SEG 118 – Desarrollo Seguro basado en OWASP

Ing. Carlos Carreño

Nov, 2021

MODULO 4: SEGURIDAD EN EL DISEÑO DE SOFTWARE

- Criterios Básicos de Seguridad.
- Manejo seguro de errores.
- Manejo de información sensible
- Auditoría y Logging
- Diseño de Autenticación y Autorización
- Diseño de protección contra Denial of Service (D.O.S)
- Errores de Logica de negocio

Criterios Básicos de Seguridad

- Integridad
- Confidencialidad
- Disponibilidad
- No Repudio
- Gestión de identidades (Autenticación y autorización)

Integridad

- RNFS 1: utilizar marcos de trabajo que previenen automáticamente los ataques XSS (Cross-Site Scripting o inyección de código malicioso).
- RNFS 2: validar los datos que se reciben y velar por la integridad de los datos que se devuelven.
- RNFS 3: prevenir los ataques CSRF (del inglés Cross-Site Request Forgery o falsificación de petición en sitios cruzados).
- RNFS 4: evitar las inyecciones de código.
- RNFS 5: utilizar LIMIT y otros controles SQL para evitar la fuga masiva de datos en caso de inyecciones SQL.
- RNFS 6: validar la entrada de datos al servidor utilizando “listas blancas”.
- RNFS 7: cifrar los datos sensibles que sean almacenados.

Confidencialidad

- RNFS 8: proteger las conexiones autenticadas o que involucren funciones o información relevante.
- RNFS 9: evitar mostrar referencias hacia objetos internos de la aplicación.
- RNFS 10: evitar mostrar mensajes con información que ayude a recopilar información sobre el producto o las configuraciones del servidor.
- RNFS 11: evitar la elevación de privilegios en las cuentas de usuarios.
- RNFS 12: revisar todos los elementos de la infraestructura para asegurar que no contengan ninguna vulnerabilidad conocida, así como las herramientas administrativas usadas para el mantenimiento de los diferentes componentes.
- RNFS 13: evitar almacenar datos sensibles de manera innecesaria.
- RNFS 14: deshabilitar el almacenamiento en caché de datos sensibles.

Disponibilidad

- RNFS 15: realizar estudio sobre las posibles vulnerabilidades que se puedan presentar en la tecnología a utilizar en el desarrollo.
- RNFS 16: utilizar tecnologías seguras para el desarrollo.
- RNFS 17: cumplir los requisitos exclusivos de los límites de negocio de las aplicaciones.
- RNFS 18: controlar el receptor de escucha de las Bases de Datos.
- RNFS 19: garantizar que el servidor no envíe directrices o cabeceras de seguridad a los clientes o que se encuentren configurados con valores inseguros.
- RNFS 20: actualizar las configuraciones apropiadas de la tecnología usada de acuerdo a las advertencias de seguridad y seguir un proceso de gestión de parches.
- RNFS 21: utilizar una herramienta para mantener un inventario y control de versiones de los componentes
- RNFS 22: utilizar componentes únicamente de orígenes oficiales y utilizando los canales seguros.
- RNFS 23: analizar riesgos y vulnerabilidades del entorno de despliegue del cliente atendiendo a sus características.

No Repudio

- RNFS 24: cifrar todos los datos en tránsito utilizando protocolos seguros.
- RNFS 25: identificar o firmar de forma única los mensajes intercambiados.
- RNFS 26: almacenar los mensajes intercambiados en ficheros logs para su posterior consulta.

Autenticación y Autorización

- RNFS 27: evitar mantener credenciales creadas por defecto, débiles o muy conocidas especialmente en el caso de los administradores del sistema.
- RNFS 28: definir mecanismos de autenticación personalizado para todos los usuarios del sistema.
- RNFS 29: evitar utilizar cuentas suministradas por defecto.
- RNFS 30: evitar ataques de fuerza bruta y/o ataques automatizados.
- RNFS 31: utilizar controles contra contraseñas débiles.
- RNFS 32: alinear la política de longitud, complejidad y rotación de las contraseñas establecidas.
- RNFS 33: limitar el tiempo de respuesta de cada intento fallido de inicio de sesión.
- RNFS 34: controlar el ciclo de vida de las contraseñas.
- RNFS 35: restringir el acceso de un usuario estándar (no administrador) a modificar sus privilegios en la aplicación o los de otro usuario con su mismo rol.
- RNFS 36: cerrar automáticamente la sesión de un usuario cuando ha estado inactivo durante un cierto lapso de tiempo.
- RNFS 37: destruir el ID de sesión luego de salir o cerrar el sistema.

OWASP y Los Requisitos de Seguridad

- OWASP Top ten vulnerabilidades y los requisitos de seguridad

Riesgos/Vulnerabilidades	Requisitos que se relacionan
Inyección	RNFS 1, RNFS 2, RNFS 3, RNFS 4, RNFS 5, RNFS 6
Pérdida de autenticación	RNFS 27, RNFS 28, RNFS 29, RNFS 30, RNFS 31, RNFS 32, RNFS 33, RNFS 34, RNFS 35, RNFS 36, RNFS 37
Exposición de datos sensibles	RNF 7, RNFS 13, RNFS 14
Entidades externas XML	RNFS 6
Pérdida de control de acceso	RNFS 11, RNFS 17, RNFS 35
Configuración de seguridad incorrecta	RNFS 15, RNFS 16, RNFS 19, RNFS 20, RNFS 22
Ataques de XSS (Cross-Site Scripting)	RNFS 1, RNFS 2,
Uso de componentes con vulnerabilidades conocidas	RNFS 19, RNFS 20, RNFS 21, RNFS 22, RNFS 23

Reference: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992018000500015&lng=es&tlng=pt

Practica

- Realizar las actividades usando Webgoat sobre A1:2017 Inyección.

Manejo seguro de errores

- Existen dos maneras de crear un error de seguridad.
  - Errores no controlados correctamente y
  - crear errores que brinden información.
- El correcto control de errores desde el desarrollo del software es de vital importancia ya que estos pueden brindar información útil o ser aprovechados como pivotes para un atacante.

Diseño de Autenticación y Autorización

- Pueden existir debilidades de autenticación si la aplicación:
  - ❑Permite ataques automatizados como la reutilización de credenciales conocidas, cuando el atacante ya posee una lista de pares de usuario y contraseña válidos.
  - ❑Permite ataques de fuerza bruta y/o ataques automatizados.
  - ❑Permite contraseñas por defecto, débiles o muy conocidas, como “Password1”, “Contraseña1”o “admin/admin”.
  - ❑Posee procesos débiles o inefectivos en el proceso de recuperación de credenciales, como “respuestas basadas en el conocimiento”, las cuales no se pueden implementar de forma segura.
  - ❑Almacena las contraseñas en texto claro o cifradas con métodos de hashingdébiles (vea A3:2017-Exposición de Datos Sensibles).
  - ❑No posee autenticación multi-factor o fue implementada de forma ineficaz.
  - ❑Expone Session IDs en las URL, no la invalida correctamente o no la rota satisfactoriamente luego del cierre de sesión o de un periodo de tiempo determinado.

Autenticación: Medidas Preventivas

- Implemente autenticación multi-factor para evitar ataques automatizados, de fuerza bruta o reúso de credenciales robadas.
- No utilice credenciales por defecto en su software, particularmente en el caso de administradores.
- Implemente controles contra contraseñas débiles. Cuando el usuario ingrese una nueva clave, la misma puede verificarse contra la lista del Top 10.000 de peores contraseñas.
- Alinear la política de longitud, complejidad y rotación de contraseñas con las recomendaciones de la Sección 5.1.1 para Secretos Memorizados de la Guía NIST 800-63 B'su otras políticas de contraseñas modernas, basadas en evidencias.

… continua

- Mediante la utilización de los mensajes genéricos iguales en todas las salidas, asegúrese que el registro, la recuperación de credenciales y el uso de APIs, no permiten ataques de enumeración de usuarios.
- Limite o incremente el tiempo de respuesta de cada intento fallido de inicio de sesión. Registre todos los fallos y avise a los administradores cuando se detecten ataques de fuerza bruta.
- Utilice un gestor de sesión en el servidor, integrado, seguro y que genere un nuevo ID de sesión aleatorio con alta entropía después del inicio de sesión. El Session-IDno debe incluirse en la URL, debe almacenarse de forma segura y ser invalidado después del cierre de sesión o de un tiempo de inactividad determinado por la criticidad del negocio.

Practica

- Realizar las actividades usando Webgoat sobre A2:2017 Perdida de Autenticación.

Manejo de información sensible

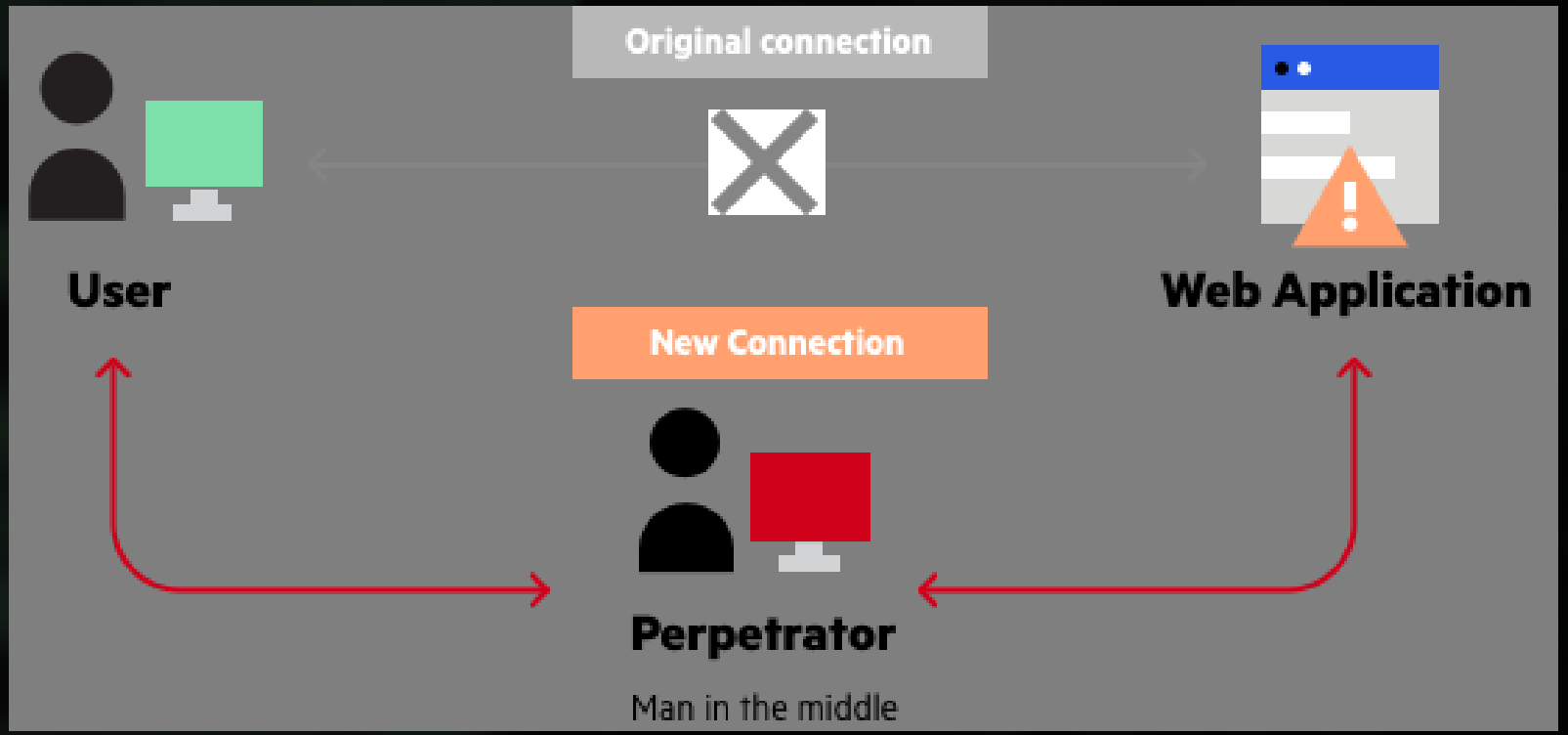
- Que son los datos sensibles?
- Principios básicos de la protección de datos sensibles:
  - ✓Licitud
  - ✓Calidad de datos
  - ✓Finalidad
  - ✓Consentimiento
  - ✓Información
  - ✓Responsabilidad
  - ✓Proporcionalidad
  - ✓Lealtad

Tipos de datos sensibles

- Datos sensibles en transito
- Datos sensibles en reposo

Ataque de datos sensibles en transito

- **Sniffing**: Cuando los datos no están encriptados.



...

- Divulgación de información

Ataque de datos sensibles en reposo

- Ataque de autenticación a servidores de base de datos o repositorios

Impacto de ataque a información sensible

- Financiero
- Legal
- Imagen corporativa
- Confianza del cliente

Vulnerabilidades que exponen datos sensibles

- Transmisión de datos en texto claro
- Algoritmos criptográficos obsoletos

Seguridad de datos sensibles

- La seguridad de los datos sensibles se realiza:
  - A través del cifrado y seguimiento de amenazas o actividades inusuales, así como las respuestas que a ellas se les da

Practica

- Realizar las actividades usando Webgoat sobre A3:2017 Exposición de datos sensibles.

Auditoría y Logging

- La ausencia de Auditoria y Logging (registro de incidentes) es la madre de la mayor parte de incidentes de seguridad.
- La falta de auditoria y logging permite al atacante lograr su objetivo sin ser detectado.

Auditoria y logging: Vulnerabilidades

- 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 para detectar actividades sospechosas.
- Los registros son almacenados únicamente de forma local.
- Los umbrales de alerta y de escalamiento de respuesta no están implementados o no son eficaces.
- Las pruebas de penetración y escaneos utilizando herramientas DAST(como OWASP ZAP) no generan alertas.
- La aplicación no logra detectar, escalar o alertar sobre ataques en tiempo real.

Auditoria y logging: Medidas Preventivas

- Asegúrese de que todos los errores de inicio de sesión, de control de acceso y de validación de entradas de datos del lado del servidor se pueden registrar para identificar cuentas sospechosas. Mantenerlo durante el tiempo suficiente para permitir un eventual análisis forense.
- Asegúrese de que las transacciones de alto impacto tengan una pista de auditoría con controles de integridad para prevenir alteraciones o eliminaciones.
- Asegúrese que todas las transacciones de alto valor poseen una traza de auditoría con controles de integridad que permitan detectar su modificación o borrado, tales como una base de datos con permisos de inserción únicamente u similar.
- Establezca una monitorización y alerta efectivos de tal manera que las actividades sospechosas sean detectadas y respondidas dentro de períodos de tiempo aceptables.
- Establezca o adopte un plan de respuesta o recuperación de incidentes, tales como NIST 800-61 rev.2 o posterior.

Diseño de protección contra Denial of Service (D.O.S)

- Un ataque de denegación de servicio (Dos) es un intento malintencionado de afectar la disponibilidad del sistema atacado, como por ejemplo, un sitio web una aplicación, para legitimar a usuarios finales.
- Los atacantes suelen generar grandes volúmenes de paquetes o requerimientos para, finalmente, sobrecargar el sistema objetivo.
- En el caso de un ataque de denegación de servicio distribuidos (DDoS) el atacante utiliza múltiples fuentes de vulnerabilidad o fuentes controladas para generar el ataque.

DDoS y Modelo OSI

- En general, los ataques DDoS pueden ser segregados según la capa del modelo de interconexión de sistemas abiertos (OSI) que atacan. Son más comunes en las siguientes capas:
  - red (capa 3)
  - transporte (capa 4)
  - presentación (capa 6) y
  - aplicación (capa 7)

Modelo OSI y Vectores de Ataque

#	Capa	Aplicación	Descripción	Ejemplo de vector
7	Aplicación	Datos	Procesamiento de red para la aplicación	Inundaciones HTTP, inundaciones de consultas DNS
6	Presentación	Datos	Representación de datos y cifrado	Abuso de SSL
5	*Sesión*	*Datos*	*Comunicación entre hosts*	*N/D*
4	Transporte	Segmentos	Conexiones integrales y confiabilidad	Inundaciones SYN
3	Red	Paquetes	Determinación de la ruta y direccionamiento lógico	Ataques de reflexión UDP
2	*Enlace de datos*	*Marcos*	*Direccionamiento físico*	*N/D*
1	*Físico*	*Bits*	*Medios, señal y transmisión binaria*	*N/D*

Clasificación de los ataques DDoS

- Ataques a la capa de infraestructura
  - Los ataques a las capas 3 y 4, en general, están clasificados como ataques a las capas de infraestructura. Además, son los ataques DDoS más comunes e incluyen vectores como inundaciones sincronizadas (SYN) y otros ataques como inundaciones de paquetes de datagramas de usuario (UDP).
- Ataques a la capa de aplicación
  - Los ataques a las capas 6 y 7 se clasifican como ataques a las capas de aplicación. Mientras que estos ataques son menos comunes, tienden a ser más sofisticados. Estos ataques son, en general, más pequeños en volumen en comparación con los ataques a las capas de infraestructura pero tienden a focalizarse en partes especificas y costosas de la aplicación e impiden que esté disponible a los usuarios reales. Por ejemplo, una inundación de requerimientos de HTTP a una página de inicio de sesión o a una búsqueda costosa de una API o incluso inundaciones Wordpress XML-RPC (también conocidas como ataques pingback Wordpress).

Técnicas de protección DDoS

- Reduzca la superficie expuesta a ataques
- Plan para escalado
  - ❑Capacidad de tránsito
  - ❑Capacidad del servidor
- Conozca qué es el tráfico normal y anormal
- Implemente firewalls para ataques sofisticados de aplicaciones

Errores de Lógica de negocio

- Diferencia entre lógica de negocio y lógica funcional

… continua

- La lógica funcional es la que todos manejamos desde siempre para definir el alcance y los flujos de información, interfaz de usuario, etc de un sistema de información.
- La lógica de negocio en cambio se refiere al flujo externo, transversal a la organización, entre distintos sistemas de información dentro y fuera del área funcional.
- Para una cierta entidad o proceso, *la lógica funcional define lo que ocurre **dentro** del sistema de información* como consecuencia de cualquier evento sobre esa entidad o proceso, mientras que la *lógica de negocio define lo que ocurre **fuera** del sistema de información* como consecuencia de aquellos eventos sobre dicha entidad o proceso, que tengan impacto en procesos de negocio (y por tanto, en otras áreas funcionales y sistemas de información).

Lógica de Negocio: Error, Defecto y Fallo

- Error
- Defecto
- Fallo

Lógica de Negocio: Error

- Es una acción humana que produce un resultado incorrecto, una idea equivocada de algo. El error es una equivocación de parte del desarrollador o del analista. Un error puede llevarnos a generar uno o más defectos.
- Ejemplos de errores pueden ser:
  - ❑Error en la lógica de la programación
  - ❑Un requerimiento que esté mal especificado

Lógica de Negocio: Defecto

- El defecto se encuentra en algún componente del sistema. Es la imperfección de un componente causado por un error. El analista de pruebas es quien debe reportar el defecto ya que es el encargado de ejecutar los casos de prueba y encontrar los mismos.
- Ejemplos de defecto pueden ser:
  - ❑Un módulo de registro de usuarios tiene mala configuración en la función de conexión a base de datos
  - ❑Una función de login cuenta con las variables de usuario y contraseña declaradas incorrectamente.

Lógica de Negocio: fallo

- Es la manifestación visible de un defecto. Es decir que si un defecto es encontrado durante la ejecución de una aplicación entonces va a producir un fallo.
- Ejemplo de Fallo:
  - ❑Visualización de un mensaje de alerta que no fue definido previamente por el desarrollador.
  - ❑Un formulario de login que contenga los datos de acceso no te permita ingresar a la aplicación al hacer clic en el botón de ingresar.

Lógica de Negocio: Medidas preventivas

- Eliminar todos los errores de la capa de negocio.
- Establecer contratos claros de las APIS de acceso a lógica de negocio.
- Realizar pruebas de escenarios de lógica de negocio.

Referencias

- https://www.youtube.com/watch?v=C_-ea63FUto
- https://blog.a3sec.com/desarrollo-seguro-principios-y-como-comenzar
- https://www.docusign.mx/blog/datos-sensibles
- https://www.youtube.com/watch?v=I_9Giwoy0Lg
- https://www.youtube.com/watch?v=s2riErLuM4s
- https://www.youtube.com/watch?v=SMaYQ3gbEK4
- https://aws.amazon.com/es/shield/ddos-attack-protection/
- https://www.modusoperantic.com/es/logica-de-negocio-vs-logica-funcional/