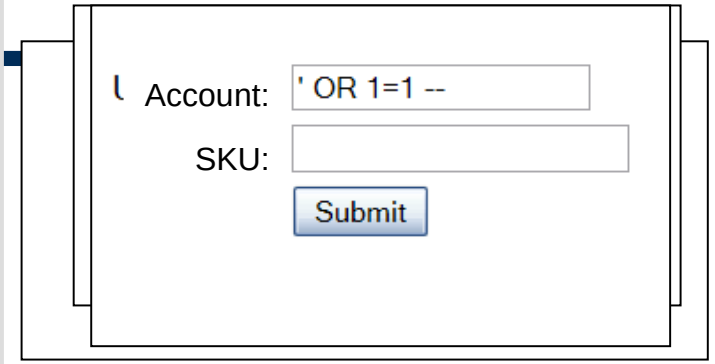
Grupo de Seguridad Informática

A1: Inyección

-
-
-
-

Inyección SQL – Demostración





1. Aplicación presenta un formulario web al atacante
2. Atacante envía un ataque en los datos del formulario
3. Aplicación dirige el ataque a la base de datos en una consulta SQL
4. Base de datos ejecuta el ataque y envía los resultados cifrados nuevamente a la aplicación
5. Aplicación descifra los datos normalmente y envía los resultados al atacante

A1: Evitar Fallas de Inyección

- Evitar el intérprete completamente
- Utilizar una interfaz que soporte variables parametrizadas (Ej., declaraciones preparadas, o procedimientos almacenados),
- Usar variables parametrizadas
- Decodificar y convertir todas las entradas del usuario a su forma mas simple antes de enviarlas al interprete
- Siempre efectuar una validación 'positiva' de todas las entradas realizadas por el usuario
- Seguir el principio de mínimo privilegio en las conexiones con bases de datos para reducir el impacto de una falla
  - –

A2: Pérdida de Autenticación y Gestión de Sesiones

- Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente
- Permiten a los atacantes comprometer contraseñas, llaves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios

Grupo de Seguridad Informática

A2: Soy Vulnerable?

- Permite ataques automatizados como la reutilización de credenciales conocidas, cuando el atacante posee una lista de pares de usuario y contraseña válidos.
- Permite ataques de fuerza bruta u otros ataques automatizados.
- Permite contraseñas por defecto, débiles o bien conocidas, como "Password1", "Contraseña1" o "admin/admin"

GRUPO DE SEGURIDAD INFORMÁTICA

A2: Cómo prevenirlo?

- Implemente la autenticación multifactor (MFA)
  - evita ataques automatizados, de relleno de credenciales, fuerza bruta o reuso de credenciales robadas
- No incluya o implemente en su software credenciales por defecto, particularmente para administradores
- Implemente un control contra contraseñas débiles, tal como verificar que la contraseña no esté incluida en la lista del top 10000 de peores contraseñas.
- Definir y aplicar políticas de contraseñas
  - Por ej: pautas de la sección 5.1.1 para Secretos Memorizados de la guía NIST 800-63 B's

A3: Exposición de Datos Sensibles

- Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular parámetros para acceder a datos no autorizados
- Aplicaciones o APIs no siempre verifican que el usuario está autorizado a trabajar con el recurso objetivo

A3: Soy vulnerable? (I)

- Identificar las necesidades de protección de datos sensibles
  - Datos en tránsito y en almacenamiento
  - Contraseñas, números de tarjetas de crédito, registros médicos, información personal y datos sensibles del negocio requieren una protección adicional
  - Reglamentaciones, por ej, Reglamento General de Protección de Datos de la UE (GDPR), PCI Data Security Standard (PCI DSS).

A3: Soy vulnerable? (II)

- ¿Se transmite algún dato en texto claro? Esto se refiere a protocolos como HTTP, SMTP y FTP. No sólo externo, también todo el tráfico interno, por ejemplo, entre los balanceadores de carga, servidores web o sistemas backend
- ¿Se utilizan algoritmos criptográficos antiguos o débiles?
- ¿Se utilizan claves criptográficas predeterminadas, se generan o reutilizan claves criptográficas débiles, o falta una gestión o rotación adecuada de las claves?

A3: Cómo se previene?

- Clasificar los datos procesados, almacenados o transmitidos por el sistema
- Aplicar los controles para cada clasificación
- No almacene datos sensibles innecesariamente
- Asegúrese de que se utilizan únicamente algoritmos y protocolos estándares y fuertes

Grupo de Seguridad Informática

A4: Entidades Externas XML (XXE)

- Atacantes pueden explotar procesadores XML vulnerables si pueden cargar XMLs o incluir contenido hostil en un documento XML, explotando código vulnerable, dependencias o integraciones.
- De forma predeterminada, muchos procesadores XML no modernos permiten la especificación de una entidad externa, una URI que se referencia y evalúa durante el procesamiento XML.

A4: Soy vulnerable?

- La aplicación acepta XML directamente o carga XML
  - especialmente de fuentes no confiables
  - inserta datos no confiables en documentos XML
- SAML utiliza XML para afirmaciones de identidad, pudiendo ser vulnerable.

A4: Cómo se previene?

- Utilizar formatos menos complejos como JSON
- Actualice todos los procesadores y bibliotecas XML
- Deshabilitar entidades externas de XML y procesamiento DTD en todos los analizadores sintácticos XML
- Implementar validación de entrada positiva ("lista blanca"), filtrado, o sanitización para prevenir datos hostiles dentro de documentos ,cabeceras o nodos XML

A5: Pérdida en control de acceso

- El control de acceso aplica la política de modo que los usuarios no puedan actuar fuera de los permisos previstos
- La explotación del Control de Acceso es una habilidad clave de los atacantes
- El control de acceso es detectable utilizando medios manuales, o posiblemente a través de la automatización por la ausencia de controles de acceso en ciertos frameworks

A5: Soy vulnerable?

- Pasar por alto las comprobaciones de control de acceso modificando la URL, el estado interno de la aplicación o página HTML, o utilizando una herramienta personalizada de ataques a API.
- Permitir que la clave primaria se cambie a la de otro usuario, pudiendo ver o editar la cuenta de otra persona.
- Elevación de privilegios: actuar como un usuario sin iniciar sesión, o actuar como un administrador iniciando sesión como usuario.

A5: Soy vulnerable? (2)

- Manipulación de metadatos, como reproducir o manipular un token de control de acceso JWT (JSON Web Token), una cookie o un campo oculto para elevar los privilegios, o abusar de la invalidación de tokens JWT.
- La configuración incorrecta de CORS permite el acceso no autorizado a la API.
- Forzar la navegación a páginas autenticadas como un usuario no autenticado o a páginas privilegiadas como usuario estándar. Acceder a API con controles de acceso ausentes para verbos POST, PUT y DELETE.

A5: Cómo se previene?

- El control de acceso solo es efectivo si es aplicado del lado del servidor o en la API
- Con la excepción de los recursos públicos, denegar de forma predeterminada
- Implemente los mecanismos de control de acceso una vez y reutilizarlo en toda la aplicación, incluyendo minimizar el control de acceso HTTP (CORS).

A6: Configuración de Seguridad Incorrecta

- Los atacantes a menudo intentarán explotar defectos sin parchear o acceder a cuentas predeterminadas, páginas no utilizadas, archivos y directorios desprotegidos, etc. para obtener acceso o conocimiento no autorizado del sistema.

A6: Soy vulnerable? (I)

- Falta de hardening adecuado en el stack tecnológico, o permisos mal configurados en los servicios de la nube
- Característica innecesarias habilitadas
- Cuentas predeterminadas y sus contraseñas siguen activas
- El manejo de errores revela trazas de la aplicación u otros mensajes de error demasiado informativos a los usuarios

A6: Soy vulnerable? (II)

- Para sistemas actualizados, las nuevas funciones de seguridad se encuentran desactivadas o no se encuentran configuradas de forma segura
- El servidor no envía cabezales de seguridad a los clientes o no se encuentran configurados con valores seguros
- El software se encuentra desactualizado o posee vulnerabilidades (ver A9: 2017)

A6: Cómo se previene?

- Un proceso de fortalecimiento reproducible que agilite y facilite la implementación de otro entorno que esté asegurado de manera apropiada.
- Una plataforma minimalista sin funcionalidades innecesarias, componentes, documentación o ejemplos. Elimine o no instale frameworks y funcionalidades no utilizadas.
- Un proceso para revisar y actualizar las configuraciones apropiadas para todas las advertencias de seguridad

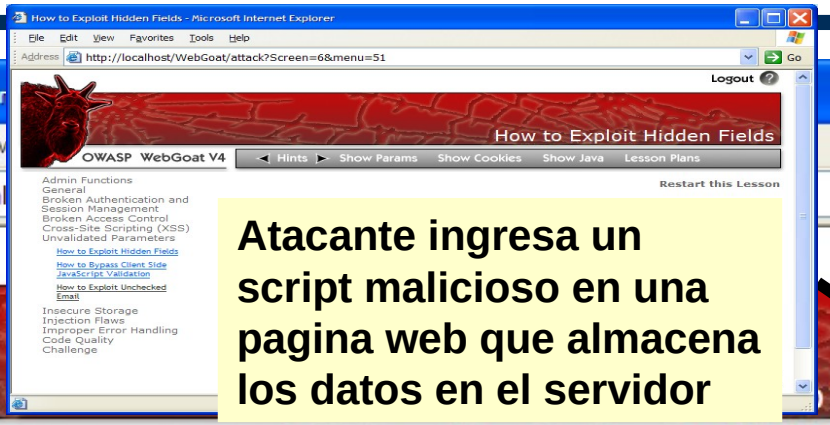
A7: XSS (Cross Site Scripting)

- Las fallas de XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada.
- XSS permite a los atacantes ejecutar secuencia de comandos cruzados (XSS) en el navegador de la victima los cuales pueden secuestrar las sesiones de usuario de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso

XSS - Demostración

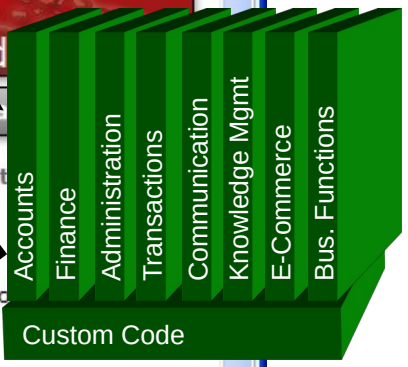
GRUPO DE SEGURIDAD INFORMÁTICA

1 **Atacante establece una trampa – actualizar perfil**



Atacante ingresa un script malicioso en una pagina web que almacena los datos en el servidor

Aplicación con vulnerabilidad XSS Almacenado



2 **Victima visualiza la pagina – accede al perfil**

Script se ejecuta en el navegador de la victima

2 **Script silenciosamente envía la sesión de la victima al atacante**

Grupo de Seguridad Informática

A7: Como evitar Fallas de XSS

- Eliminar la Falla: No incluir entradas suministradas por el usuario en la pagina de salida
- Defenderse de la Falla
- Siempre efectuar una validación 'positiva' de todas las entradas realizadas por el usuario
- XSS Prevention CheatSheet: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

A8: Deserialización Insegura

- La explotación de la deserialización es algo difícil, ya que los exploits como son distribuidos raramente funcionan sin cambios o ajustes en el código de exploit subyacente.

A8: Soy vulnerable?

- Aplicaciones y APIs serán vulnerables si deserializan objetos hostiles o manipulados por un atacante
- Dos tipos primarios de ataques:
  - Ataques relacionados con la estructura de datos y objetos donde el atacante modifica la lógica de la aplicación o logra una ejecución remota de código si existen clases disponibles para la aplicación que pueden cambiar el comportamiento durante o después de la deserialización.

A8 – Deserialización Insegura (2)

- Ataques típicos de manipulación de datos, como ataques relacionados con el control de acceso en los que se utilizan estructuras de datos existentes pero se modifica su contenido.

A8: Cómo prevenirlo?

- No aceptar objetos serializados de fuentes no confiables o utilizar medios de serialización que sólo permitan tipos de datos primitivos
- Si no es posible, considere uno o mas de los siguientes:
  - Implementar verificaciones de integridad (por ej. firmas digitales) con el fin de detectar modificaciones no autorizadas
  - Cumplimiento estricto de verificaciones del tipo de los datos
  - Aislar el código que realiza la deserialización, de modo que ejecute en un entorno con los mínimos privilegios
  - Registrar excepciones y fallas en la deserialización, tales como cuando el tipo recibido no es el tipo esperado, o la deserialización lanza excepciones.

A9: Uso de Componentes con Vulnerabilidades Conocidas

- En teoría, debiera ser fácil distinguir si estas usando un componente o biblioteca vulnerable
- Más aún, no todas las bibliotecas usan un sistema numérico de versiones entendible.
  - No todas las vulnerabilidades son reportadas a un centro de intercambio fácil de buscar, Sitios como CVE y NVD se están volviendo fáciles de buscar.

A9: Soy Vulnerable?

- Para determinar si es vulnerable necesita buscar en estas bases de datos, así como también mantenerse al tanto de la lista de correos del proyecto
- Debe evaluar cuidadosamente si es o no vulnerable revisando si su código utiliza la parte del componente vulnerable y si el fallo puede resultar en un impacto

A9: Cómo prevenirlo?

- Identificar todos los componentes y la versión que están ocupando, incluyendo dependencias
- Revisar la seguridad del componente en bases de datos públicas, lista de correos del proyecto, y lista de correo de seguridad, y mantenerlos actualizados.
- Establecer políticas de seguridad que regulen el uso de componentes

A10: Registro y Monitoreo Insuficientes

- Eventos auditables, tales como los inicios de sesión, fallos en el inicio de sesión, y transacciones de alto valor no son registrados
- Advertencias y errores generan registros poco claros, inadecuados o ninguno en absoluto
- Registros en aplicaciones o APIs no son monitoreados por actividad sospechosa
- Registros son almacenados únicamente de forma local

A10: Cómo prevenirlo?

- Errores de inicio de sesión, de control de acceso y de validación de entradas se deben registrar con el contexto de usuario suficiente para identificar cuentas sospechosas o maliciosas
- Transacciones de alto impacto deben de tener una pista de auditoría con controles de integridad para prevenir alteraciones o eliminaciones
- Establecer monitoreo y alerta de actividades sospechosas

Riesgos adicionales a considerar

- El Top 10 cubre mucho, pero hay otros riesgos que deberían considerarse
- Algunos de estos han aparecido en versiones previas del OWASP Top 10, y otros no, incluyendo nuevas técnicas de ataque que están siendo identificadas todo el tiempo

GRUPO DE SEGURIDAD INFORMÁTICA

Resumen

Riesgo	Agentes de Amenaza	Vectores de Ataque	Debilidades de Seguridad		Impacto		Puntuación
		Explotabilidad	Prevalencia	Detectabilidad	Técnico	Negocio	
A1: 2017- Inyección	Específico de la Aplicación	FACIL: 3	COMUN: 2	FACIL: 3	GRAVE: 3	Específico de la Aplicación	8,0
A2: 2017 - Pérdida de Autenticación	Específico de la Aplicación	FACIL: 3	COMUN: 2	PROMEDIO: 2	GRAVE: 3	Específico de la Aplicación	7,0
A3: 2017 - Exposición de Datos Sensibles	Específico de la Aplicación	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	GRAVE: 3	Específico de la Aplicación	7,0
A4: 2017 - Entidad Externa de XML (XXE)	Específico de la Aplicación	PROMEDIO: 2	COMUN: 2	FACIL: 3	GRAVE: 3	Específico de la Aplicación	7,0
A5: 2017 - Pérdida de Control de Acceso	Específico de la Aplicación	PROMEDIO: 2	COMUN: 2	PROMEDIO: 2	GRAVE: 3	Específico de la Aplicación	6,0
A6: 2017 - Configuración de Seguridad Incorrecta	Específico de la Aplicación	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2	Específico de la Aplicación	6,0
A7: 2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	Específico de la Aplicación	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2	Específico de la Aplicación	6,0
A8: 2017 - Deserialización Insegura	Específico de la Aplicación	DIFICIL: 1	COMUN: 2	PROMEDIO: 2	GRAVE: 3	Específico de la Aplicación	5,0
A9: 2017 - Componentes con Vulnerabilidades Conocidas	Específico de la Aplicación	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	MODERADO: 2	Específico de la Aplicación	4,7
A10: 2017 - Registro y Monitoreo Insuficientes	Específico de la Aplicación	PROMEDIO: 2	DIFUNDIDO: 3	DIFICIL: 1	MODERADO: 2	Específico de la Aplicación	4,0

Referencias

G. McGraw, *Software Security*, Addison-Wesley, 2006

OWASP Top Ten 2017