NACHOGROUP

Informe de auditoría y ejercicios

WebGoat

02 julio, 2023 – Versión 1.0

Realizado para

Keepcoding – Introducción a Ciberseguridad

Realizado por

Juan Ignacio González Zamorano

Sinopsis

A mediados de junio, NachoGroup empezó a cursar un Bootcamp de Ciberseguridad en la escuela de Keepcoding (keepcoding.io). El primer módulo de este curso es Introducción en Ciberseguridad, durante el cuál se solicitó la entrega de una práctica, consistente en realizar un informe de auditoría y distintos ejercicios en la aplicación web WebGoat, con fecha límite de entrega 02/07/2023.

Para realizar con éxito de esta práctica, se ha usado como entorno controlado VirtualBox y Kali Linux, dentro del cual, se han utilizado la consola, la herramienta BurpSuite, y las extensiones en Firefox de Wappalyzer y Shodan.

Objetivo

La meta de esta práctica es realizar con éxito los ejercicios propuestos por la escuela Keepcoding, así como recabar información de puertos, S.O., y programas utilizados en la aplicación web WebGoat, y la creación de este informe de auditoría.

Ejercicios propuestos a resolver:

- A3 Injection - SQL Injection (intro) – Apartado 10
- A3 Injection - SQL Injection (intro) – Apartado 11
- Intenta obtener toda la información que puedas de la base de datos utilizando los fallos disponibles en la sección A3 Injection – SQL Injection
- A3 Injection – Cross Site Scripting – Apartado 7
- A5 Security Miconfiguration – Apartado 4
- A5 Security Miconfiguration – Apartado 7
- A6 Vuln & outdated Components – Apartado 5
- A7 Identity & Auth Failure – Secure Passwords – Apartado 4

Conclusiones

Durante la práctica se ha aprendido a manejar con mayor soltura la consola de Kali y la utilización de sus comandos, como nmap y sqlmap, así como la herramienta BurpSuite.

También, gracias a la aplicación web WebGoat, se ha podido aprender y comprender los ataques mediante el uso de inyecciones, tanto para SQL, como JavaScript y HTML, y la importancia de tener bien preparadas, programadas y actualizadas las bases de datos y las aplicaciones web, para poder prevenir este tipo de ataques.

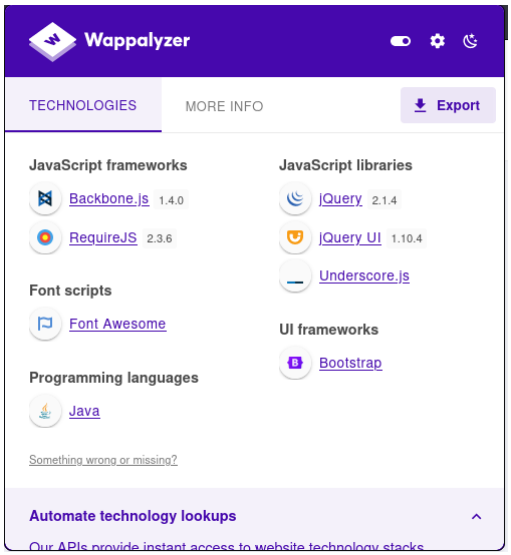
Finalmente, se ha generado mayor conciencia sobre la creación de contraseñas de mejor seguridad, para proteger la información privada.

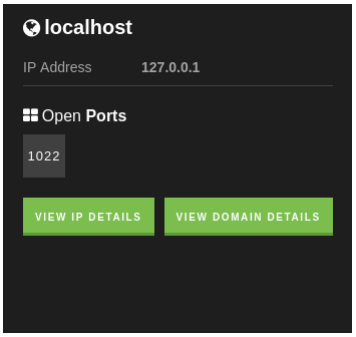
Información WebGoat

Utilizando el comando **nmap -O** en Kali si ha podido recabar información sobre los puertos abiertos de la aplicación 8080, 8081 y 9090. También se ha podido averiguar que el sistema operativo es Linux.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 127.0.0.1
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-02 07:57 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000082s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
8080/tcp open  http-proxy
8081/tcp open  blackice-icecap
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
```

A través de las extensiones Firefox de **Wappalyzer** y **Shodan** se ha podido averiguar que WebGoat utiliza JavaScript, su IP es 127.0.0.1y tiene otro puerto abierto 1022.

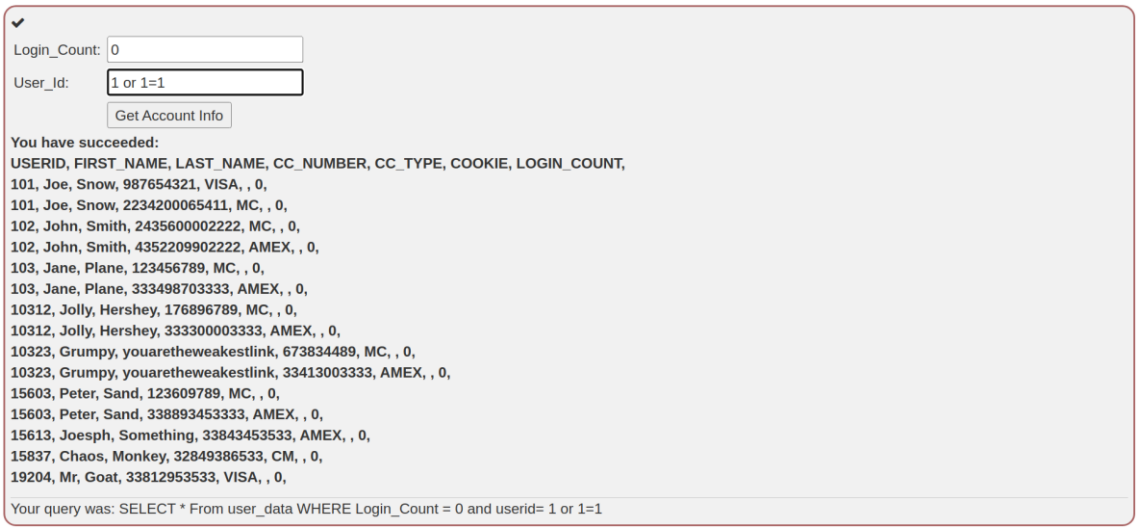




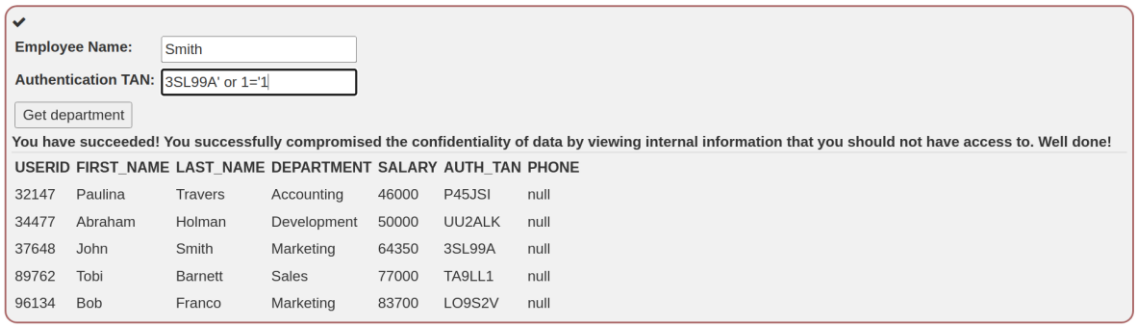
Ejercicios de WebGoat

A3 Injection - SQL Injection (intro) – Apartado 10 y 11

Apartado 10 – Se tratada de comprobar cuál de los dos campos es susceptible de ser inyectado. Intentamos inyectar el primer campo, pero no es susceptible de una inyección SQL, por lo que solo nos deja como opción el segundo. Tras inyectar correctamente el campo User_Id obtenemos todos los datos de la tabla.

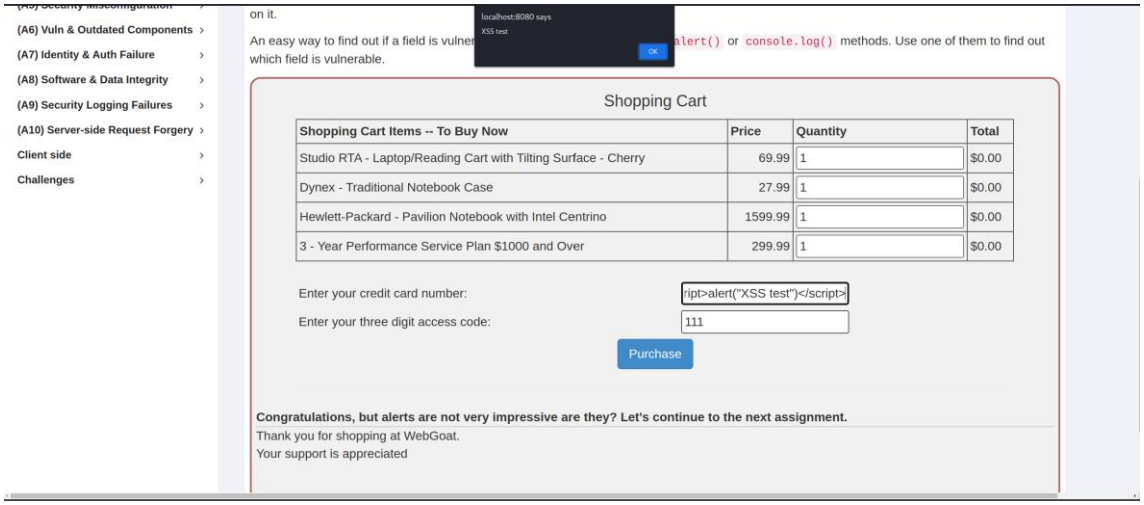


Apartado 11 – Parecido al ejercicio anterior. Se tratada de comprobar cuál de los dos campos es susceptible de ser inyectado. Intentamos inyectar el primer campo, pero no es susceptible de una inyección SQL, por lo que solo nos deja como opción el segundo. Tras inyectar correctamente el campo Authentication_TAN obtenemos todos los datos de la tabla.



A3 Injection – Cross Site Scripting – Apartado 7

Se trata de averiguar que campo es susceptible de ser comprometido, sólo se puede escribir en dos campos, probamos con el de la tarjeta de crédito insertando lo que nos comentan en la página 2 **<script>alert("XSS test")</script>** y nos devuelve el valor insertado, por lo que ese campo es susceptible de comprometerse, por lo que ahí insertaríamos nuestro código.



A5 Security Miconfiguration – Apartado 4 y 7

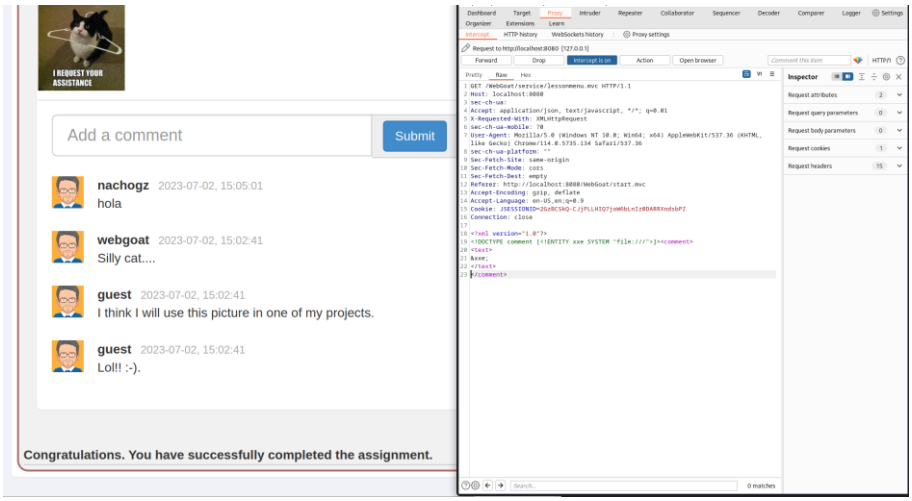
Apartado 4 – Se trata de ejecutar una inyección XXE en el campo de comentarios para extraer el directorio root del filesystem. Mediante la herramienta BurpSuite interceptamos la petición y sustituimos el comentario interceptado,

<?xml version="1.0"?><comment> <text>hola</text></comment>

por

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

Tras este cambio, volvemos a lanzar la petición y completamos con éxito el ejercicio.

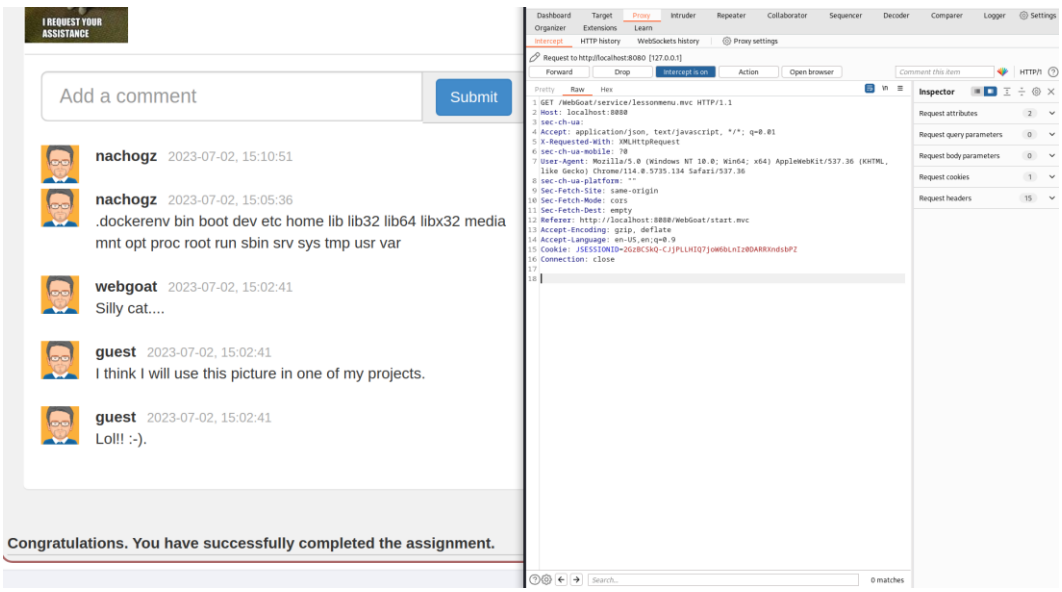


Apartado 7 – Ejercicio parecido al anterior, pero ejecutando una inyección XML. Utilizando el proceso del apartado 4, interceptamos la petición y sustituimos el comentario interceptado,

<?xml version="1.0"?><comment> <text>hola</text></comment> Por

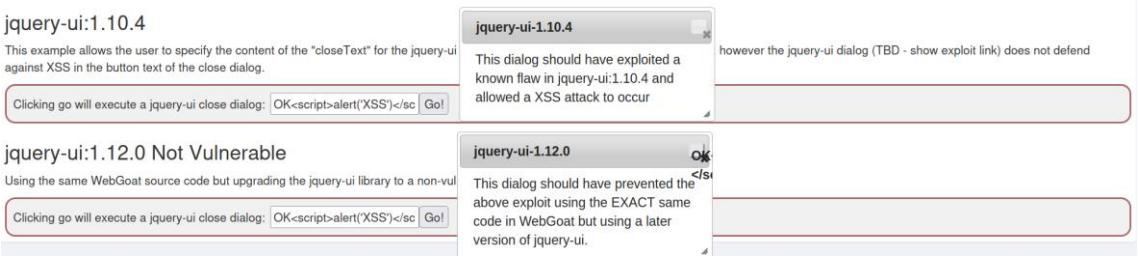
<?xml version="1.0"?><!DOCTYPE comment [<!ENTITY xxe SYSTEM "file:///">]><comment><text>&xxe;</text></comment>

Antes de lanzar la petición, hay que localizar Content-Type: aplication/json y cambiarlo por aplicaton/xml**.** Tras este cambio, lanzamos la petición y completamos con éxito el ejercicio.



A6 Vuln & outdated Components – Apartado 5

Este ejercicio muestra como distintas versiones pueden hacer que un mismo componente sea explotable (primera parte) o no (segunda parte)



A7 Identity & Auth Failure – Secure Passwords – Apartado 4

En este ejercicio se aprende que la longitud, el uso de letras mayúsculas y minúsculas, números, y otro tipo de caracteres, ayudan a que las contraseñas sean mucho más seguras, y por tanto, difíciles de descifrar.