UNIDAD 4 – APLICACIONES DE SEGURIDAD

Logging

Tiene como finalidad:

- Identificar incidentes de Seguridad
- Monitorear violaciones a las politicas
- Asistir en controles de no-repudio
- Proveer información sobre problemas o situaciones atípicas
- Contribuir con información específica para la investigación de incidentes que no pueda obtenerse de otras fuentes.
- Contribuir con la defensa ante vulnerabilidades y exploits mediante la detección de ataques

Registro

- Sistemas de archivos
- Almacenamiento en la nube
- Bases de datos SQL y NoSQL

Elementos a registrar

- Fallos de validación
- Autenticación
- Autorización
- Anomalías
- Fallas de aplicación
- Eventos legales

Elementos que no deben ser registrados

- Código fuente
- Identificadores de sesión
- Credenciales y tokens de acceso
- Claves de cifrado
- SPI

Enmascaramiento de datos

Proceso por el cual se reemplazan o sustituyen los datos sensibles o identificables de un sistema con el objetivo de proteger la privacidad de la información. Su objetivo principal es el de preservar la utilidad e integridad de los datos.

Debilidades Asociadas al Logging

- CWE-117 Improper Output Neutralization for Logs
- CWE-223 Omission of Security-relevant information
- CWE-532 Insertion of Sensitive Information into Log File

- CWE-778 Insufficient Logging

Validación de entrada

La falta de validación apropiada de las entradas del cliente son en gran parte, la debilidad de seguridad más común en aplicaciones web. Esta debilidad da espacio a todas las vulnerabilidades en las aplicaciones web, tales como inyección a interprete, ataques locale/Unicode, ataques al sistema de archivos, etc. La validación debe cubrir tanto los aspectos semánticos como sintácticos de la información ingresada.
Es aconsejable el uso de "White List" para validar el ingreso de datos permitidos. Para este fin, se utilizan listas de valores o expresiones regulares.

Autenticación en la Web

Su objetivo es proveer servicios de autenticación segura a las aplicaciones web mediante:

- La vinculación de una unidad del sistema a un usuario individual por medio de una credencial.
- Proveer controles de autenticación razonables de acuerdo al riesgo de la aplicación.
- Denegación del acceso a atacantes que usan varios métodos para atacar el sistema de autenticación.

Consideraciones Generales

- La autenticación es igual de fuerte como sus procesos de administración de usuarios.
- Use la forma mas apropiada de autenticación adecuada para su clasificación de bienes.
- Re-autenticar al usuario para transacciones de alto valor y acceso a áreas protegidas.
- Autenticar la transacción, no el usuario.
- Las contraseñas son trivialmente rotas y no son adecuadas para sistemas de alto valor.

Buenas Practicas

- User IDs
- Fortaleza de contraseñas
- Implementar métodos seguros de recuperación
- Almacenar contraseñas de forma segura
- Transmitir contraseñas solo sobre TLS
- Solicitar re- autenticación
- Utilizar sistemas de autenticación de factor múltiple
- Manejo de mensajes de error
- Prevenir ataques por fuerza bruta

Métodos de Protección

- MFA o Autenticación de factor múltiple
- Bloqueo de cuenta
- CAPTCHA
- Preguntas de seguridad o palabras clave

Técnicas de Autenticación

- Autenticación básica y segura (HTTP-Basic, HTTP-Digest)
- Autenticación basada en formularios (Usuario-Contraseña)
- Autenticación integrada (ISS, ASP.NET, Active Directory)
- Autenticación basada en certificado (x509)
- Autenticación fuerte o factor múltiple (dato personal)
- Re-uso de contraseñas
- Autenticación sin contraseñas

Contraseñas de un solo uso

- **OTP:** (One Time Password), corresponde a un valor confidencial que no puede ser reutilizado.
- **HOTP:** (HMAC-based One Time Password), algoritmo de generación de contraseñas de un solo uso basadas en HMAC. RFC-4226.
- **TOTP:** (Time-based One Time Password), valor confidencial que no puede ser reutilizado y que además cuenta con un tiempo de vida acotado. RFC-6238.

JWT – JASON Web Token

Estándar abierto utilizado para transmitir de forma segura información en formato JSON. Compuesto por 3 partes.

1. Encabezado que especifica el tipo de token y el algoritmo de firma
2. La carga útil que contiene los datos relevantes
3. La firma que verifica la integridad del token

Son utilizados por aplicaciones web y servicios API para la autenticación y autorización ya que son compactos, seguros y autocontenidos. Requieren una clave necesaria para verificar la firma del token y es posible utilizarlos en diferentes sistemas.

Contraseña Olvidada

- Utilizar un mensaje único ya sea que la cuenta exista o no
- Asegurar que el tiempo de respuesta al usuario sea uniforme
- Utilizar otro canal para informar el método de re-seteo de la contraseña
- Utilizar tokens con URLs para una implementación simple y rápida
- Asegurar que los tokens sean aleatorios, con longitud que provea resistencia, almacenamiento seguro, con tiempo de expiración y uso único.
- No alterar la cuenta del usuario hasta la presentación del token

Objetivos

- Asegurar que solamente usuarios autorizados puedan realizar acciones permitidas con su correspondiente nivel de privilegio
- Controlar el acceso a recursos protegidos mediante decisiones basadas en el rol o el nivel de privilegio
- Prevenir ataques de escalada de privilegios

Métodos de Control de Acceso

- **RBAC** (Role Based Access Control): Modelo de control de acceso que se utiliza para administrar los permisos y accesos de los usuarios dentro de un sistema o aplicación. Los permisos se asignan a roles en lugar de asignarlos directamente a usuarios individuales. Se definen diferentes roles basados en las funciones y responsabilidades de un individuo dentro de la organización o de la base de usuarios. (HIPAA, Gramm-Leach-Bliley, etc).
- **DAC** (Discretionary Access Control): Modelo de control de acceso donde los propietarios de los recursos son responsables de controlar el acceso a los mismos. Las decisiones de dichos accesos se basan normalmente en las credenciales que el usuario presentó en el momento de la autenticación. El inconveniente que presenta DAC es que los administradores no son capaces de gestionar de forma centralizada los permisos de los archivos/datos almacenados en el servidor web. (Sistemas de archivos de un sistema Unix (rwx))
- **MAC** (Mandatory Access Control): Modelo de control de acceso en el que el acceso a los recursos se basa en reglas y políticas predefinidas por un administrador del sistema. Las políticas de acceso son impuestas por el sistema operativo o por una entidad de seguridad centralizada. Asegura la información mediante la asignación de etiquetas. Los recursos también poseen etiquetas que indican la sensibilidad en la información o clasificación de los datos. Es utilizado para sistemas totalmente seguros. (Aplicaciones militares seguras multinivel o aplicaciones de datos de misión crítica).
- **ABAC** (Attribute Based Access Control): Modelo de control de acceso que utiliza atributos como base para tomar decisiones de autorización. Se centra en evaluar los atributos relevantes de los sujetos, recursos y contextos para determinar si se permite o deniega el acceso al recurso.

Buenas Prácticas de Implementación

- Codificar el control en la actividad objetivo
- Disponer de un Controlador Centralizado (ACL)
- Utilizar un Control Central de Acceso en las diferentes capas
- Verificar la política del lado del servidor (Server-side)

Ataques de Control de Acceso

Intentos maliciosos de eludir o comprometer los mecanismos de control de acceso de un sistema con el objetivo de obtener acceso no autorizado a recursos, datos o funcionalidades.

- **Vertical Access Control Attacks:** Un usuario convencional obtiene accesos superiores o de administrador
- **Horizontal Access Control Attacks:** Con el mismo rol o nivel el usuario puede acceder a información de otros usuarios
- **Business Logic Access Control Attacks:** Abusar de una o más actividades para realizar una operación con un resultado no autorizado para ese usuario

Administración de Usuarios y Privilegios

Objetivos

- Las funciones de nivel de administrador están segregadas apropiadamente de la actividad del usuario
- Los usuarios no pueden acceder o utilizar funcionalidades administrativas
- Proveer la necesaria auditoria y trazabilidad de funcionalidad administrativa

Mejores Practicas

- En el diseño, trazar la funcionalidad administrativa fuera y asegurarse que los controles apropiados de acceso y auditoria están en su lugar
- Considerar todos los procesos teniendo en cuenta que los usuarios puedan ser prevenidos de utilizar una característica especifica
- Acceso de servicio de asistencia
- Diseñar con cuidado la funcionalidad de servicio de asistencia/moderador/soporte al cliente, alrededor de una capacidad administrativa limitada y aplicación segregada o acceso
- Todos los sistemas deberían tener un acceso para administradores separado del resto de los usuarios
- Sistemas de alto valor deberían estar separados del resto de los sistemas en un servidor aparte. El ingreso a los mismos debería ser por medio de una VPN o red de un centro de operaciones de confianza

DevSecOps

Conjunto de practicas que combinan el desarrollo de software (***Dev***), seguridad ***(Sec***) y operaciones de tecnología de la información (***Ops***) para asegurar y acortar el ciclo de vida del desarrollo de software.

Web Services

Nivel más simple, los servicios web pueden ser vistos como aplicaciones web especializadas que difieren principalmente en la capa de presentación. Basados en XML/SOAP. Los servicios Web típicamente representan una interfaz publica funcional que se llama de forma programática.

Comités de estándares

- **W3C**
- **OASIS**, estándares de WS-Security
- **OASIS,** UDDI
- **OASIS,** SAML
- **Grupo de interoperabilidad de servicios web** WS-L

WS-Security -WSS

Especificación estándar que se utiliza en servicios web para proporcionar seguridad en la comunicación y proteger la integridad y confidencialidad de los datos transmitidos. Define un conjunto de extensiones y mecanismos de seguridad que se pueden utilizar en conjunto con los protocolos SOAP y XML. Las áreas principales definidas por el estándar son:

- Maneras de agregar encabezados de seguridad a los sobres de SOAP
- Adjuntar testigos de seguridad y credenciales al mensaje
- Insertando un estampado de tiempo
- Firmar el mensaje
- Cifrado del mensaje
- Extensibilidad

Aspectos

- **WS-Policy:** Describe capacidades y limitaciones de la seguridad, políticas e intermediarios.
- **WS-Trust:** Describe un framework para facilitar la interoperación de WS en forma segura.
- **WS-Privacy:** Describe el modelo sobre como los WS manejan las peticiones y preferencias de seguridad.
- **WS-SecureConversation:** Describe como manejar y autenticar el intercambio de mensajes, contexto de seguridad y claves de sesión.
- **WS-Federation:** Describe como administrar y manejar las relaciones de confianza entre sistemas federados.
- **WS-Authorization:** Describe como administrar la autorización de datos y políticas.

ReST

Estilo de arquitectura para diseñar sistemas distribuidos y servicios web. Los recursos se representan mediante identificadores únicos (URL) y se accede a ellos mediante operaciones HTTP estándar como GET, POST, PUT y DELETE

SEGURIDAD EN CI/CD (DevOps/DevSecOps)

CI
Integración continua, implica la integración regular y automatizada del código fuente de un equipo de desarrollo en un repositorio centralizado. Cuando se realiza una integración, se ejecutan pruebas automatizadas para verificar la calidad del código y posibles problemas.

CD
Entrega continua, refiere a la automatización del proceso de entrega de software. Una vez pasado por el CI, se puede construir automáticamente, probar y empaquetar en un formato listo para su implementación.

1. Mecanismos de Control de Flujo Insuficientes

Definición
Refieren a la capacidad de un atacante que ha obtenido permisos debido a la falta de mecanismos que hagan cumplir una aprobación o revisión adicional.

Descripción
Las organizaciones introducen continuamente medidas y controles destinados a garantizar que ninguna entidad, ya sea humana o aplicación, pueda enviar código o artefactos a través de la canalización sin tener que someterse a un conjunto estricto de revisiones y aprobaciones.

Impactos
Un atacante puede abusar de los insuficientes mecanismos de control de flujo para implementar artefactos maliciosos. Estos artefactos se envían a través del pipeline potencialmente hasta la producción sin ninguna aprobación o revisión.

Recomendaciones
Establecer mecanismos de control de flujo en el pipeline para garantizar que ninguna entidad individual pueda enviar código y artefactos confidenciales a través del pipeline sin verificación o validación externa.

- Limitar el uso de reglas de fusión automática y asegurarse de que sean aplicables a la cantidad mínima de contextos.
- Evitar que las cuentas desencadenen canalizaciones de desarrollo e implementación de producción sin aprobación o revisión adicional.

- Permitir que los artefactos fluyan a través de la canalización solo con la condición de que hayan sido creados por una cuenta de servicio de CI preaprobada.

2. Gestión Inadecuada de Identidad y Acceso

Definición
Derivan de las dificultades para gestionar la gran cantidad de identidades repartidas en los diferentes sistemas de ecosistema. La existencia de identidades mal administradas aumenta el potencial y el alcance del daño.

Descripción
Los procesos de entrega de software consisten en múltiples sistemas conectados entre si con el objetivo de mover el código y los artefactos desde el desarrollo hasta la producción. Cada sistema proporciona múltiples métodos de acceso e integración. Los diferentes tipos de cuentas y métodos de acceso pueden tener potencialmente su propio método de aprovisionamiento único, conjunto de políticas de seguridad y modelo de autorización.

Impacto
La existencia de cientos de identidades junto con la falta de solidas prácticas de administración de acceso e identidad, podrían otorgar poderosas capacidades en el entorno y podría servir como transición al entorno de producción.

Recomendaciones

- Llevar un análisis y un mapeo continuo de todas las identidades en todos los sistemas dentro del ecosistema de ingeniería.
- Eliminar los permisos que no sean necesarios para el trabajo en curso de cada identidad.
- Determinar un periodo aceptable para deshabilitar/eliminar cuentas obsoletas.

3. Abuso sobre la Cadena de Dependencias

Definición
Capacidad de un atacante para abusar de las fallas relacionadas con la forma en que las estaciones de trabajo y los entornos de construcción obtienen las dependencias del código.

Descripción
Dada la cantidad total de sistemas involucrados en el proceso en todos los contextos de desarrollo hace que la gestión de dependencias y paquetes externos utilizados por el código escrito hace cada vez mas complejo el desarrollo. Hay que detectar el uso de paquetes con vulnerabilidades conocidas y realizar análisis estáticos de código escrito por ellos mismos y terceros. Vectores de ataque:

- **Dependency Confussion:** Confusión de dependencia
- **Dependecy Hijacking:** Secuestro de dependencias

- **Typosquatting:** Similitud de tipeo/nombre
- **BrandHijacking:** Robo de marca

Impacto
El objetivo de los adversarios es ejecutar un código malicioso en un host que extrae el paquete, se puede aprovechar para el robo de credenciales y el movimiento lateral dentro del entorno en el que se ejecuta. También se puede dar que el código malicioso llegue a los entornos de producción haciendo muy difícil su detección.

Recomendaciones

- No se debe permitir que ningún cliente extraiga paquetes de código para obtener paquetes directamente de Internet o de fuentes no confiables.
- Habilitar la verificación de la suma de control y la verificación de la firma para los paquetes extraídos.
- Evitar configurar clientes para extraer la última versión de un paquete.
- Asegurarse que todos los paquetes privados estén registrados bajo el alcance de la organización.
- Asegurar que exista un contexto separado para los scripts.
- Asegurarse que los proyectos internos siempre contengan archivos de configuración de administradores de paquetes.
- Evitar publicar nombres de proyectos internos en repositorios públicos
- Prevención completa del abuso de la cadena por parte de terceros.
- Todos los sistemas deben fortalecerse de acuerdo con las pautas bajo el riesgo “CICD-SEC-7”

4. Ejecución de un Pipeline Envenenado

Definición
Capacidad de un atacante con acceso a los sistemas de control de código fuente para manipular el proceso de construcción mediante la inyección de códigos/comandos maliciosos en la configuración de la pipeline de construcción.

Descripción
El vector PPE abusa de los permisos contra un repositorio SCM (Sistema de Control de Versiones), de manera tal que hace que una pipeline de CI ejecute comandos maliciosos. Los usuarios que tienen permiso para manipular los archivos de configuración de CI, pueden modificarlos para que contengan comandos maliciosos y “envenenar” la pipeline de CI que ejecuta estos comandos.

- **PPE directo (D-PPE):** el atacante modifica el archivo de configuración de CI en un repositorio al que tiene acceso.
- **PPE indirecto (I-PPE):** la posibilidad de D-PPE no esta disponible para un adversario con acceso a un repositorio SCM.
- **PPE publico (3PE):** la ejecución de un ataque PPE requiere acceso al repositorio que aloja el archivo de configuración del pipeline o a los archivos a los que hace referencia.

Impacto

- Acceso a cualquier secreto disponible para el trabajo de CI
- Acceso a activos externos
- Capacidad de enviar código y artefactos bajo la apariencia de código legitimo generado por el proceso de compilación
- Capacidad de acceder a hosts y activos adicionales en la red/entorno del nodo de trabajo

Recomendaciones

- Asegurar que pipelines que ejecutan código no revisado se ejecuten en nodos aislados.
- Evaluar la necesidad de desencadenar pipelines en repositorios públicos de colaboradores externos.
- Revisar cada archivo de configuración de CI antes de ejecutar el pipeline
- Eliminar los permisos otorgados en el repositorio de SCM a los usuarios que no los necesitan.
- El pipeline debe tener acceso a las credenciales que necesita únicamente.

5. Insuficientes Controles de Acceso Basados en Pipeline

Definición
Al ejecutar código malicioso dentro de un pipeline, los atacantes aprovechan los riesgos de PBAC (Controles de acceso basados en pipeline) insuficientes para abusar del permiso otorgado a la pipeline para moverse lateralmente dentro o fuera del sistema CI/CD.

Descripción
Los nodos que ejecutan pipeline llevan a cabo los comandos especificados en la configuración del pipeline y al hacerlo, realizan una amplia gama de actividades confidenciales, tales como:

- Acceder al código fuente
- Obtener secretos de varias ubicaciones como variables de entorno, etc
- Crear, modificar y desplegar artefactos.

` Es imperativo limitar cada pipeline al conjunto exacto de datos y recursos a los que necesita acceder. Para ello, el PBAC incluye controles como:

- Acceso dentro del entorno de ejecución del pipeline
- Permisos para el host subyacente y otros nodos de pipeline
- Filtros de ingreso y egreso a internet

Impactos
Al ejecutarse el código malicioso en el pipeline, se obtiene acceso a secretos, host subyacente y mismo, conectarse a cualquiera de los sistemas a los que tiene acceso dicho

pipeline. Esto da lugar a la exposición de datos confidenciales, movimiento lateral dentro del entorno de CI y la implementación de artefactos maliciosos en el proceso, incluida la producción. El alcance del daño potencial está determinado por la granularidad del PBAC en el entorno.

Recomendaciones

- No utilizar un nodo compartido para pipelines con diferentes niveles de sensibilidad.
- Determinar que el pipeline solo tenga acceso a los secretos que necesita.
- Revertir el nodo de ejecución a su estado original después de cada ejecución de pipeline
- Asegurar que al usuario del SO que ejecuta el trabajo de pipeline se le hayan otorgado permisos de sistema operativo.
- Parches correspondientes.
- Asegurar la segmentación de la red en el entorno en el que se ejecuta el trabajo.

6. Insuficiente "Higiene" de Credenciales

Definición
Capacidad de un atacante para obtener y usar varios secretos y tokens distribuidos a lo largo del pipeline debido a fallas que tienen que ver con los controles de acceso alrededor de las credenciales.

Descripción
Debido a que los entornos de CI/CD están construidos con múltiples sistemas que se comunican y se autentican entre si, es difícil manejar la protección de las credenciales. Algunas fallas importantes son:

- Código que contiene credenciales que se envían a una de las ramas de un SCM
- Credenciales utilizadas de forma insegura dentro del repositorio de procesos de compilación e implementación
- Credenciales en capas de imágenes de contenedores
- Credenciales impresas en la salida de la consola
- Credenciales no rotadas

Impacto
Las credenciales son el objeto mas codiciado por los atacantes y esto es debido al gran potencial de error humano, junto con la falta de conocimiento sobre la gestión segura de credenciales. Esto pone en riesgo los recursos de alto valor de muchas organizaciones.

Recomendaciones

- Establecer procedimientos para mapear continuamente las credenciales.
- Evitar compartir el mismo conjunto de credenciales en varios contextos.
- Optar por usar credenciales temporales sobre credenciales estáticas.

- Asegurar que los secretos que se utilizan en los sistemas de CI/CD tengan un alcance que permita que cada pipeline y paso tenga acceso solo a los secretos que necesita.
- Opciones de proveedores integradas o herramientas de terceros para evitar que los secretos se impriman en las salidas de la consola.

7. Configuración Insegura del Sistema

Definición
Derivan en fallas en la configuración de seguridad y el hardening de los diferentes sistemas a lo largo del pipeline.

Descripción
Para optimizar la seguridad, los defensores deben poner mucho énfasis tanto en el código y los artefactos que fluyen a través del pipeline como en la postura y la resistencia de cada sistema individual. Posibles defectos de hardening:

- Versión desactualizada o carece de parches de seguridad importantes.
- Controles de acceso a la red demasiado permisivos.
- Configuraciones de sistema inseguras.
- Higiene de credenciales inadecuada.

Impacto
El atacante puede comprometer el sistema y acceder al sistema operativo subyacente. Puede abusar de fallas para manipular flujos de CI/CD legítimos, obtener tokens confidenciales, acceder a entornos de producción, permitir que se mueva lateralmente dentro del entorno y fuera del contexto de los sistemas de CI/CD.

Recomendaciones

- Mantener un inventario de sistemas y versiones en uso incluyendo mapeo de un propietario designado para cada sistema. Verificar vulnerabilidades conocidas.
- Actualizar el componente vulnerable.
- Garantizar acceso de red a los sistemas que estén alineados con el principio de acceso mínimo
- Revisar periódicamente todas las configuraciones del sistema

8. Usos No Controlados de Servicios de Terceros

Definición
Se basan en la extrema facilidad con la que se puede otorgar acceso a un servicio de terceros a los recursos en los sistemas de CI/CD.

Impacto
La implementación insuficiente de RBAC y los privilegios mínimos en torno a terceros crean un aumento significativo de la superficie de ataque a la organización.

Recomendaciones

- **Aprobación:** Los terceros deben ser aprobados previo al acceso del entorno.
- **Integración:** Introducir controles y procedimientos para mantener la visibilidad continua de todos los terceros.
- **Visibilidad sobre el uso continuo:** asegurarse de que el tercero este limitado a los recursos específicos a los que requiere acceso.
- **Desaprovisionamiento:** revisar periódicamente todos los terceros integrados y elimine los que ya no estén en uso.

9. Validación Inapropiada de Integridad de Artefactos

Definición
Permiten que un atacante con acceso a uno de los sistemas en el proceso de CI/CD envíe códigos o artefactos maliciosos por la canalización debido a la falta de garantías de validación del código y artefactos.

Descripción
El hecho de que el recurso final dependa de múltiples fuentes repartidas en los diferentes pasos, proporcionadas por múltiples contribuyentes, crea múltiples puntos de entrada a través de los cuales se puede manipular este recurso final.

Impacto
El atacante con un punto de apoyo en el proceso de entrega de software, puede abusar de la validación de integridad de artefactos incorrecta para enviar un artefacto malicioso a través de la canalización pudiendo llegar a comprometer la producción.

Recomendaciones

- Implementar procesos y tecnologías para validar la integridad de los recursos desde el desarrollo hasta la producción.
    - Firma de código
    - Software de verificación de artefactos
    - Detección de cambios de configuración
- Los recursos de terceros obtenidos en canalizaciones de compilación /implementación deben seguir una lógica similar.

10. Insuficiente Logueo y Visibilidad

Definición
Permiten que un atacante lleve a cabo actividades maliciosas sin ser detectado durante ninguna fase de la cadena de destrucción del ataque, incluida la identificación de las TTP.

Descripción
Debido a la alta cantidad de posibles vectores de ataque que aprovechan los entornos y procesos, es imperativo que los equipos de seguridad desarrollen las capacidades adecuadas para detectar estos ataques tan proto como ocurran.

Recomendaciones

- Mapeo del entorno
- Identificar y habilitar las fuentes de registro adecuadas
- Envío de registros a una ubicación centralizada para respaldar la agregación y correlación de registros.
- Crear alertas al detectar anomalías y posibles actividades maliciosas.