Auditoria Web Goat

De

Daniel Shved

OBJETIVOS:

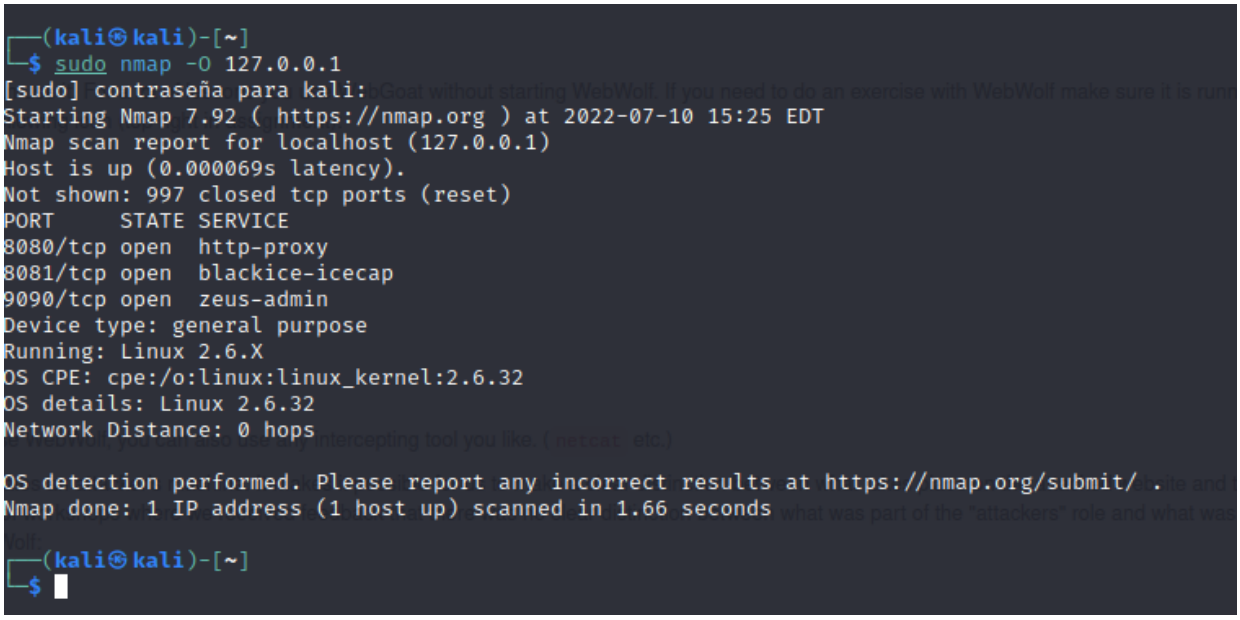
Simulación de auditoria de web goat.

HERRAMIENTAS USADAS:

- WAPPALYZER: Tecnologia y lenguajes de programación utilizados en la aplicación web
- NMAP: Sistema Operativo y puertos abiertos
- OWASP ZAP: Interceptar paquetes para resolver ejercicios.
- MODO DESARROLLADOR de Firefox: Ver como esta escrita la pagina para resolver ejercicios.

INFORMATIÓN GATHERING:

El servidor en el que corre webgoat utiliza Linux como sistema operativo y tiene abiertos los puertos tcp 8080, 8081 y 9090. Hemos sacado esta información utilizando el programa NMAP



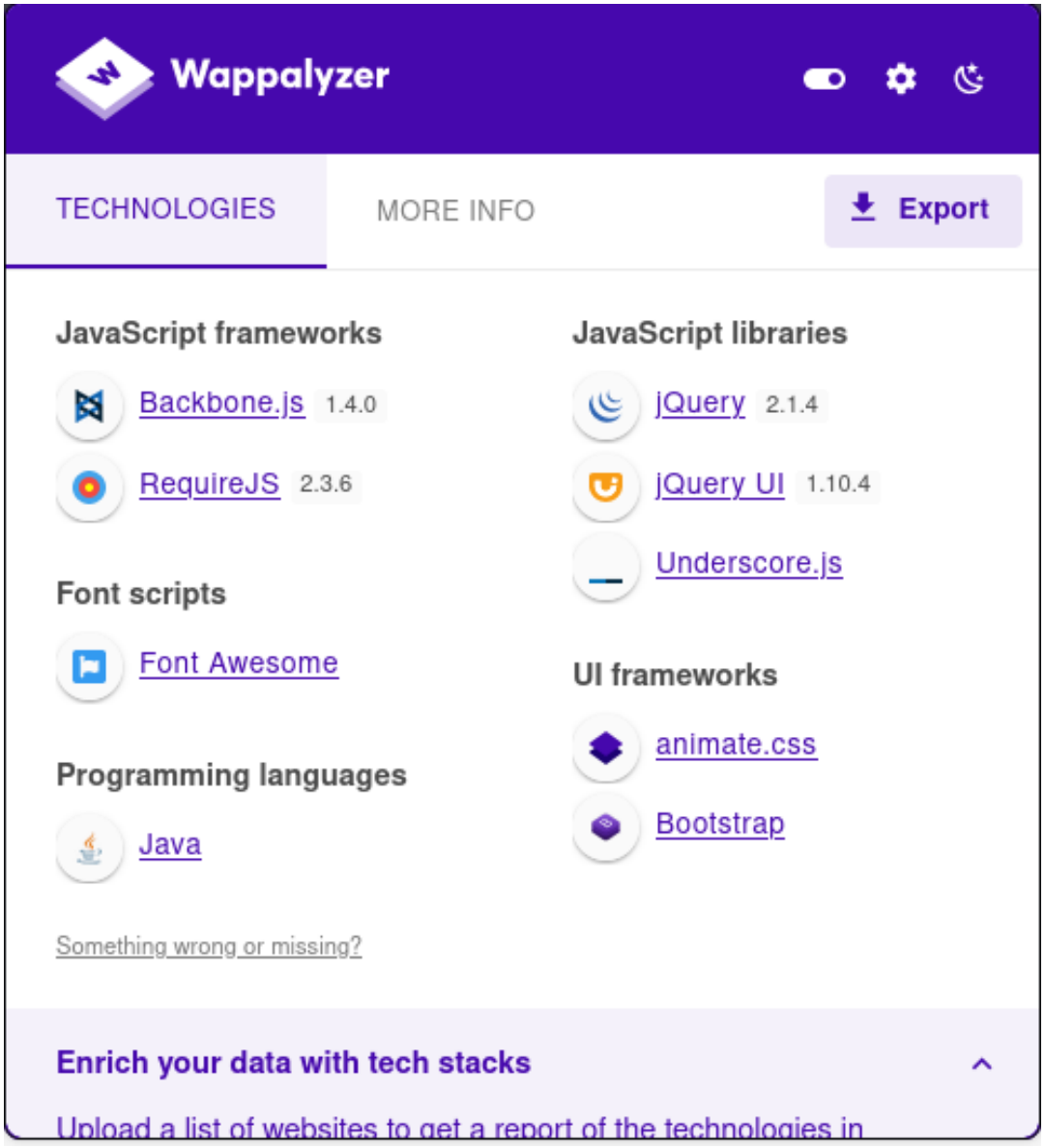
Utiliza los javascript frameworks: Backbone.is 1.4.0, ReguireJS 2.3.6 .

Librerías javascript: jQuery 2.1.3. , jQuery UI 1.10.4, Undersocre.js .

Font Scipts: Font Awesome

Ui frameworks: anímate.css y Bootstrap

Lenguaje de Programacion: Java



FALLOS DISPONIBLES EN LA SECCIÓN A1 SQL INJECTION:

Tras aprender SQL básico http://www.sqlcourse.com/ utilizando el siguiente formato se puede sacar información del servidor usando el comando select:

The **select** statement is used to query the database and retrieve selected data that match the criteria that you specify. Here is the format of a simple select statement:

```
select "column1"
  [,"column2",etc]
  from "tablename"
  [where "condition"];
  [] = optional
```

Resolucion del ejercicio A1(2):

Employees Table

userid	first_name	last_name	department	salary	auth_tan
32147	Paulina	Travers	Accounting	$46.000	P45JSI
89762	Tobi	Barnett	Development	$77.000	TA9LL1
96134	Bob	Franco	Marketing	$83.700	LO9S2V
34477	Abraham	Holman	Development	$50.000	UU2ALK
37648	John	Smith	Marketing	$64.350	3SL99A

A company saves the following employee information in their databases: a unique employee number ('userid'), last name, first name, department, salary and a transaction authentication number ('auth_tan'). Each of these pieces of information is stored in a separate column and each row represents one employee of the company.

SQL queries can be used to modify a database table and its index structures and add, update and delete rows of data.

There are three main categories of SQL commands:

- Data Manipulation Language (DML)
- Data Definition Language (DDL)
- Data Control Language (DCL)

Each of these command types can be used by attackers to compromise the confidentiality, integrity, and/or availability of a system. Proceed with the lesson to learn more about the SQL command types and how they relate to protections goals.

If you are still struggling with SQL and need more information or practice, you can visit http://www.sqlcourse.com/ for free and interactive online training.

It is your turn!

Look at the example table. Try to retrieve the department of the employee Bob Franco. Note that you have been granted full administrator privileges in this assignment and can access all data without authentication.

SQL query: SQL query

Submit

You have succeeded!

SELECT department FROM Employees WHERE userid = 96134

DEPARTMENT

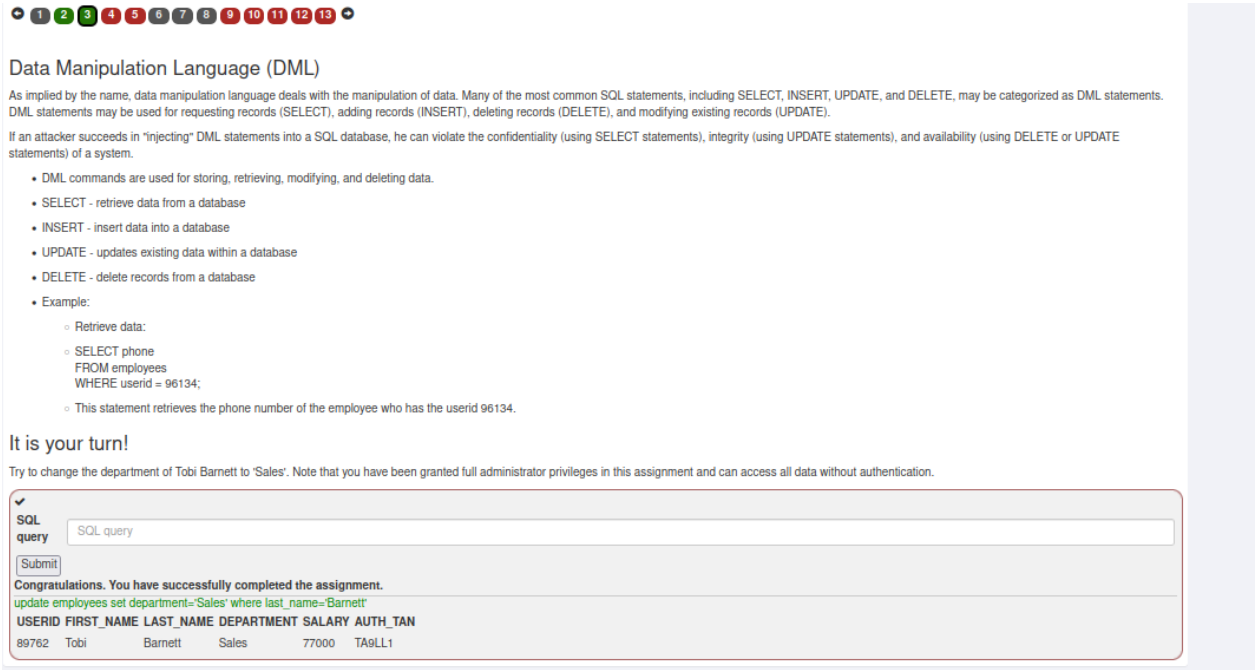
Marketing

Usando este formato a base de prueba y error se puede sobre escribir información en SQL:

The **update** statement is used to update or change records that match a specified criteria. This is accomplished by carefully constructing a where clause.

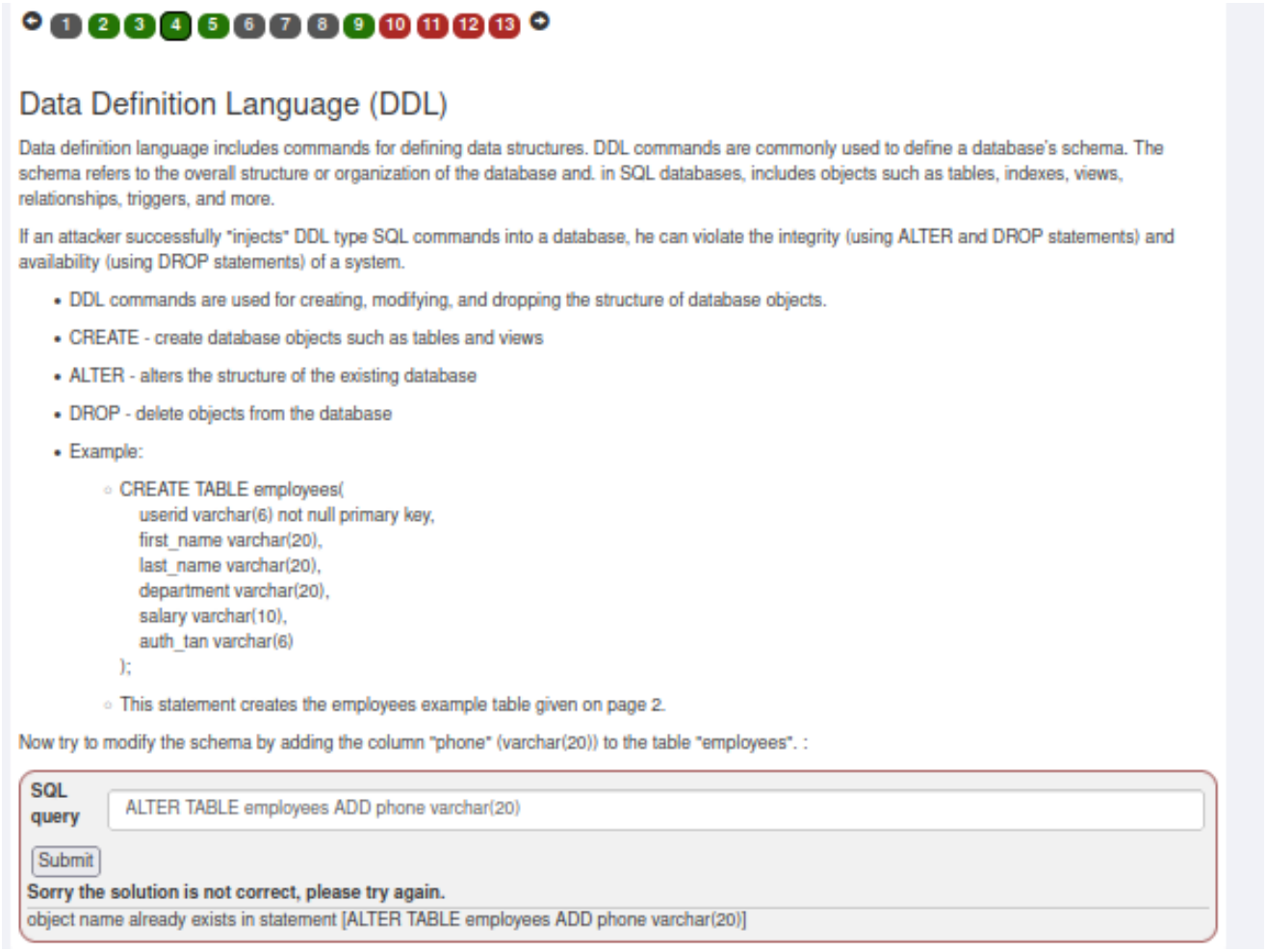
```
 update "tablename"
set "columnname" =
    "newvalue"
 [,"nextcolumn" =
   "newvalue2"...]
where "columnname"
  OPERATOR "value"
 [and|or "column"
  OPERATOR "value"];

 [] = optional
```

Resolucion ejercicio A1(3)



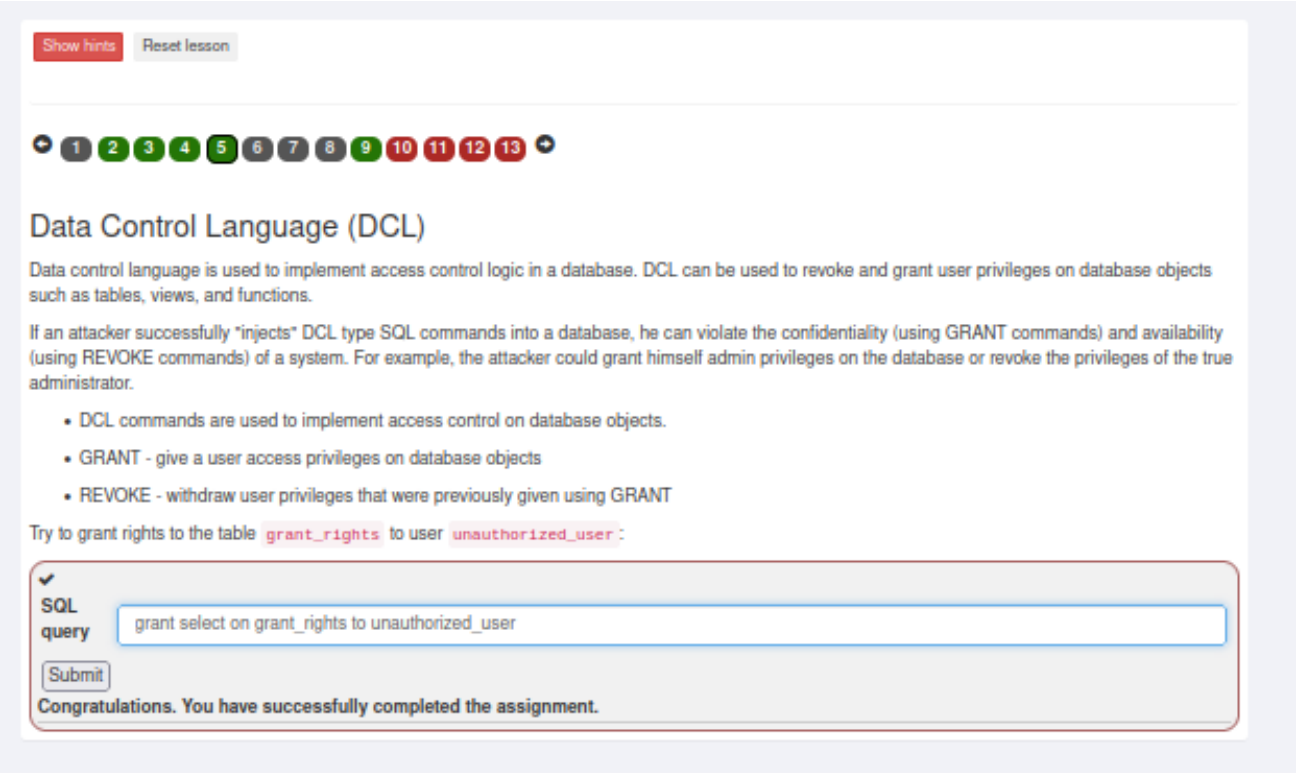
Usando el comando ALTER TABLE se puede alterar la base de datos.
Varchar(20) definde la longitud del campo añadido a la tabla.

Resolucion ejercicio A1(9)



Usando el comando GRANT se le puede dar permisos de administrador a usuario:

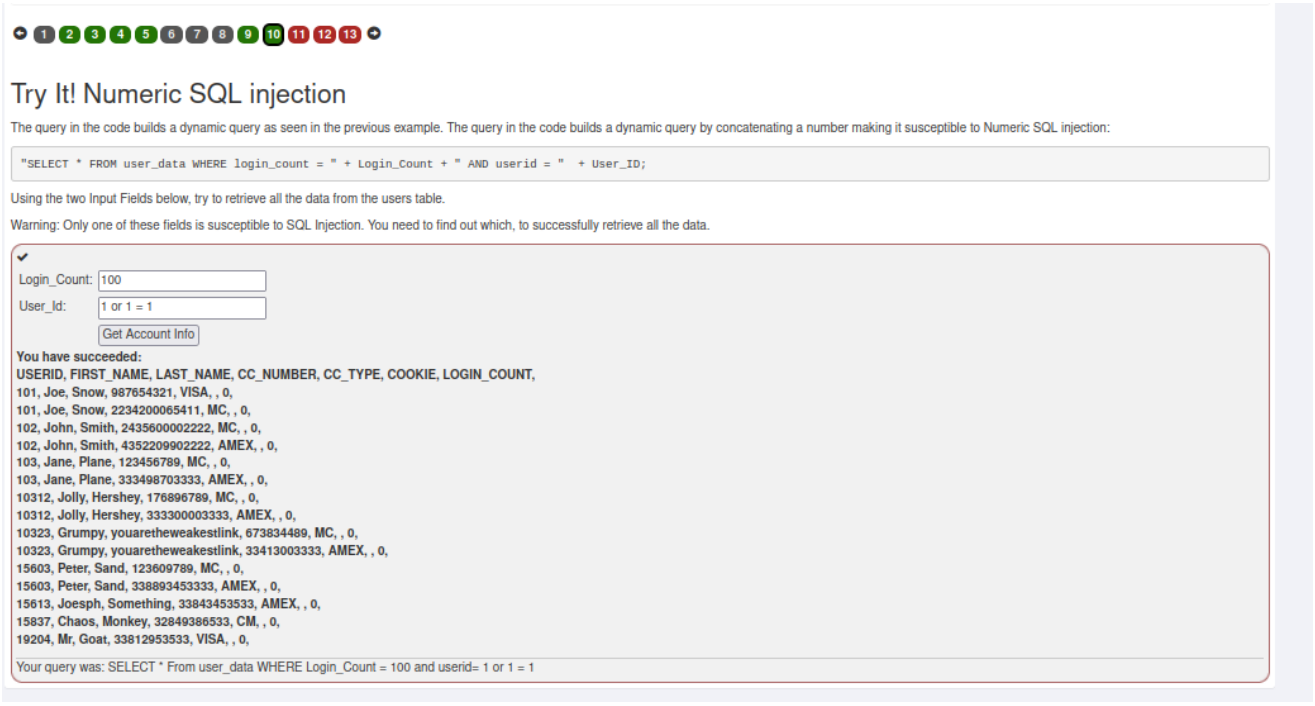
Resolucion problema A1(5):



A1 SQL INJECTION - APARTADO 10

El código lo que dice es selecciona todo de la tabla “user_data” con los datos login count y User_ID, excepto que si ponemos 1 or 1 = 1 (Boleano True) elimina la comprobación del Login_Count y el User_ID pues solo una de las condiciones de siente que cumplir; o cuadra el Login_Count y el User_Id con la info del servidor o 1 = 1.

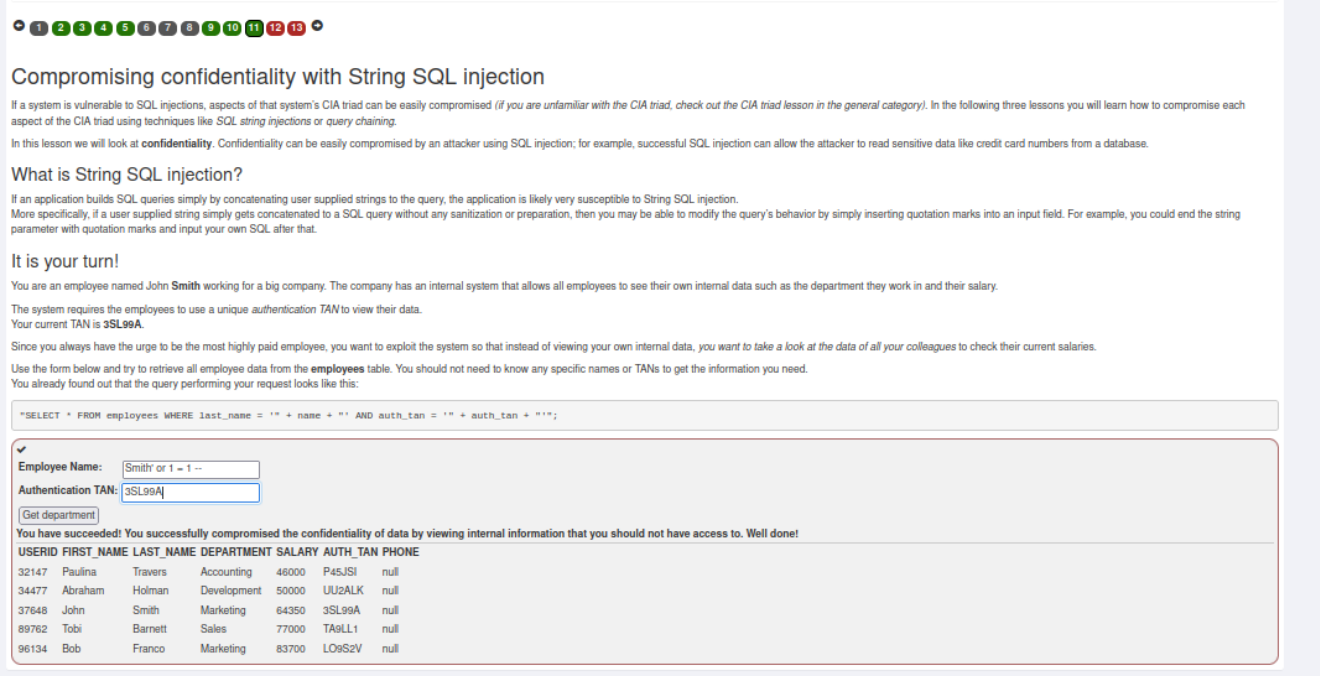
Para aclarar es irellevante lo que pongamos en Login_Count por el “or 1 = 1 ” que es true.



A1 SQL INJECTION - APARTADO 11

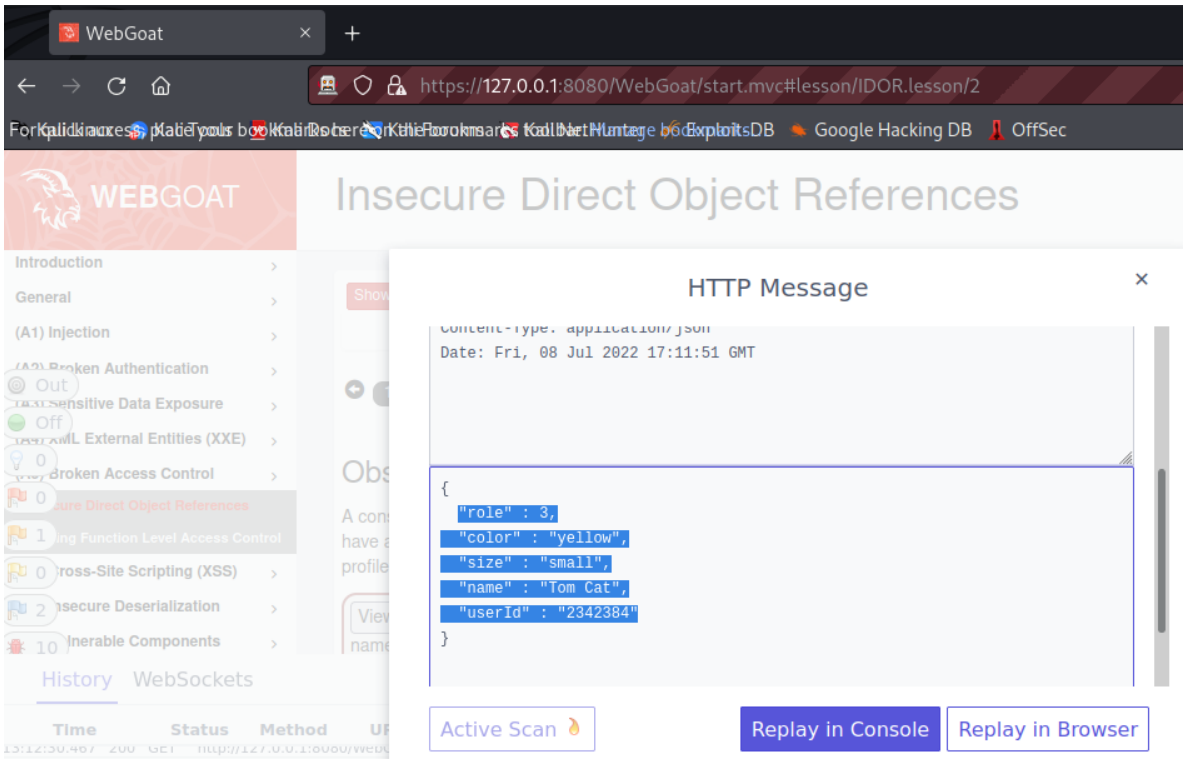
Usamos SQL Injecction para sacar datos de otros usuarios. De nuevo lo que pongamos en Authenthification TAN es irrelevante al igual que antes lo era el Login_Count. En este caso es por que el código Smith" or 1 = 1 – hace que el Authenthification TAN sea texto.

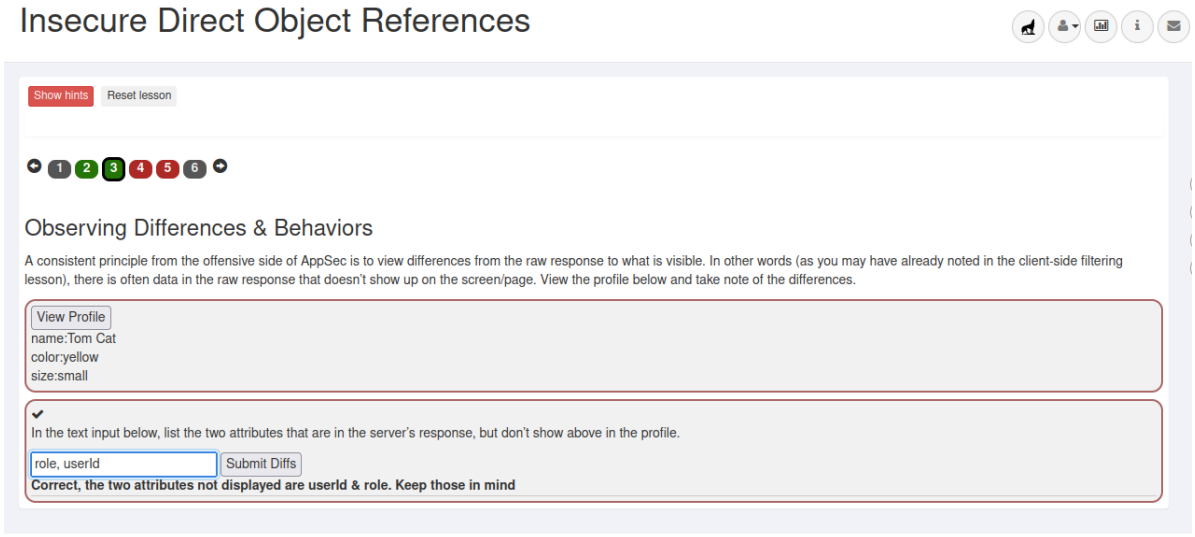
Se cumple 1 = 1 ergo te doy todos los datos que cuadren con todo.



A5 INSECURE DIRECT OBJECT REFERENCES - APARTADO 3

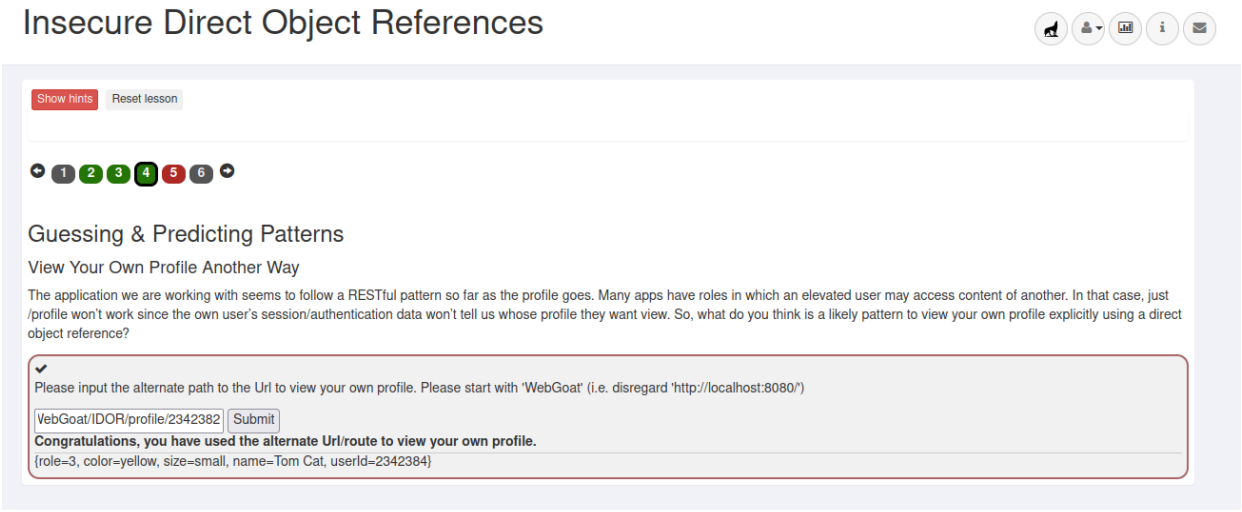
Filtrando los paquetes con owasp zap podemos ver que la respues del servidor mas de lo que presenta la pagina:





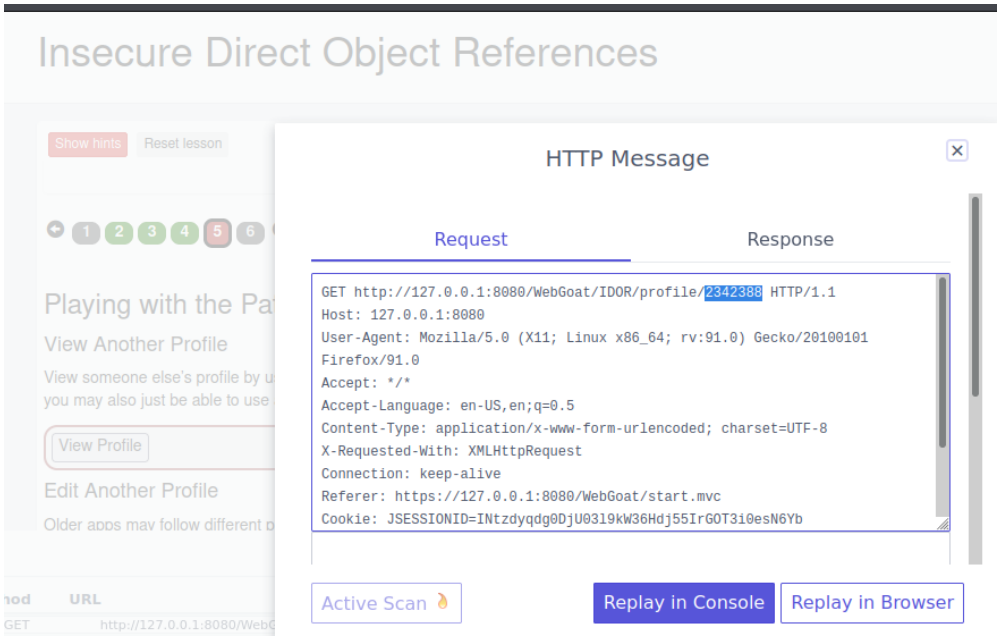
A5 INSECURE DIRECT OBJECT REFERENCES - APARTADO 4

Usando el formato de la petición (que nos da el IDOR/profile) y los datos del ejercicio anterior(Numero de id) podemos a base de prueba y error sacar :

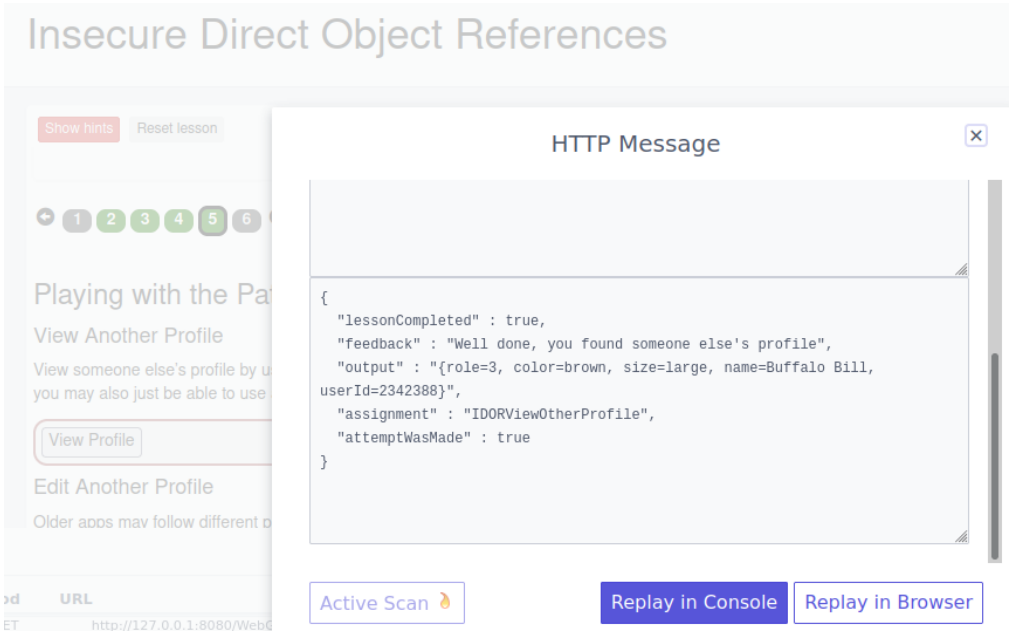


A5 INSECURE DIRECT OBJECT REFERENCES - APARTADO 5

Podemos ver información de otro usuario utilizando la dirección que obtuvimos anteriormente y a base de prueba y error el numero de ID. Creo recordar que cambie a Get en vez de un post también. Me queda por aprender como hacer fuzzing con owasp, el numero de ID lo saque probano por que en las pistas poníaque estaba cerca del mio. Tambien la segunda parte del ejercicio fui incapaz de hacerla.

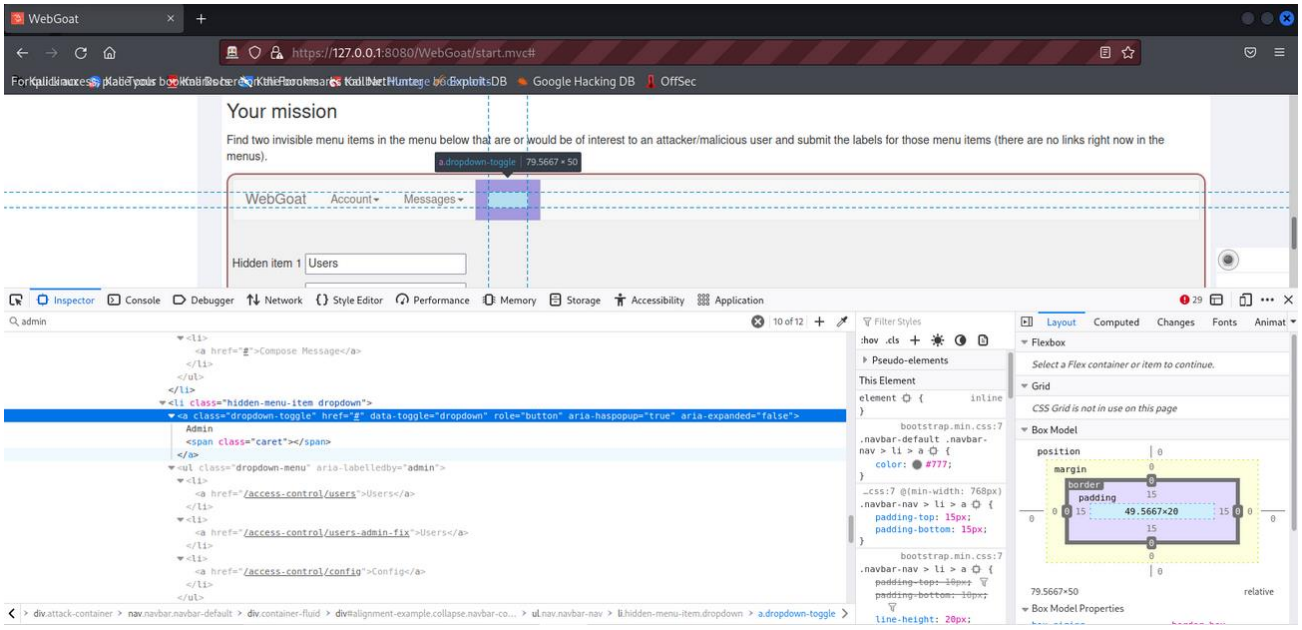


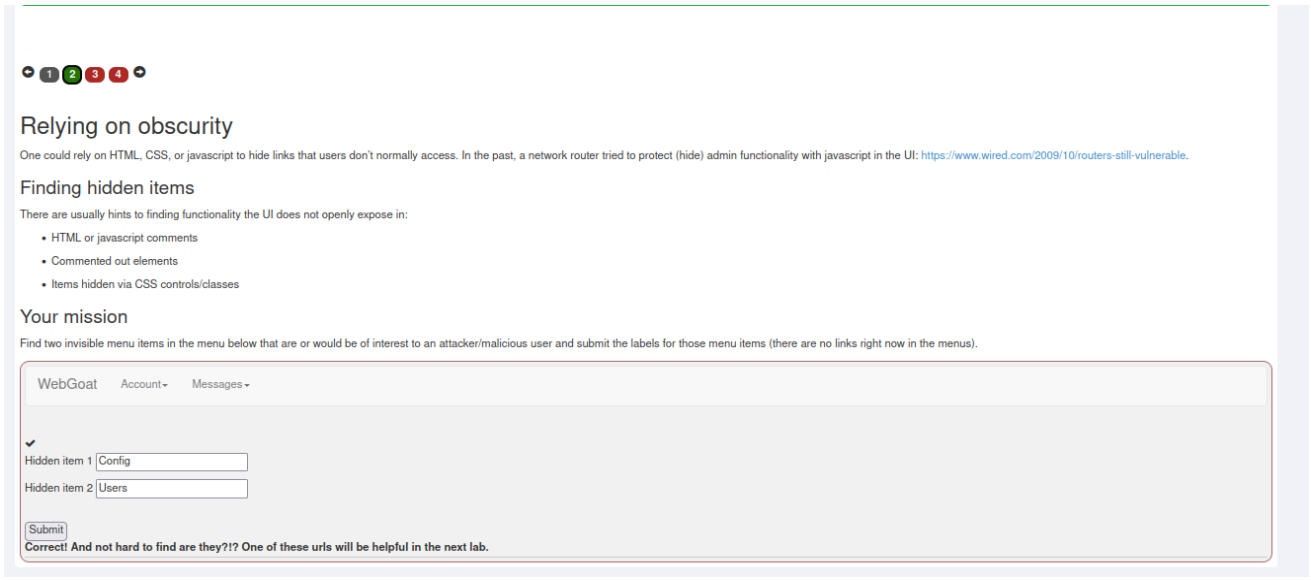
Respuesta con datos de otro Usuario.



A5 MISSING FUNCTION LEVEL ACCESS CONTROL - APARTADO 2

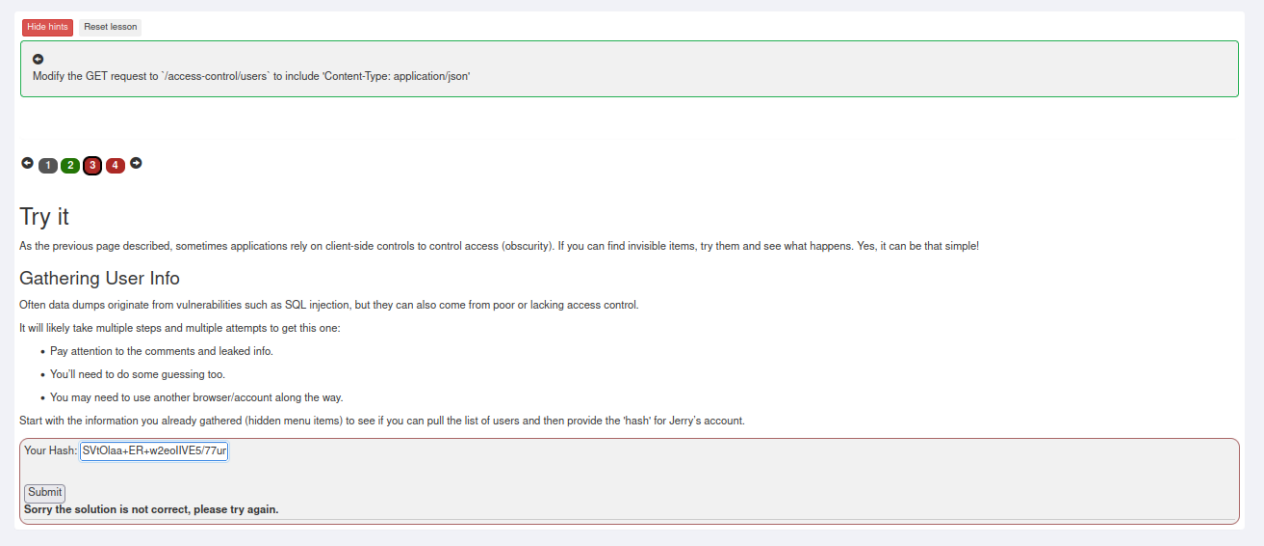
En la programación de la pagina tras mucho mucho tiempo, usando búsquedas tipo admin, pass y hidden y utilizando la función de Firefox de señalarte en que parte del código esta a lo que apuntas con el cursor encontré dos elementos ocultos en la pagina, posiblemente para ser usados o que se muestran si entras como administrador:

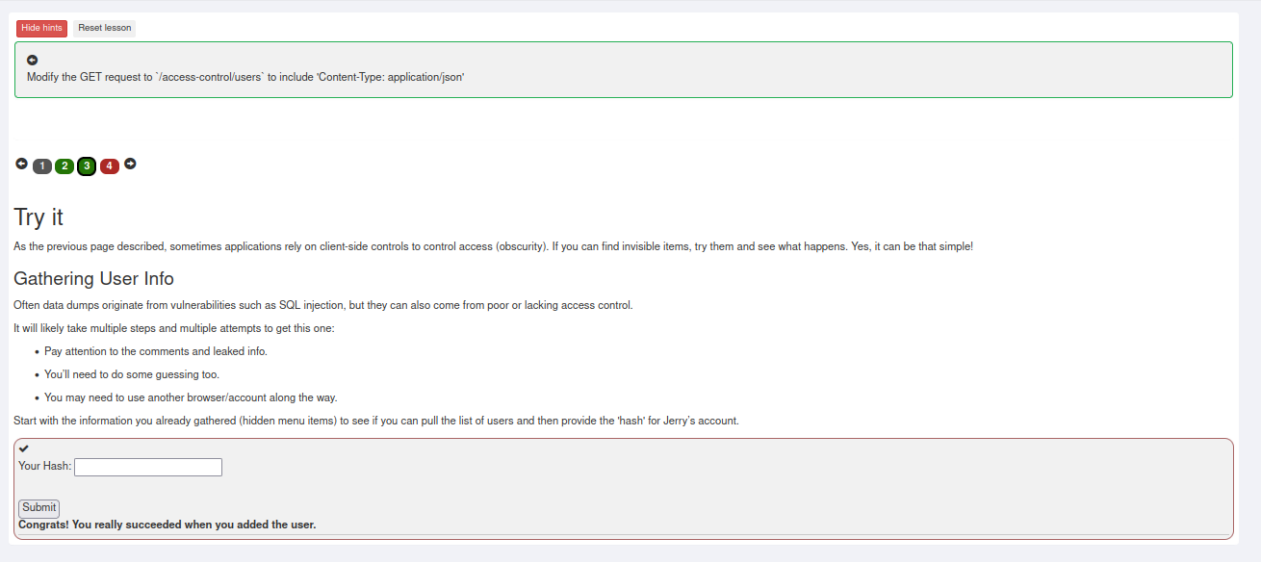




A5 MISSING FUNCTION LEVEL ACCESS CONTROL - APARTADO 3

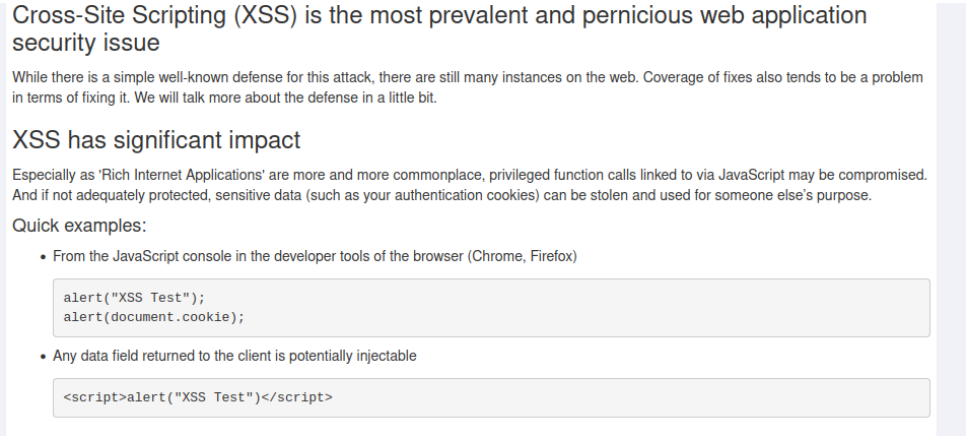
Modificando la petición anterior con Owasp Zap a Get (era un POST) y modificando el Acces Control y El Content Type puedes sacar el Hash de otros usuarios.





A7 CROSS SITE SCRIPTING - APARTADO - APARTADO 7

Usando estos comandos:



Cross-Site Scripting (XSS) is the most prevalent and pernicious web application security issue

While there is a simple well-known defense for this attack, there are still many instances on the web. Coverage of fixes also tends to be a problem in terms of fixing it. We will talk more about the defense in a little bit.

XSS has significant impact

Especially as 'Rich Internet Applications' are more and more commonplace, privileged function calls linked to via JavaScript may be compromised. And if not adequately protected, sensitive data (such as your authentication cookies) can be stolen and used for someone else's purpose.

Quick examples:

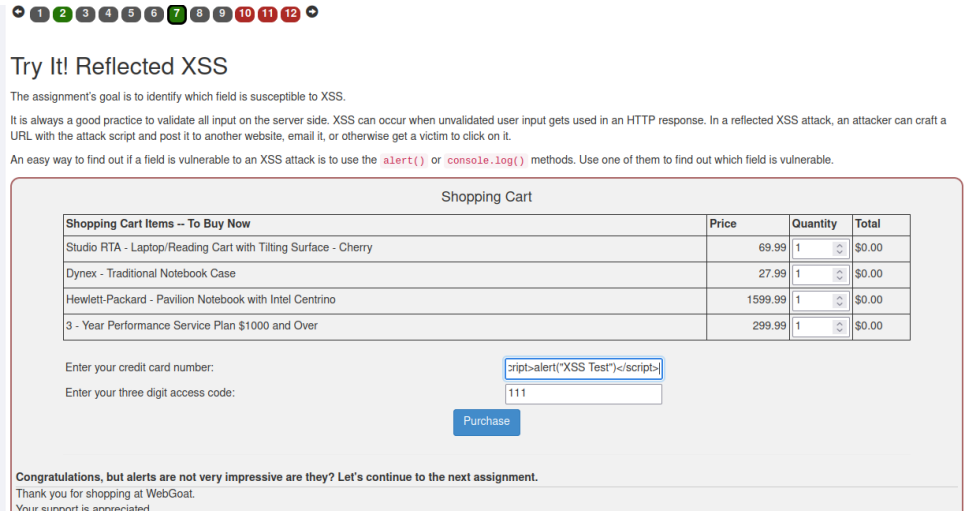
- From the JavaScript console in the developer tools of the browser (Chrome, Firefox)

```
alert("XSS Test");
alert(document.cookie);
```

- Any data field returned to the client is potentially injectable

```
<script>alert("XSS Test")</script>
```

Podemos averiguar que este campo es vulnerable:



1 2 3 4 5 6 7 8 9 10 11 12

Try It! Reflected XSS

The assignment's goal is to identify which field is susceptible to XSS.

It is always a good practice to validate all input on the server side. XSS can occur when unvalidated user input gets used in an HTTP response. In a reflected XSS attack, an attacker can craft a URL with the attack script and post it to another website, email it, or otherwise get a victim to click on it.

An easy way to find out if a field is vulnerable to an XSS attack is to use the `alert()` or `console.log()` methods. Use one of them to find out which field is vulnerable.

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	1	$0.00
Dynex - Traditional Notebook Case	27.99	1	$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	1	$0.00
3 - Year Performance Service Plan $1000 and Over	299.99	1	$0.00

Enter your credit card number: ;ript>alert("XSS Test")</script>|

Enter your three digit access code: 111

Purchase

Congratulations, but alerts are not very impressive are they? Let's continue to the next assignment.

Thank you for shopping at WebGoat.

Your support is appreciated

RESUMEN:

En resumen de cara a al ejercicio esta pagina es vulnerable a SQL Injection , tiene Insecure Direct Object References y es vulnerable a Cross Site Scripting.