

ÁMBITO Y ALCANCE DE LA AUDITORÍA

La auditoría se ha realizado en Kali Linux, utilizando la herramienta Nmap, se han recuperado los siguientes datos sobre puertos, SO y lenguaje utilizado:

Escaneando la IP <http://127.0.0.1> con nmap lanza estos resultados:

```
(kali㉿kali)-[~]
└─$ nmap -p- 127.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-06 10:47 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00022s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
8080/tcp  open  http-proxy
9090/tcp  open  zeus-admin
42525/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 4.46 seconds
```

puertos abiertos: 8080, 9090 y 42525

```
(kali㉿kali)-[~]
└─$ sudo nmap -O 127.0.0.1
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-06 11:05 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00012s 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 1.59 seconds
```

la aplicación WebGoat está creada en java 21

```
(kali㉿kali)-[/home/kali]
└─PS> sudo docker run -it -p 127.0.0.1:8080:8080 -p 127.0.0.1:9090:9090 -e TZ=Europe/Berlin webgoat/webgoat
[sudo] password for kali:
2023-12-07T15:36:32.956+01:00 INFO 1 — [main] org.owasp.webgoat.server.StartWebGoat : Starting StartWebGoat v2023.8 using Java 21.0.1 with PID 1
me/webgoat/webgoat.jar started by webgoat in /home/webgoat)
```

La auditoría se ha realizado en Kali Linux, explotando las siguientes vulnerabilidades:

A3 Injection - SQL Injection (intro) - Apartado 10

WebGoat

127.0.0.1:8080/WebGoat/start.mvc?username=miguel#lesson/SqlInjection.lesson/9

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec webgoat

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(A6) Vuln & Outdated Components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

"SELECT * FROM user_data WHERE login_count = " + Login_Count + " AND userid = " + User_ID;

Using the two Input Fields below, try to retrieve all the data from the users table.

Warning: Only one of these fields is susceptible to SQL Injection. You need to find out which, to successfully retrieve all the data.

✓

Login_Count:
User_Id:

You have succeeded:

USERID, FIRST_NAME, LAST_NAME, CC_NUMBER, CC_TYPE, COOKIE, LOGIN_COUNT,
101, Joe, Snow, 987654321, VISA, 0,
101, Joe, Snow, 2234200065411, MC, 0,
102, John, Smith, 243560002222, MC, 0,
102, John, Smith, 4352209902222, AMEX, 0,
103, Jane, Plane, 123456789, MC, 0,
103, Jane, Plane, 333498703333, AMEX, 0,
10312, Jolly, Hershey, 176896789, MC, 0,
10312, Jolly, Hershey, 333300003333, AMEX, 0,
10323, Grumpy, youaretheweakestlink, 673834489, MC, 0,
10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, 0,
15603, Peter, Sand, 123609789, MC, 0,
15603, Peter, Sand, 338893453333, AMEX, 0,
15613, Joesph, Something, 33843453533, AMEX, 0,
15837, Chaos, Monkey, 32849386533, CM, 0,
19204, Mr, Goat, 33812953533, VISA, 0,

Your query was: SELECT * From user_data WHERE Login_Count = 3 and userid= 0 or 1 = 1

A3 Injection - SQL Injection (intro) - Apartado 11

127.0.0.1:8080/WebGoat/start.mvc?username=miguel#lesson/SqlInjection.lesson/10

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec webgoat

(A9) Security Logging Failures
(A10) Server-side Request Forgery
Client side
Challenges

own SQL after that.

It is your turn!

You are an employee named John **Smith** working for a big company. The company has an internal system that allows all employees to see data such as the department they work in and their salary.

The system requires the employees to use a unique *authentication TAN* to view their data.

Your current TAN is **3SL99A**.

Since you always have the urge to be the most highly paid employee, you want to exploit the system so that instead of viewing your own data to take a look at the data of all your colleagues to check their current salaries.

Use the form below and try to retrieve all employee data from the **employees** table. You should not need to know any specific names or TAN information you need.

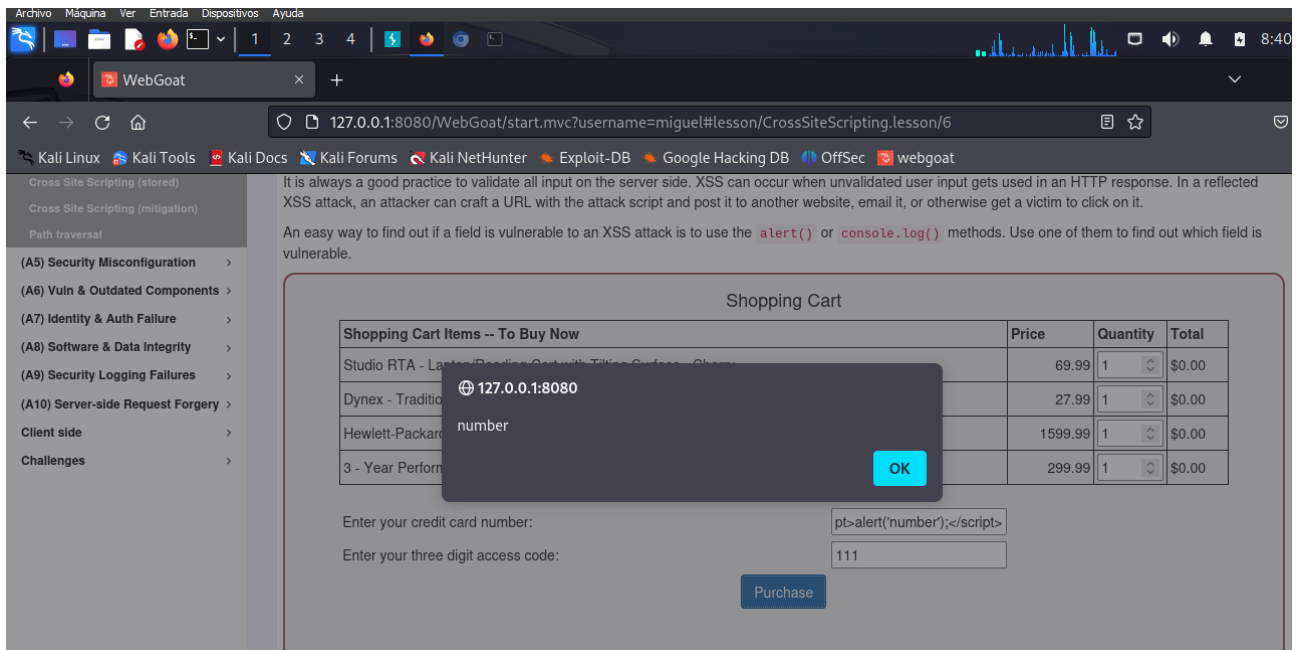
You already found out that the query performing your request looks like this:

"SELECT * FROM employees WHERE last_name = " + name + " AND auth_tan = " + auth_tan + "";

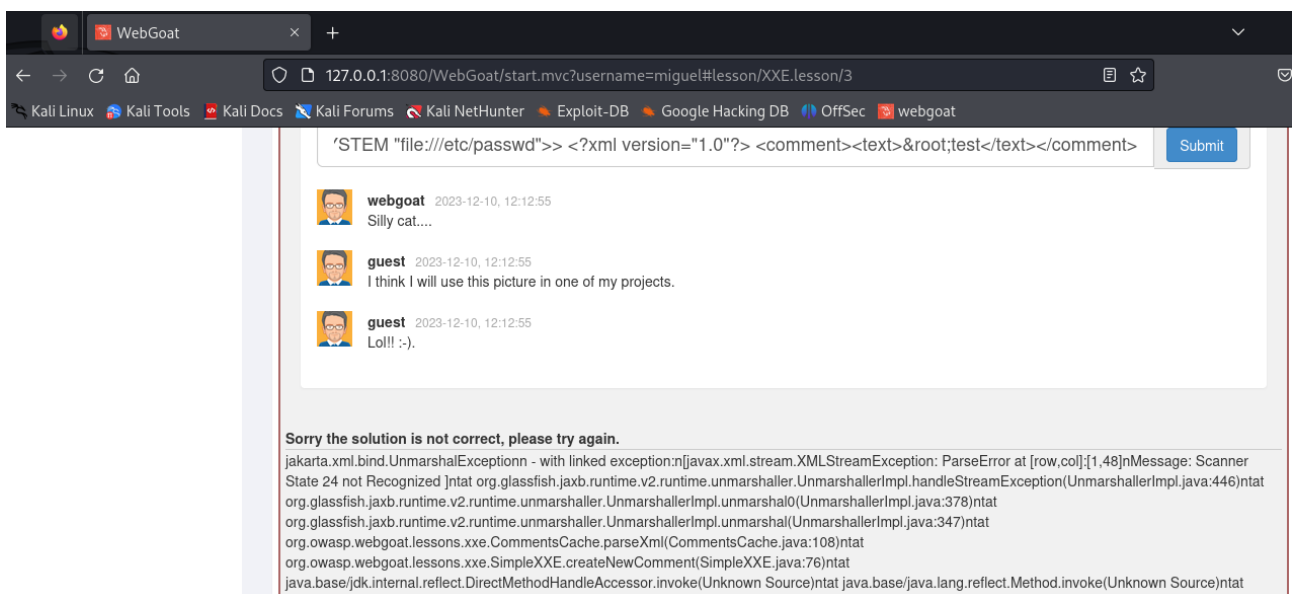
Employee Name:
Authentication TAN:

No employee found with matching last name. Or maybe your authentication TAN is incorrect?

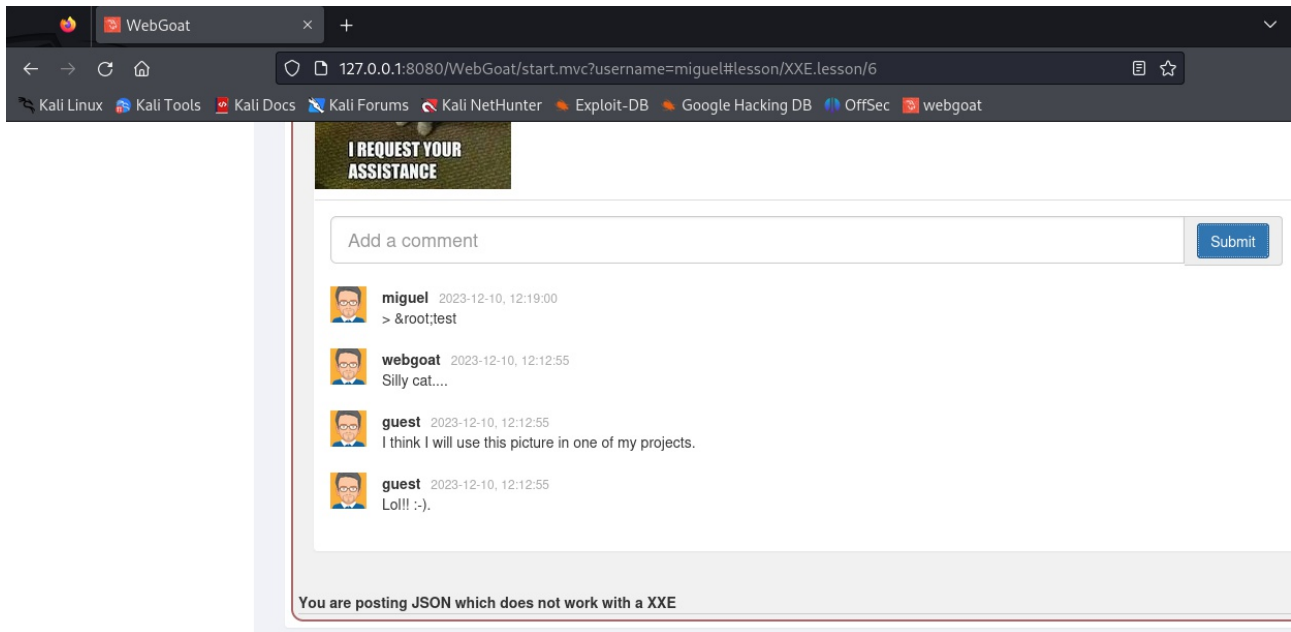
A3 Injection - Cross Site Scripting - Apartado - Apartado 7



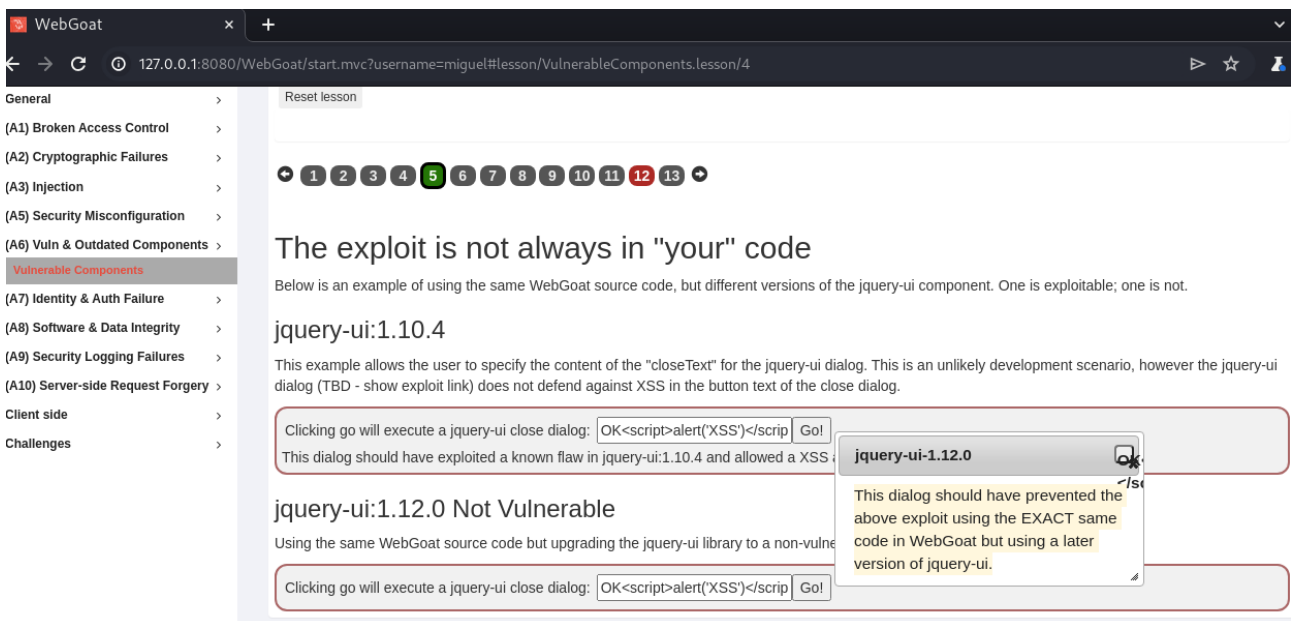
A5 Security Misconfiguration - Apartado 4



A5 Security Misconfiguration - Apartado 7



A6 Vuln & outdated Components - Apartado 5



A7 Identity & Auth Failure - Secure Passwords Apartado 4

WebGoat

traductor google - Buscar

127.0.0.1:8080/WebGoat/start.mvc?username=miguel#lesson/SecurePasswords.lesson/

Kali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSecwebgoat

(A1) Broken Access Control >

(A2) Cryptographic Failures >

(A3) Injection >

(A5) Security Misconfiguration >

(A6) Vuln & Outdated Components >

(A7) Identity & Auth Failure >

Authentication Bypasses

Insecure Login

JWT tokens

Password reset

Secure Passwords

(A8) Software & Data Integrity >

(A9) Security Logging Failures >

(A10) Server-side Request Forgery >

Client side >

Challenges >

123456

How long could it take to brute force your password?

In this assignment, you have to type in a password that is strong enough (at least 4/4).

After you finish this assignment we highly recommend you try some passwords below to see why they are not good choices:

- password
- johnsmith
- 2018/10/4
- 1992home
- abcbac
- ffffget
- poiuz
- @dmin

Enter a secure password...

Submit

You have succeeded! The password is secure enough.

Your Password: *****

Length: 13

Estimated guesses needed to crack your password: 15100000000

Score: 4/4

Estimated cracking time: 47 years 321 days 20 hours 26 minutes 40 seconds

Score: 4/4