

# WEB2S SECURITY

Reporte de pentesting y comprobación de vulnerabilidades, utilizando  
Metaexploitable2 y Kali Linux.

Web2S Security

Febrero 2024

versión 1.0

Realizado por: Miguel Ángel Fernández Parejo

[mafparejo@proton.me](mailto:mafparejo@proton.me)

[mdr716@gmail.com](mailto:mdr716@gmail.com)

[mafparejo@gmail.com](mailto:mafparejo@gmail.com)

## CONTENIDO

- 1 Autoría del documento y derechos de copyright.
- 2 Ámbito y alcance de la auditoría.
- 3 Resumen.

## 1 AUTORÍA DEL DOCUMENTO Y DERECHOS DE COPYRIGHT

El presente documento ha sido realizado por Web2S Security, y está protegido con derechos de autor *copyright*, está prohibida su copia parcial o totalmente, así como su distribución fuera del ámbito de Web2S Security y de su cliente Oblibion, S.A. Contiene información sensible estando prohibida su divulgación por cualquier medio digital o físico.

CLIENTE: Oblibion, S.A.

PROYECTO: Seguridad y pentesting.

CLASIFICACIÓN: Confidencial

AUTOR: Miguel Ángel Fernández Parejo

## 2 ÁMBITO Y ALCANCE DE LA AUDITORÍA

La auditoría se ha realizado en Metaexploitable2 y 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 21.0.1

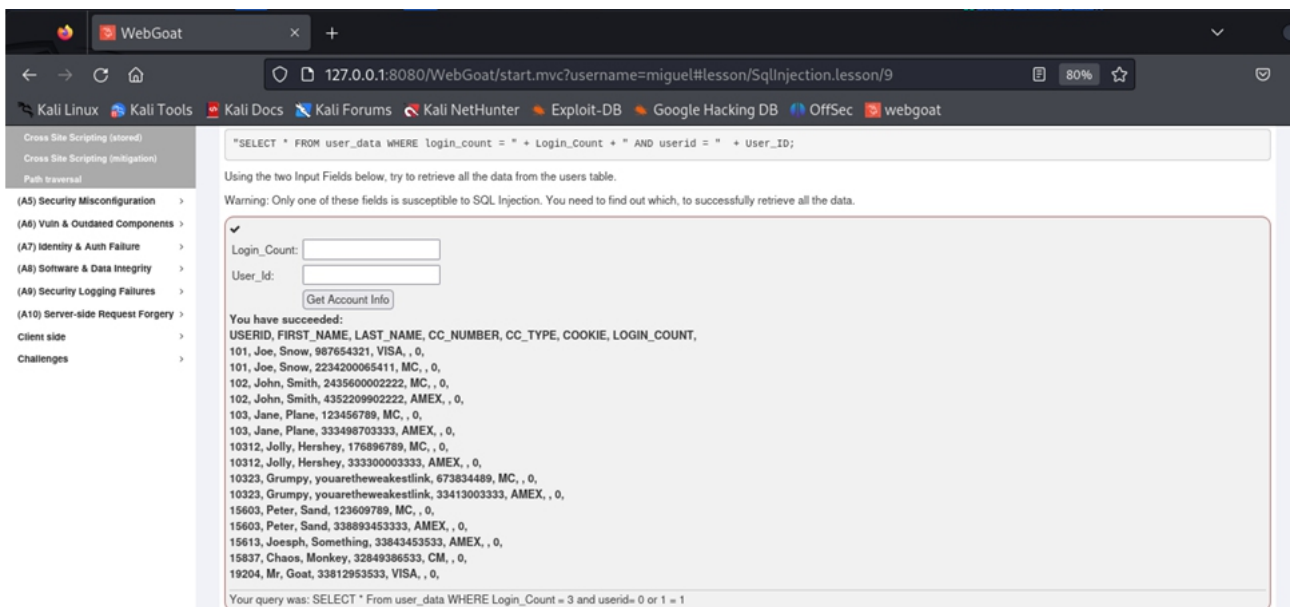
```
(kali@kali)~/home/kali
PS> sudo docker run -it -p 127.0.0.1:8080:8080 -p 127.0.0.1:9090:9090 -e TZ=Europe/Berlin webgoat/webgoat
[sudo] password for kali:
2023-12-07T15:36:32.956+01:00 INFO 1 --- [ main] org.owasp.webgoat.server.StartWebGoat : Starting StartWebGoat v2023.8 using Java 21.0.1 with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

La auditoría ha encontrado las siguientes vulnerabilidades:

## A3 Injection - SQL Injection (intro) - Apartado 10

Esta se basa en la inyección de código SQL, es decir, introducir caracteres de este lenguaje de consulta, para cada petición al servidor, que permite obtener datos como usuarios y sus tarjetas de crédito, incluso podría descargar toda la base de datos:

ej.: or 1 = 1



Recomendación para posible mitigación.

Posibles soluciones:

Uso de API segura, con parametrización, así como validaciones en el lado del servidor.

Uso de LIMIT y controles SQL en las consultas para evitar descubrir

datos.

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

En esta ocasión, mediante el campo Employee Name, podemos obtener información confidencial de usuarios y también podemos descargar la base de datos:

The screenshot shows a web browser window with the URL `127.0.0.1:8080/WebGoat/start.mvc?username=miguel#lesson/SqlInjection.lesson/10`. The browser's address bar and tabs are visible. The page content includes a sidebar with navigation links: (A9) Security Logging Failures, (A10) Server-side Request Forgery, Client side, and Challenges. The main content area is titled "It is your turn!" and contains a narrative about an employee named John Smith. It explains that the system requires an authentication TAN to view data and that the user's current TAN is 3SL99A. The challenge is to retrieve all employee data from the `employees` table. A code block shows the SQL query: `"SELECT * FROM employees WHERE last_name = '' + name + '' AND auth_tan = '' + auth_tan + ''";`. Below the code, there is a form with two input fields: "Employee Name:" containing `M employees WHERE 1=1` and "Authentication TAN:" containing `3SL99A`. A "Get department" button is also present. At the bottom, a message states: "No employee found with matching last name. Or maybe your authentication TAN is incorrect?"

Recomendación para posible mitigación.

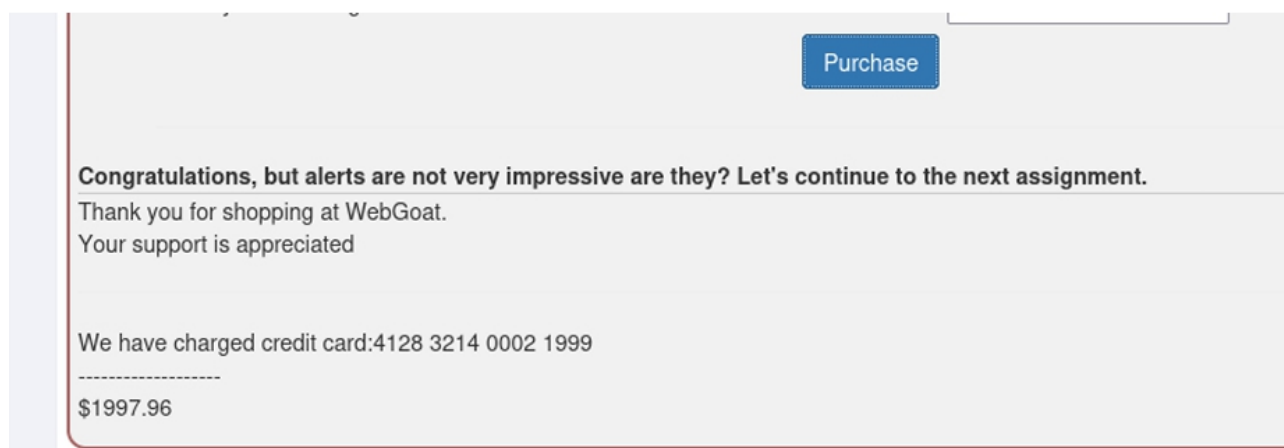
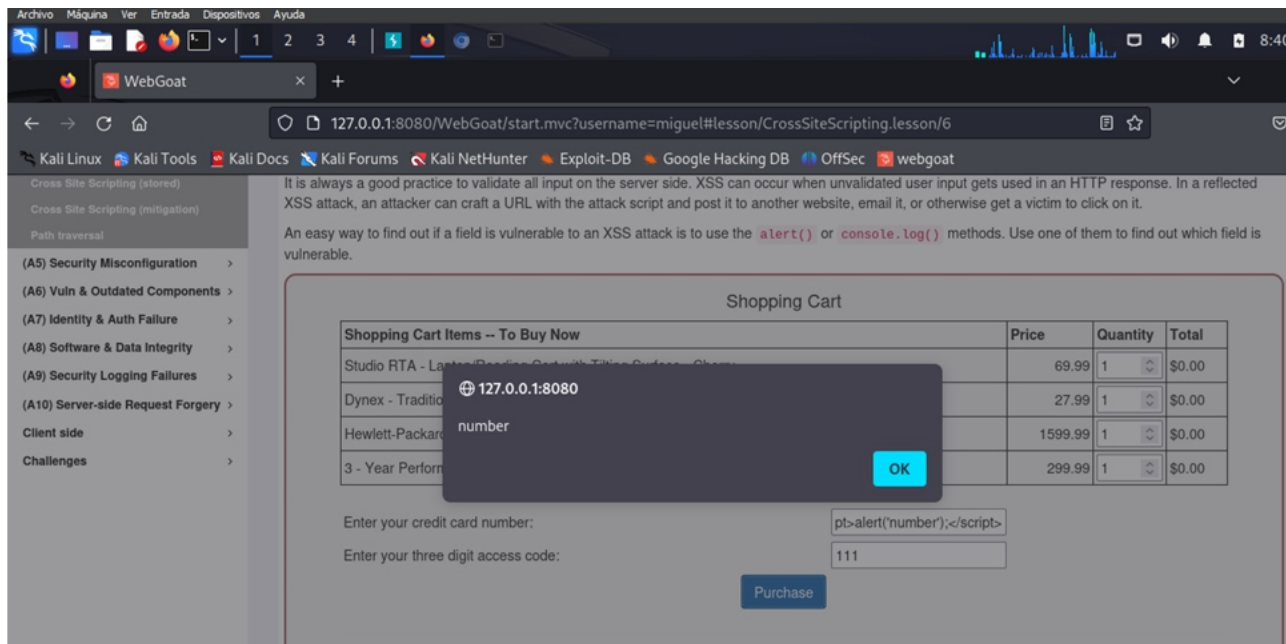
Posibles soluciones:

Uso de API segura, con parametrización, así como validaciones en el lado del servidor.

Uso de LIMIT y controles SQL en las consultas para evitar descubrir datos.

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

Se ha podido encontrar esta vulnerabilidad, que se basa en la inyección de código JavaScript. Se ha introducido en el campo de entrada “Enter your credit card number”:



Recomendación para posible mitigación.

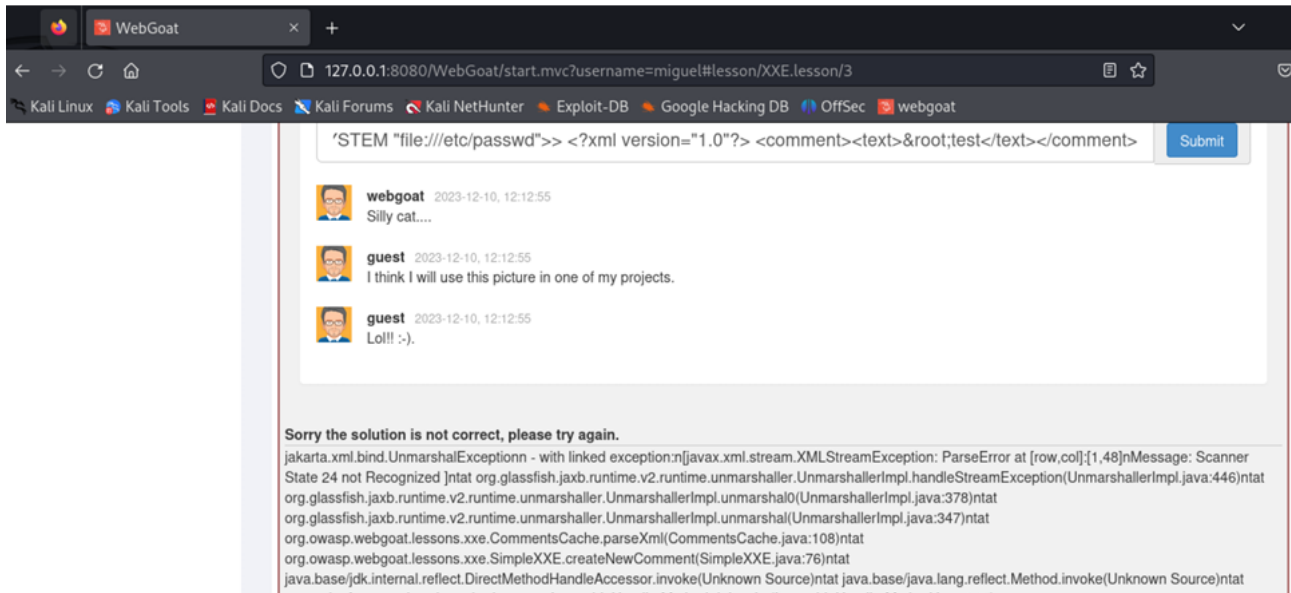
Posibles soluciones:

Antivirus como Microsoft Defender.

Extensiones *no-script* en el navegador.

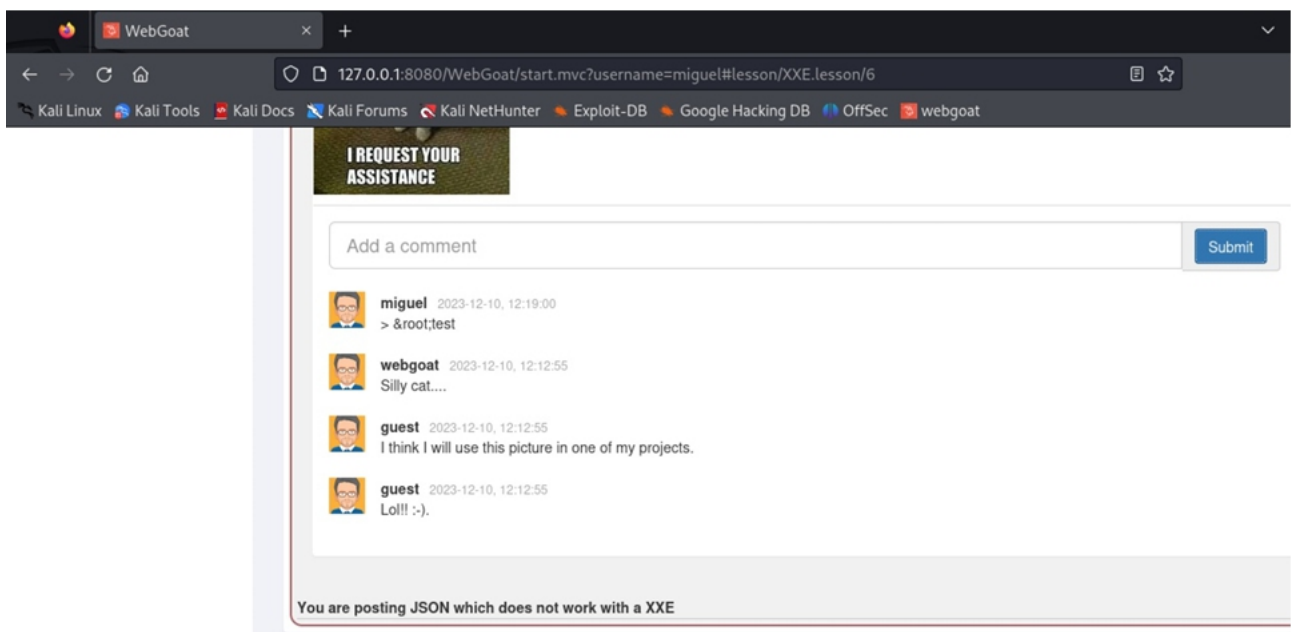
Uso de firewall de aplicaciones web (WAF), protege de los ataques XSS bloqueando las peticiones maliciosas.

## A5 Security Misconfiguration - Apartado 4



Esta trata de la falta de seguridad adecuada en cualquier parte de la aplicación o permisos configurados incorrectamente:

## A5 Security Misconfiguration - Apartado 7



Falta seguridad en aplicaciones, permisos mal configurados en

aplicaciones web, credenciales débiles, fáciles de conseguir. Puertos abiertos, los servidores tienen malas configuraciones de seguridad.

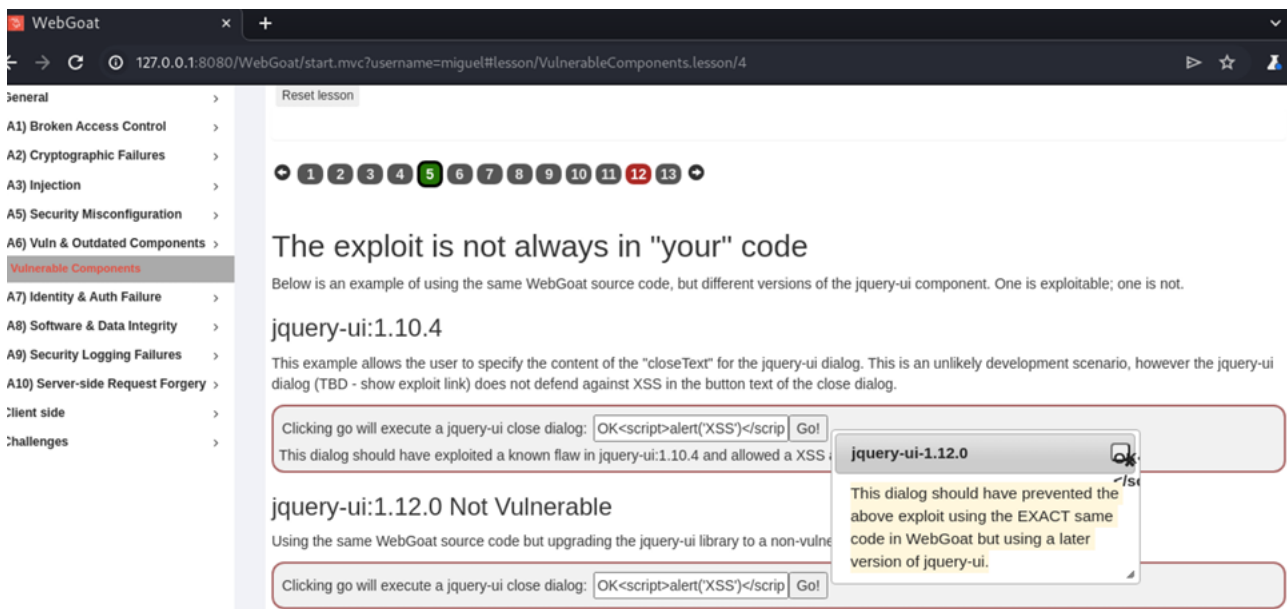
Recomendación para posible mitigación.

Posibles soluciones:

Se recomienda revisar la configuración de seguridad de la aplicación y de servidores, actualización frecuente de credenciales alfanuméricas.

## A6 Vuln & outdated Components - Apartado 5

Vulnerabilidad basada en software desactualizado; versiones obsoletas y sin posibilidad de soporte, de tal manera que no puedan ser actualizadas con parches de seguridad:



WebGoat

127.0.0.1:8080/WebGoat/start.mvc?username=miguel#lesson/VulnerableComponents.lesson/4

General

- A1) Broken Access Control
- A2) Cryptographic Failures
- A3) Injection
- A5) Security Misconfiguration
- A6) Vuln & Outdated Components
- Vulnerable Components
- A7) Identity & Auth Failure
- A8) Software & Data Integrity
- A9) Security Logging Failures
- A10) Server-side Request Forgery
- Client side
- Challenges

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. One is exploitable; one is not.

#### jquery-ui:1.10.4

This example allows the user to specify the content of the "closeText" for the jquery-ui dialog. This is an unlikely development scenario, however the jquery-ui dialog (TBD - show exploit link) does not defend against XSS in the button text of the close dialog.

Clicking go will execute a jquery-ui close dialog: OK<script>alert('XSS')</script> Go!

This dialog should have exploited a known flaw in jquery-ui:1.10.4 and allowed a XSS i

#### jquery-ui:1.12.0 Not Vulnerable

Using the same WebGoat source code but upgrading the jquery-ui library to a non-vulne

Clicking go will execute a jquery-ui close dialog: OK<script>alert('XSS')</script> Go!

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.

Recomendación para posible mitigación.

Posibles soluciones:

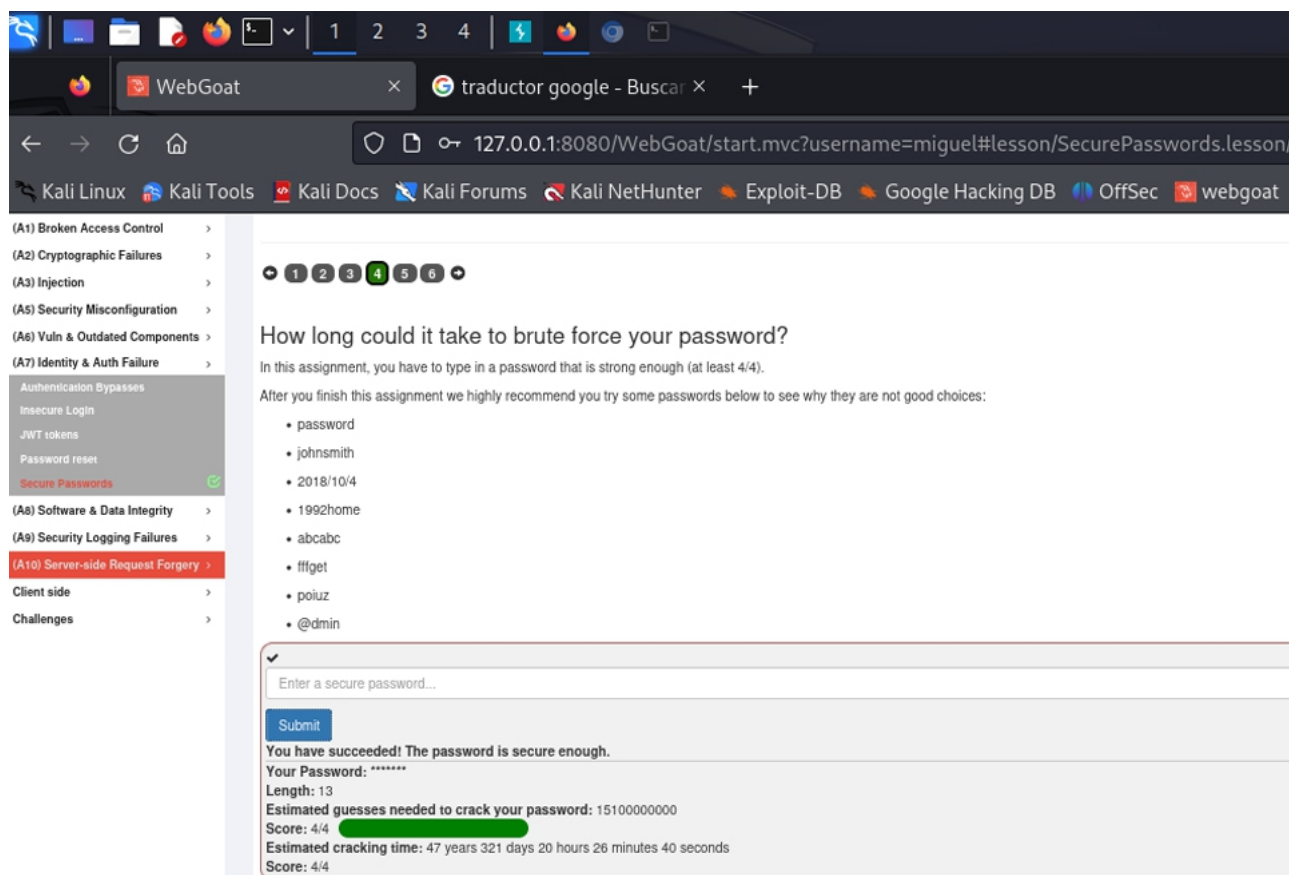


Actualización del Sistema Operativo, actualización de antivirus o EDR, si el sistema operativo carece de soporte, habría que renovar a uno que sí tenga.

También se recomienda revisar la configuración de la aplicación.

## A7 Identity & Auth Failure - Secure Passwords Apartado 4

Fallos en la parte de autenticación e identificación de la aplicación, ofreciendo un ataque por ejemplo, de fuerza bruta:



Recomendación para posible mitigación.

Posibles soluciones:

Se recomienda introducir framework como Spring Security [Spring Security](#)



(lenguaje Java) para aumentar la seguridad en la autenticación e identificación de usuarios de la aplicación, ORM 's como Hibernate [Hibernate. Everything data](#) o MyBatis [MyBatis \(github.com\)](#) para el control de las consultas SQL.

### 3 RESUMEN

Se han encontrado numerosos fallos de seguridad o vulnerabilidades de varios tipos como: inyección SQL, inyección de XSS (código javascript), fallos de software obsoleto o desactualizado sin posibilidad de soporte, fallos de configuración de permisos, así como de autenticación e identificación.