Informe de Auditoría de Seguridad Web

08.12.2024

—

Vanesa Daniela ALTAMIRANO

Curso: Introducción a la Ciberseguridad

Academia: KeepCoding

Versión del informe: 1.0

Índice

1. Ámbito y Alcance

- Aplicación auditada: WebGoat (versión 8.1.0).

- Entorno de pruebas:

- **Sistema Operativo: Kali GNU/Linux Rolling**
  - Versión: 2024.4 -
  - Arquitectura: ARM64
  - Kernel: Linux 6.11.2-arm64
  - Entorno: Máquina virtual (QEMU) sobre UTM en macOS
- **Herramientas**: Docker, Burp Suite, sqlmap, nmap.
- **Navegador**: Firefox 115.1.0

- Objetivo: Identificar y documentar vulnerabilidades críticas presentes en la aplicación.

2. Informe Ejecutivo

a. Breve Resumen

La auditoría realizada sobre WebGoat tuvo como objetivo identificar vulnerabilidades críticas y evaluar el impacto que estas podrían tener en un entorno de producción.

b. Vulnerabilidades Destacadas

Vulnerabilidad	Impacto	Nivel de Riesgo	Posible impacto
SQL Injection	Exposición de datos	ALTO	Robo de información
Cross-Site Scripting	Robo de sesiones	MEDIO	Suplantación de identidad de los usuarios

Security Misconfiguration	Accesos no autorizados	ALTO	Filtración de datos confidenciales

c. Conclusiones

La aplicación WebGoat analizada posee vulnerabilidades graves que podrían comprometer la seguridad de los datos.

d. Recomendaciones

1. Es necesario implementar validaciones en el servidor para prevenir inyecciones tales como SQL y XSS.

2. Se deben mantener actualizados los componentes desactualizados para reducir el riesgo de exploits conocidos.

3. Llevar a cabo el chequeo de las configuraciones del servidor para eliminar configuraciones inseguras.

3. Descripción del Proceso de Auditoría

a. Reconocimiento (Information Gathering)

- Puertos abiertos:

Puerto	**Estado**	**Servicio**
8080	Open	http-proxy
9090	Open	zeus-admin

—**Recomendación**: Se debe considerar mantener abiertos solo los puertos esenciales para el funcionamiento de la aplicación. En caso de ser necesario mantenerlos abiertos, usar un firewall para hacer mas seguro el acceso a los puertos y mantener un monitoreo. Se pueden realizar pruebas de vulnerabilidades sobre los mismos.

- Sistema Operativo del servidor: Linux version 2.6.32 (escaneo con Nmap)

- Tecnologías identificadas (información obtenida a través del acceso a logs del contenedor Docker):

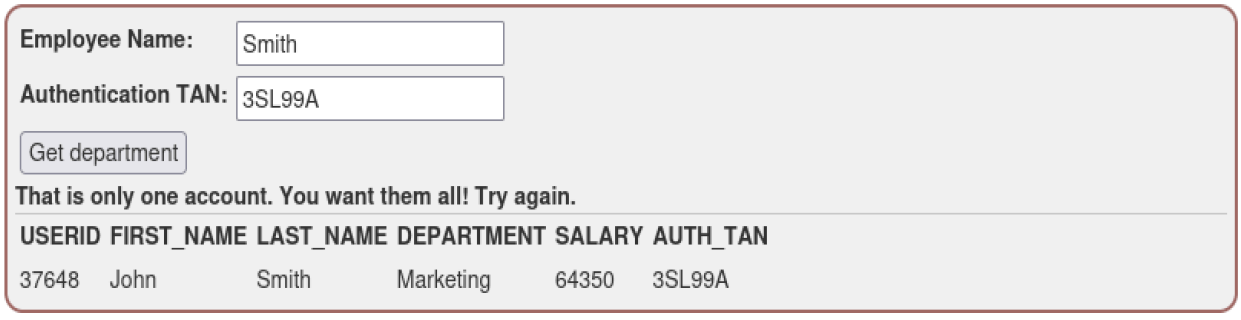
  -Lenguaje de Programación: JAVA.

  -Frameworks: Spring Boot, Jetty.

b. Explotación de Vulnerabilidades

1. SQL Injection (A3 Injection - Apartado 11):

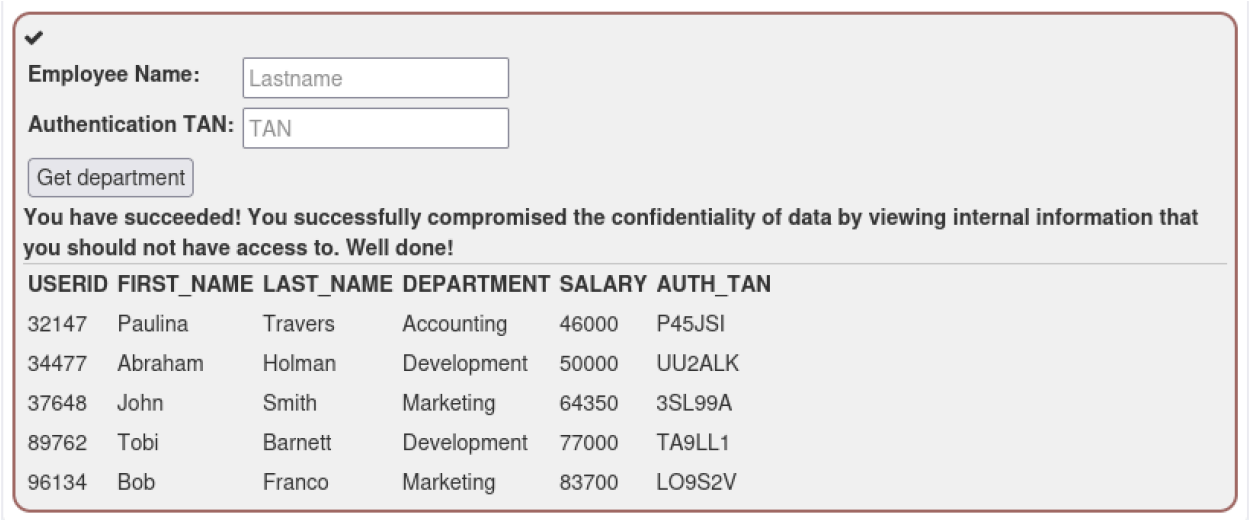
Prueba realizada: Se procedió a verificar que al realizar la consulta con los datos de un Usuario determinado el resultado sea solo su información personal.



Posteriormente y basado en el conocimiento de la estructura de la consulta en el backend, se reemplazó en el campo ¨Lastname¨el siguiente texto:

' SELECT * FROM employees –

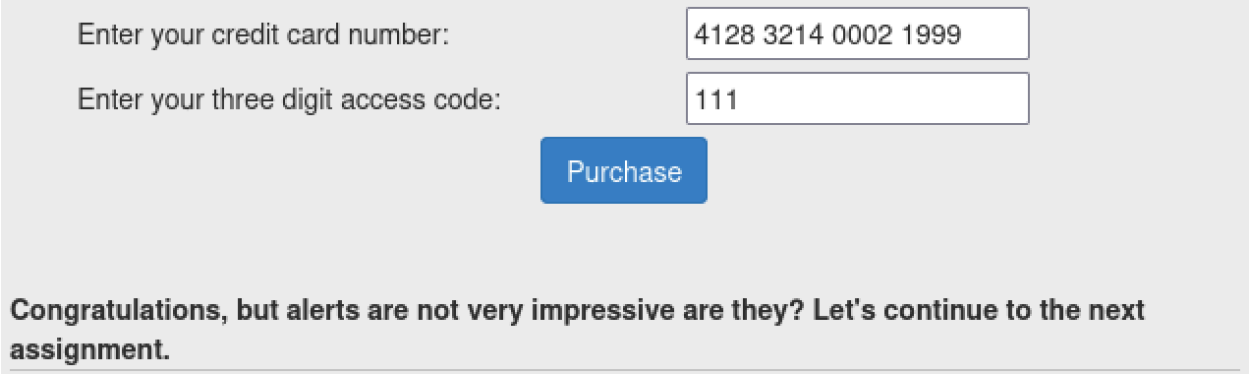
De esta manera, combinando la consulta original con caracteres como comillas o guiones que sirven para evitar errores de sintaxis se pudo acceder a la totalidad de la tabla sin ninguna otra forma de validación.



Resultado: Se vulneró el acceso a la base de datos de los empleados.

2. Cross-Site Scripting (A3 Injection - Apartado 7):

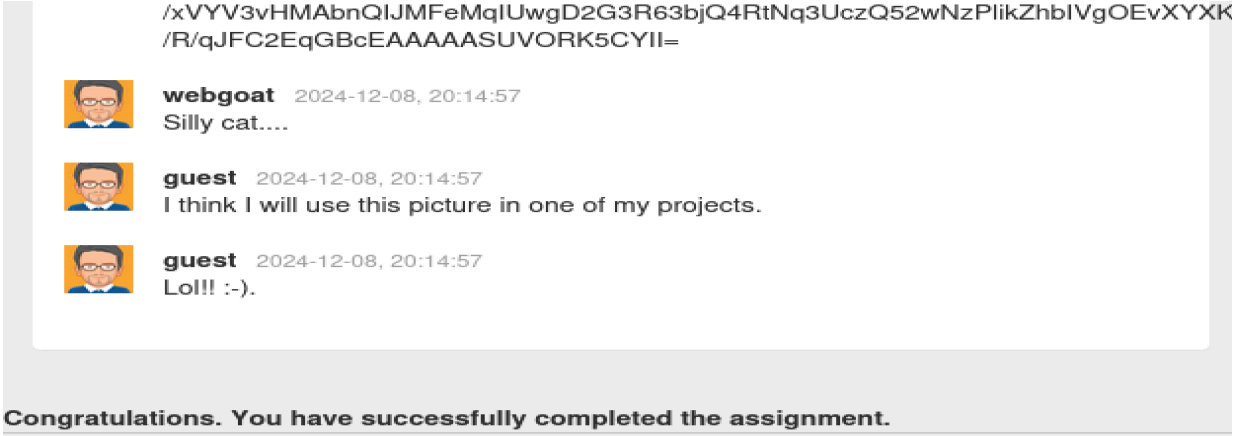
Prueba realizada: Se ingreso un dato en uno de los campos de entrada usando métodos básicos de scripting **<script>** y alert().



Resultado: Se logró la inyección de código malicioso demostrando una vulnerabilidad en XSS.

3. Security Misconfiguration (A5 - Apartado 4):

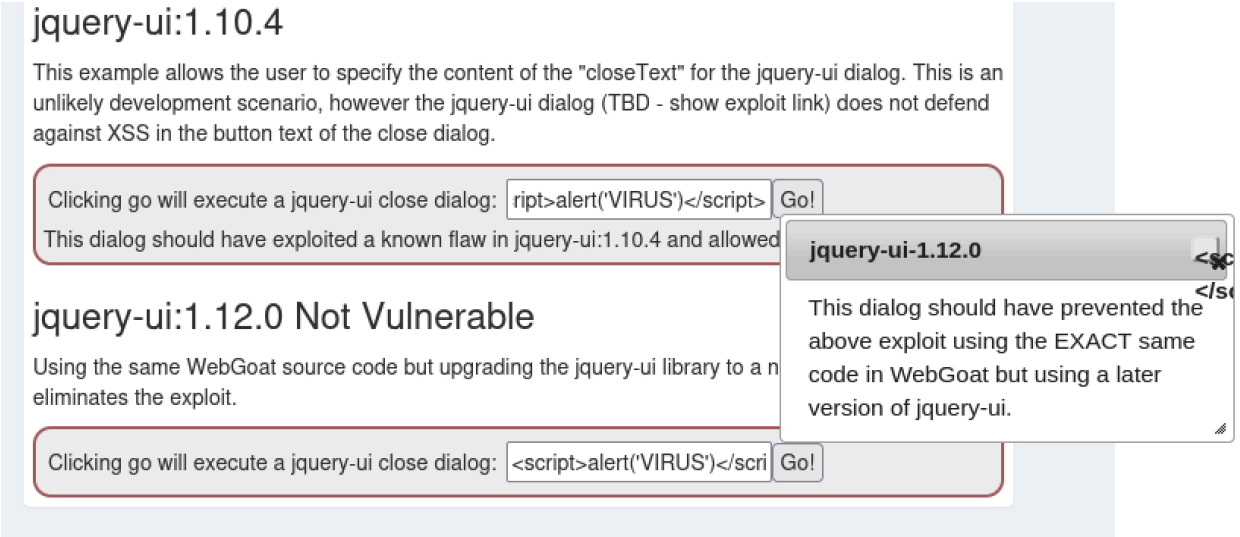
Prueba realizada: se inspeccionó el elemento para determinar el directorio raíz del mismo y se ha colocado este directorio en el campo “Add a comment”, buscando identificar si existe una vulnerabilidad de en XXE injection.



Resultado: El servidor no está configurado correctamente y puede exponer información sensible o confidencial.

4. Vuln & outdated Components (A6 - Apartado 5):

Prueba realizada: Se utilizaron los mismos fragmentos de código con diferentes versiones de JQuery-UI para verificar la existencia de vulnerabilidades XSS.



Resultado: Se demuestra que al usar versiones actualizadas se eliminan las vulnerabilidades por lo cual se aconseja realizar actualizaciones y realizar pruebas periódicas.

5. Identity & Auth Failure A7 - Secure Passwords Apartado 4

Prueba realizada: Comprobar si las contraseñas del listado proporcionado son suficientemente seguras para evitar la revelación de las mismas. Se probaron las contraseñas determinando los motivos por los cuales eran inseguras.

Resultado: Se generó una nueva contraseña tal como ElGat0gr3y/, en la cual la longitud, la mezcla de caracteres alfanumericos en mayusculas y minusculas, más el uso de caracteres especiales completó el score solicitado de 4/4, mostrando un aumento considerable en el tiempo estimado de descifrado de contraseñas.

c. Post-Explotación

- Información obtenida luego de la explotación de vulnerabilidades:

- Bases de datos comprometidas
- Información sensible obtenida
- Cookies de sesión de usuarios
- Suplantación de identidad
- Credenciales expuestas

-Posibles impactos:

- Robo o exposición de datos
- Inserción de códigos maliciosos

d. Posibles Mitigaciones

Vulnerabilidad	**Mitigación**
SQL Injection	Uso de consultas parametrizadas
Cross-site Scripting	Validaciones en entrada de usuario
Security Misconfiguration	Configuración segura del servidor

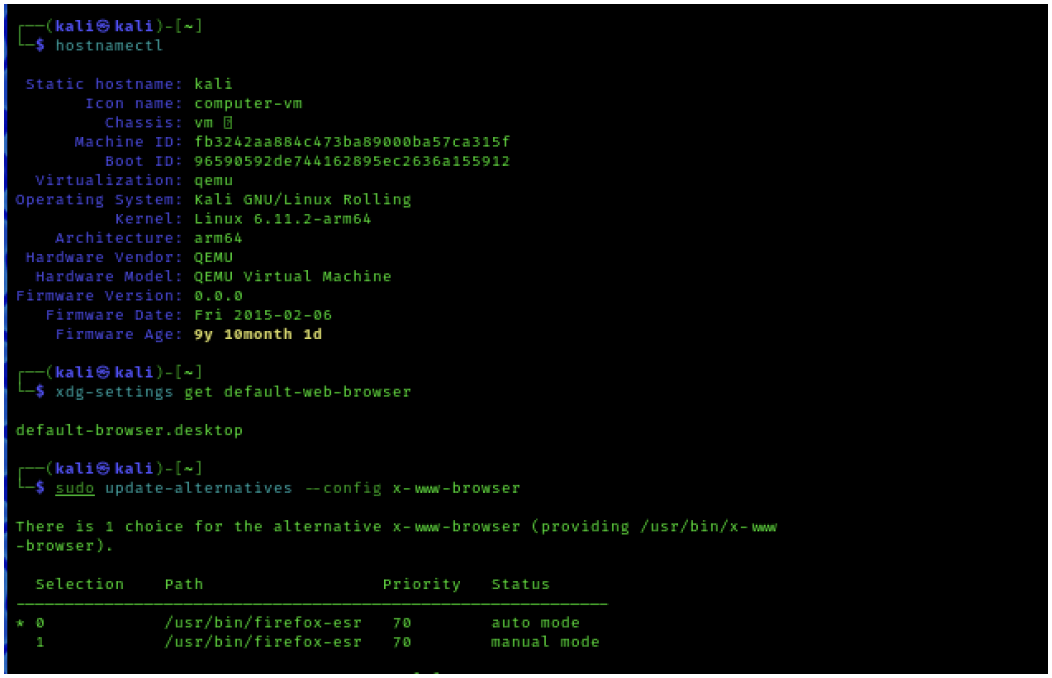
e. Herramientas Utilizadas

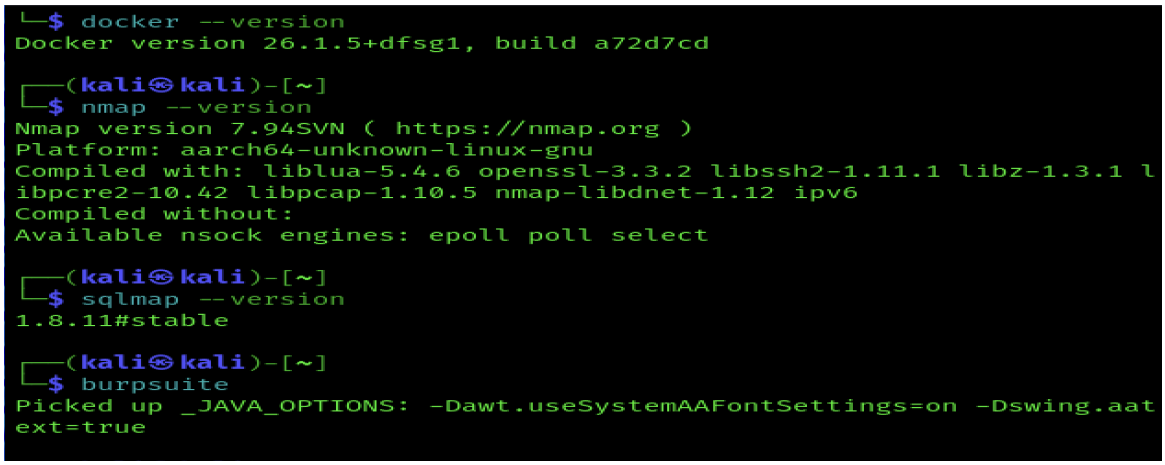
Herramienta	**Versión**	**Uso principal**
Docker	26.1.5	Despliegue de WebGoat
Nmap	7.94SVN	Escaneo de puertos
Burp Suite	Community	Interceptar tráfico HTTP
Sqlmap	1.8.11	Explotar vulnerabilidades SQL

4. Anexos

a. Evidencias

- Verificación de la configuración del entorno.





- Identificación de puertos abiertos y detección de sistema operativo.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 127.0.0.1

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-07 10:05 PST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000062s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
8080/tcp open  http-proxy
9090/tcp open  zeus-admin
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https:
//nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.42 seconds
```

- Búsqueda de información sobre lenguajes y frameworks

```
┌──(kali㉿kali)-[~]
└─$ docker logs webgoat

2024-12-07T13:40:31.200+01:00  INFO 1 --- [           main] org.owasp
.webgoat.server.StartWebGoat     : Starting StartWebGoat v2023.8 using
 Java 21.0.1 with PID 1 (/home/webgoat/webgoat.jar started by webgoat
 in /home/webgoat)
2024-12-07T13:40:31.205+01:00  INFO 1 --- [           main] org.owasp
.webgoat.server.StartWebGoat     : No active profile set, falling back
 to 1 default profile: "default"
2024-12-07T13:40:31.407+01:00  INFO 1 --- [           main] org.owasp
.webgoat.server.StartWebGoat     : Started StartWebGoat in 0.385 secon
ds (process running for 0.639)

2024-12-07T13:40:31.436+01:00  INFO 1 --- [           main] org.owasp
.webgoat.server.StartWebGoat     : No active profile set, falling back
 to 1 default profile: "default"
2024-12-07T13:40:31.833+01:00  INFO 1 --- [           main] .s.d.r.c.
RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repos
itories in DEFAULT mode.
```

b. Referencias

1. OWASP Top 10: [https://owasp.org/Top10/]

2. Documentación de WebGoat: [https://github.com/WebGoat/WebGoat]