TEMA 126. LA SEGURIDAD EN EL NIVEL DE APLICACIÓN. TIPOS DE ATAQUES Y PROTECCIÓN DE SERVICIOS WEB, BASES DE DATOS E INTERFACES DE USUARIO.

Actualizado en 2022
en2020211/01/2022

1. La seguridad a nivel de aplicación

Es necesario contemplar la seguridad en el desarrollo de aplicaciones desde el comienzo de los proyectos.

2. seguridad en interfaces de usuario

Aspectos a tener en cuenta:

- **Autenticación:** proceso por el que el usuario acredite su identidad mediante unas credenciales
- **Autorización:** proceso por el que el sistema comprueba que un usuario tiene permiso para realizar la operación que ha solicitado.
- **Gestión de excepciones**: es conveniente no proporcionar al usuario información interna del sistema que un usuario malintencionado pueda aprovechar para atacar.
- **Validación de datos de entrada**: puede provocar que, a través de la introducción de datos inválidos, un atacante interactúe con un sistema de forma no deseada.
- **Trazabilidad y auditorías**: todas las capas de un sistema deben guardar información de excepciones o errores que permitan trazar su origen y corregir posibles defectos del software.

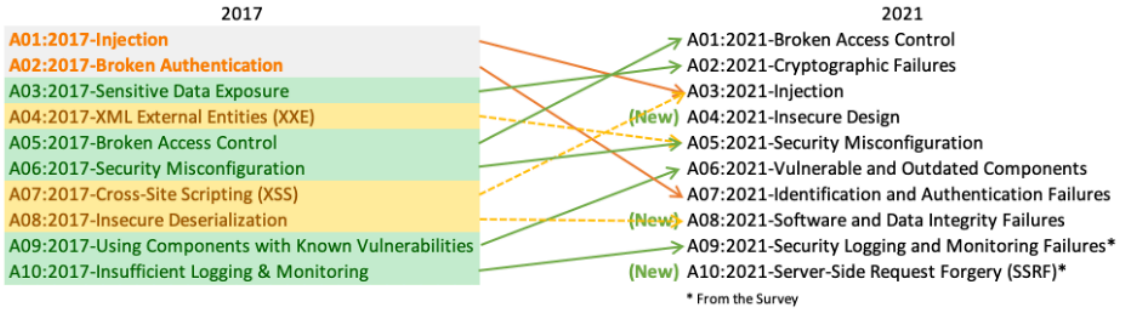
3. seguridad en aplicaciones web

3.1. owasp (Open Web Application Security Project)

Comunidad de seguridad informática que publica y actualiza cada 3 años el proyecto **OWASP Top 10** que es un documento donde se recogen los diez riesgos de seguridad más importantes en aplicaciones Web.

3.2. amenazas y ataques en aplicaciones web

Los diez riesgos más críticos en Aplicaciones Web según OWASP Top 10 - 2021 son:



- A01:2021-Broken Access Control: El control de acceso hace cumplir la política de manera que los usuarios no puedan actuar fuera de los permisos previstos. Los fallos suelen provocar la divulgación de información no autorizada, la modificación o destrucción de todos los datos o la realización de una función empresarial fuera de los límites del usuario.
- A02:2021-Cryptographic Failures Nombrado anteriormente como Exposición de datos sensibles, el punto clave es la criptografía utilizada o la falta de esta, tanto en datos almacenados como en tránsito.

- **A03:2021-Injection** Algunas de las inyecciones más comunes son SQL, NoSQL, comando OS, Object Relational Mapping (ORM), LDAP y Expression Language (EL) o Object Graph Navigation Library (OGNL). El concepto es idéntico entre todos los intérpretes. La revisión del código fuente es el mejor método para detectar si las aplicaciones son vulnerables a las inyecciones. Se recomienda encarecidamente realizar pruebas automatizadas de todos los parámetros, cabeceras, URL, cookies, JSON, SOAP y entradas de datos XML. Las organizaciones pueden incluir herramientas de pruebas de seguridad de aplicaciones estáticas (SAST), dinámicas (DAST) e interactivas (IAST) en la cadena de producción de CI/CD para identificar los fallos de inyección introducidos antes del despliegue en producción.
- **A04:2021-Insecure Design** El diseño inseguro es una categoría amplia, creada en 2021, que representa diferentes debilidades, expresadas como "diseño de control ausente o ineficaz". El diseño inseguro no es el origen de todas las demás categorías de riesgo del Top 10. Hay una diferencia entre diseño inseguro e implementación insegura. Diferenciamos entre defectos de diseño y defectos de implementación por una razón, tienen diferentes causas de origen y remediación. Un diseño seguro puede seguir teniendo defectos de implementación que den lugar a vulnerabilidades que puedan ser explotadas. Un diseño inseguro no puede arreglarse con una implementación perfecta, ya que, por definición, los controles de seguridad necesarios nunca se crearon para defenderse de ataques específicos. Uno de los factores que contribuyen a un diseño inseguro es la falta de un perfil de riesgo empresarial inherente al software o al sistema que se está desarrollando y, por tanto, la incapacidad de determinar qué nivel de diseño de seguridad se requiere.
- **A05:2021-Security Misconfiguration** Sin un proceso concertado y repetible de configuración de la seguridad de las aplicaciones, los sistemas corren un mayor riesgo. La antigua categoría de A4:2017-Entidades externas XML (XXE) forma parte ahora de esta categoría de riesgo
- **A06:2021-Vulnerable and Outdated Components** Los componentes vulnerables son un problema conocido que nos cuesta probar y evaluar el riesgo, y es la única categoría que no tiene ninguna Enumeración de Debilidades Comunes (CWE) asignada a los CWE incluidos, por lo que se utiliza una ponderación de exploits/impacto por defecto de 5,0. Los CWE más destacados que se incluyen son el CWE-1104: Uso de componentes de terceros sin mantenimiento y los dos CWE de los Top 10 de 2013 y 2017.
- **A07:2021-Identification and Authentication Failures** Conocida anteriormente como Broken Authentication, la confirmación de la identidad del usuario, la autenticación y la gestión de la sesión son fundamentales para protegerse de los ataques relacionados con la autenticación.
- **A08:2021-Software and Data Integrity Failures** Una nueva categoría para 2021 se centra en hacer suposiciones relacionadas con las actualizaciones de software, los datos críticos y los conductos de CI/CD sin verificar la integridad. Los fallos en la integridad del software y los datos están relacionados con el código y la infraestructura que no protegen contra las violaciones de la integridad. Un ejemplo de esto es cuando una aplicación depende de plugins, bibliotecas o módulos de fuentes, repositorios y redes de distribución de contenidos (CDN) que no son de confianza. Un canal de CI/CD inseguro puede introducir el potencial de acceso no autorizado, código malicioso o compromiso del sistema. A8:2017-Insecure Deserialization es ahora parte de esta categoría.
- **A09:2021-Security Logging and Monitoring Failures** Anteriormente conocida como A10:2017-Insufficient Logging & Monitoring, el registro, la detección, la supervisión y la respuesta activa pueden tener un gran impacto en la rendición de cuentas, la visibilidad, la alerta de incidentes y el análisis forense. El registro y la supervisión pueden ser difíciles de probar, a menudo implicando entrevistas o preguntando si se detectaron ataques durante una prueba de penetración, sin embargo son críticos.
- **A10:2021-Server-Side Request Forgery** Los fallos SRF se producen cuando una aplicación web obtiene un recurso remoto sin validar la URL proporcionada por el usuario. Permite a un atacante coaccionar a la aplicación para que envíe una solicitud manipulada a un destino inesperado, incluso cuando está protegida por un cortafuegos, una VPN u otro tipo de lista de control de acceso a la red (ACL). A medida que las aplicaciones web modernas ofrecen a los usuarios finales características convenientes, la búsqueda de una URL se convierte en un

escenario común. Como resultado, la incidencia de SSRF está aumentando. Además, la gravedad de la SSRF es cada vez mayor debido a los servicios en la nube y a la complejidad de las arquitecturas.

OTROS ATAQUES

- Desplazamientos por directorios (*path traversal*)
- *Cross-Site Tracing* (XST)
- Cross-Site Request Forgery (CSRF)
- Remote file inclusion (RFI) y Local File Inclusion (LFI)
- Ataques contra SSL/TLS

3.3. METODOLOGÍAS DE SEGURIDAD EN APLICACIONES WEB

Una metodología de seguridad en aplicaciones web debe tener en cuenta lo siguiente:

- La seguridad en el ciclo de vida de desarrollo.
- Realizar pruebas de caja negra.
- Realizar pruebas de caja blanca.
- Implementar firewalls de nivel de aplicación (Web Application Firewalls, WAF).

4. SEGURIDAD EN SERVICIOS WEB

4.1 WS-SECURITY

WS-Security (WSS): extensión al estándar SOAP para proporcionar mecanismos de seguridad a los mensajes SOAP. Proporciona autenticación, integridad y confidencialidad en el intercambio.

Especifica cómo usar tecnologías y estándares como aserciones SAML o un certificado X.509, XML Encryption y XML DSignature. Mediante la lectura de las extensiones WSS de cabecera del mensaje SOAP se identifica qué mecanismos ha escogido el emisor de este para autenticar al usuario final, cifrar el mensaje y firmarlo y de esta forma proceder a su validación.

Hoy en día WS-Security se está convirtiendo en el método principal para asegurar Servicios Web.

SEGURIDAD A NIVEL DE TRANSPORTE (TLS) Y A NIVEL DE MENSAJE (WSS)

Diferencias entre los dos mecanismos:

TLS	*WSS*
Extremo a extremo de nivel de transporte (IP+puerto)	Extremo a extremo de nivel de aplicación.
Credenciales de autenticación de navegador y servidor únicos sin intermediarios.	Credenciales de autenticación de usuario final. Permite múltiples credenciales de autenticación.
No es flexible: se aplica a todo el payload de igual manera durante toda la sesión	Muy flexible: puede aplicarse sólo a una parte del mensaje y discriminar entre Requests y Responses
Se sitúa entre el nivel TCP y el de aplicación	La misma seguridad se puede aplicar en diferentes tecnologías de transporte.

Tecnologías de autenticación, confidencialidad e integridad soportadas por WSS

Tecnologías soportadas:

- **Integridad**: WSS emplea **XML Signature** (W3C) para la firma de los mensajes SOAP. Un mensaje SOAP puede llevar ninguna, una o múltiples firmas XML Signature.
- **Confidencialidad**: se basa en la especificación **XML-Encryption** (W3C) que al igual que ocurre con XML Signature permite encriptar parte o la totalidad del mensaje.
- **Autenticación**: WS-Security utiliza extensiones XML en la cabecera denominadas *tokens de seguridad*. Ejemplos de tokens disponibles: **username/password**, certificados **X.509**, tickets **Kerberos**, aserciones **SAML**, tokens XrML, tokens XCBF o referencias a URIs. El uso de SAML permite también intercambiar aserciones de autorización para usuarios del servicio.

Otras extensiones para web services

WS-Security se usa únicamente para autenticación, confidencialidad (cifrado) e integridad (firma).
Existen otras extensiones SOAP **para el ámbito de la seguridad**:

- **WS-Policy**: permite a un servicio web especificar sus requisitos de calidad de servicio y de seguridad para que puedan ser consultadas antes de invocar la interfaz WS. Las extensiones para especificar la tecnología aceptada de seguridad se denominan **WS-SecurityPolicy** y permiten definir qué tokens de autenticación se admiten, qué partes deben estar encriptadas, qué partes firmadas etc.
- **WS-Addressing**: proporciona mecanismos neutrales, independientes del protocolo de transporte, para direccionar mensajes y servicios web. Convierte los mensajes SOAP en unidades autónomas de comunicación
- **WS-Trust**: define cómo intercambiar tokens de seguridad con credenciales de acceso a servicios web de distintos dominios de confianza.
- **WS-Federation**. Una Federación es un conjunto de dominios de seguridad que han acordado compartir recursos de forma segura. Está soportado por las extensiones ofrecidas por WS-Security (seguridad en el mensaje SOAP), WS-Trust (intercambio de tokens de seguridad entre distintos dominios de seguridad), and WS-SecurityPolicy (publicidad de la política de seguridad).
- **WS-SecureConversation**: Extensión a WS-Security para el establecimiento de contextos de seguridad para el intercambio de una serie de mensajes SOAP durante toda la sesión que se establezca entre partes de un Web-Service (lo que se denomina una “conversación”). Se diferencia de WS-Security en que ésta norma protege cada uno de los mensajes SOAP de forma independiente, lo que introduce un elevado overhead, mientras que WS-SecureConversation cubre toda la conversación.

Crítica a WSS y alternativas

Principal crítica realizada a WS sobre SOAP y sus extensiones WSS: el overhead que introducen y su lentitud.

Alternativa: los servicios RESTFul que son más ligeros y su despliegue suele ser más rápido. Ofrecen interfaces directas HTTP, los mensajes van directamente encapsulados en el payload de este protocolo

No ofrecen soluciones globales interoperables, como pretende WSS y sus extensiones, pero ofrecen una solución más ligera y rápida. Su seguridad suele estar garantizada mediante TLS.

4.2 SAML

SAML (*Security Assertion Markup Language*): lenguaje de marcado basado en XML desarrollado por el consorcio OASIS para el intercambio de información de autenticación y autorización entre proveedores de servicios de identidad, proveedores de servicios y usuarios de los servicios. Permite la extensión de las operaciones de autenticación y autorización de un dominio de seguridad a otro distinto con el que el primero mantiene una relación de confianza. La última versión es la 2.0 y data de 2005.

El uso más extendido de SAML es como mecanismo de Single Sign On y de federación de identidades, permite también intercambiar atributos de los usuarios de los servicios con el fin de establecer perfiles para procesos de autorización.

Características generales

- Múltiples tecnologías de transporte (http, SOAP, etc)
- Distribuido no requiere una autoridad central.

Mecanismos de funcionamiento

- **Aserciones SAML.** Define esquemas XML para expresar lo que denomina "aserciones" (assertions), que son verificaciones constatadas sobre un usuario. Existen 3 tipos:
  - **Autenticaciones**: verificaciones de identidad.
  - **Autorizaciones**: las autoridades de autorización (normalmente las mismas que las de identificación) confirman o deniegan permisos de usuarios concretos ya identificados para realizar determinadas acciones.
  - **Atributos**. Expresan datos sobre una entidad (normalmente personas) que permiten a los proveedores de servicio aplicar su propia política de control de acceso o de servicio en función de la respuesta.

- **Roles**. SAML considera 3 roles:
  - **Principal**: se denomina así a la entidad que requiere ser autenticada; puede ser un usuario final u otro sistema.
  - **Proveedor de servicio (SP)**: desea autenticar, autorizar o conocer algún dato de algún usuario o sistema (principal)
  - **Proveedor de identidad (IDP)**: autentica, autoriza o envía datos de usuarios (principales).

- **Protocolos**.

Protocolo	Función
Authentication Request Protocol	Solicitud de autenticación de un usuario
Single Logout Protocol	Define la forma de realizar un logout simultáneo de las sesiones asociadas a un usuario. El logout puede ser iniciado directamente por el usuario, o iniciado por un IDP o SP

	debido a la expiración por tiempo o también por decisión de un administrador
Assertion Query and Request Protocol	Solicitud de aserciones. La forma de Request de este protocolo permite solicitar a un IDP un Assertion haciendo referencia a su ID. La forma de Query define como solicitar Assertions en base al usuario y el tipo de Assertion
Name Identifier Management Protocol	Provee mecanismos para cambiar el valor o formato del identificador usado para referirse al usuario. El protocolo también define la forma de terminar una asociación de la identidad del usuario entre el IDP y el SP
Name Identifier Mapping Protocol	Provee un mecanismo para poder mapear un identificador de usuario usado entre un IDP y un SP de forma de obtener el utilizado entre el IDP y otro SP.
Artifact Resolution Protocol	Permite transportar los Assertions por referencia (a esta referencia se le denomina Artifact en el estándar) en vez de valor. Esto facilita el uso de SAML en un entorno web.

- **Bindings:** especificación que indica cómo encapsular los mensajes Request/Response del protocolo SAML en el payload del mensaje de un mecanismo de transporte concreto. SAML define numerosos bindings. Algunos de los más importantes son:

    - Binding de SAML sobre SOAP (Binding SOAP): indica cómo insertar un mensaje SAML en el cuerpo (*body*) de un mensaje SOAP con indicaciones de cómo transportar a su vez el mensaje SOAP sobreHTTP.
    - Binding de SAML sobre el método POST de HTTP. (Binding POST).
    - Binding de SAML sobre mensajes HTTP de redirección (Binding Redirección).

- **Perfiles SAML:** definen cómo combinar aserciones SAML, protocolos y bindings para poder cubrir determinados escenarios de uso y garantizar la interoperabilidad. Cabe destacar:
    - Perfil SSO para browsers (Web Browser SSO Profile). Pensado para ofrecer servicios de SSO a clientes con navegadores estándar.
    - Perfil mejorado de cliente y proxy (Enhanced Client and Poxy - ECP). Caso especial en que se emplea un mediador de proveedores de servicios de identidad (caso de Cl@ve).

SAML y Web Service-Security

Mientras que los mensajes de SAML sobre SOAP van en el cuerpo (*body*) de SOAP y forman parte de un protocolo Request/Response de autenticación, autorización o consulta de atributos; los mensajes SAML de la norma WSS son sólo aserciones que van en la cabecera (*header*) SOAP y tienen como objetivo

participar de la seguridad del mensaje SOAP en sí. Normalmente son aserciones de autenticación que van firmadas mediante XML DSignature y que permiten establecer la identidad del emisor del mensaje SOAP.

5. SEGURIDAD DE LAS BASES DE DATOS

Vulnerabilidades más habituales en BBDD:

- Debilidad de credenciales.
- Funciones no utilizadas.
- Inyección SQL.

Buenas prácticas:

- Mínimos privilegios.
- Elección y mantenimiento adecuados del SGBD.
- Gestión de usuarios mediante grupos.