INTERNSHIP

PROJECT-1

Aniket Sunil Pagare

DVWA

DVWA (Damn Vulnerable Web Application) is a deliberately insecure PHP/MySQL web app designed for educational and professional use. Here's a breakdown:

🎯 Purpose & Audience

- **Learning & Testing**: Tailored for security professionals, students, and developers to practice common web vulnerabilities (e.g., SQL injection, cross-site scripting, command injection) in a safe, legal environment .
- **Educational Tool**: Helps users understand how insecure code can be exploited and how to fix it .

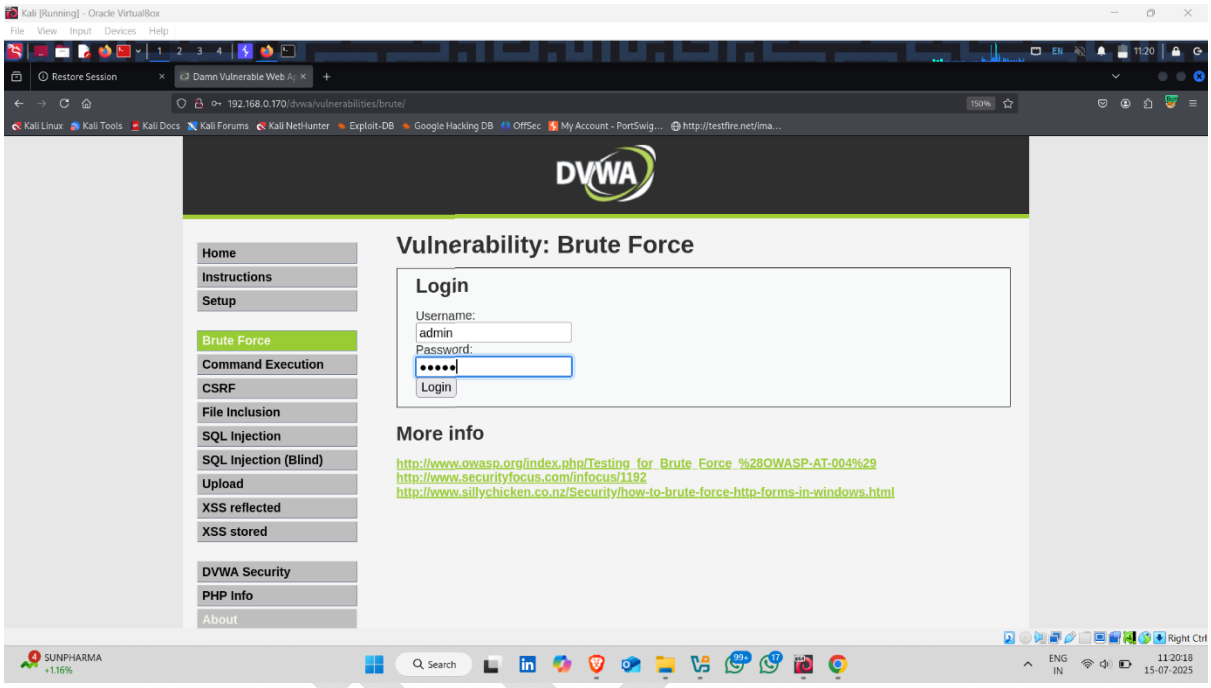
⚙️ Features & Structure

- **Configurable Vulnerability Levels**: Users can adjust the difficulty (low, medium, high, impossible) to match their skill level .
- **Variety of Attack Vectors**: Built-in modules include SQLi, XSS, CSRF, file upload, command injection, and more .
- **Both Documented & Undocumented Flaws**: Encourages exploration and discovery beyond guided tasks .

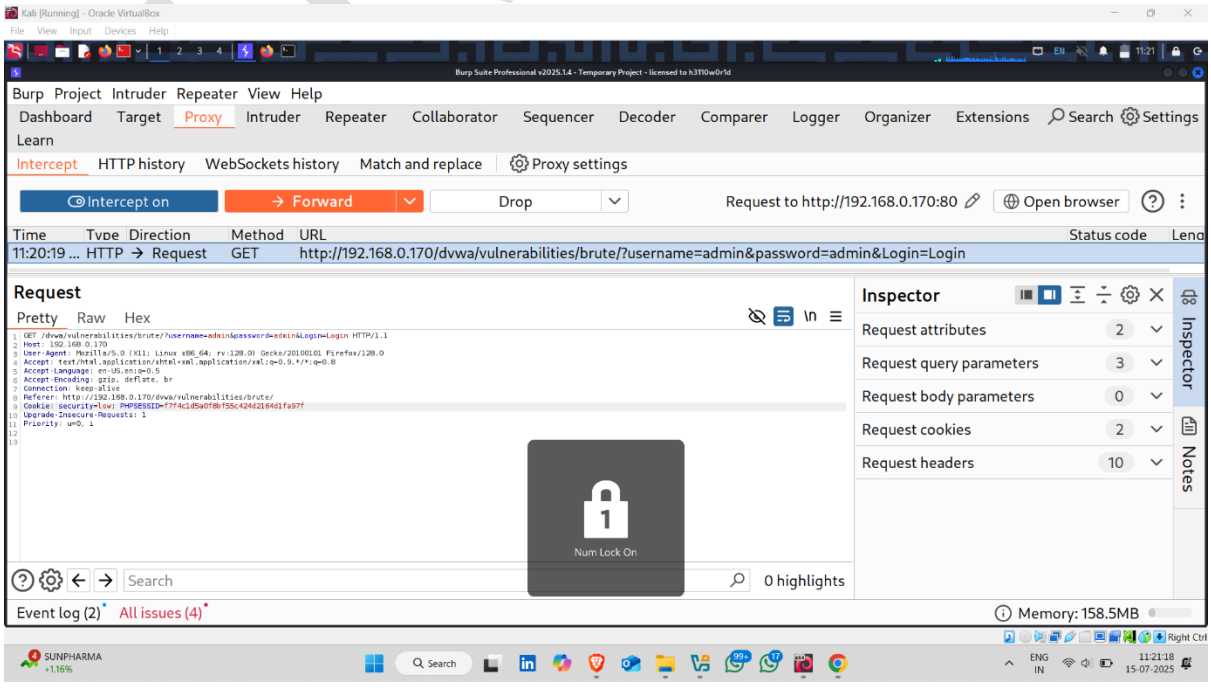
Security Level – Low

Task -1 – Brute Force Attack

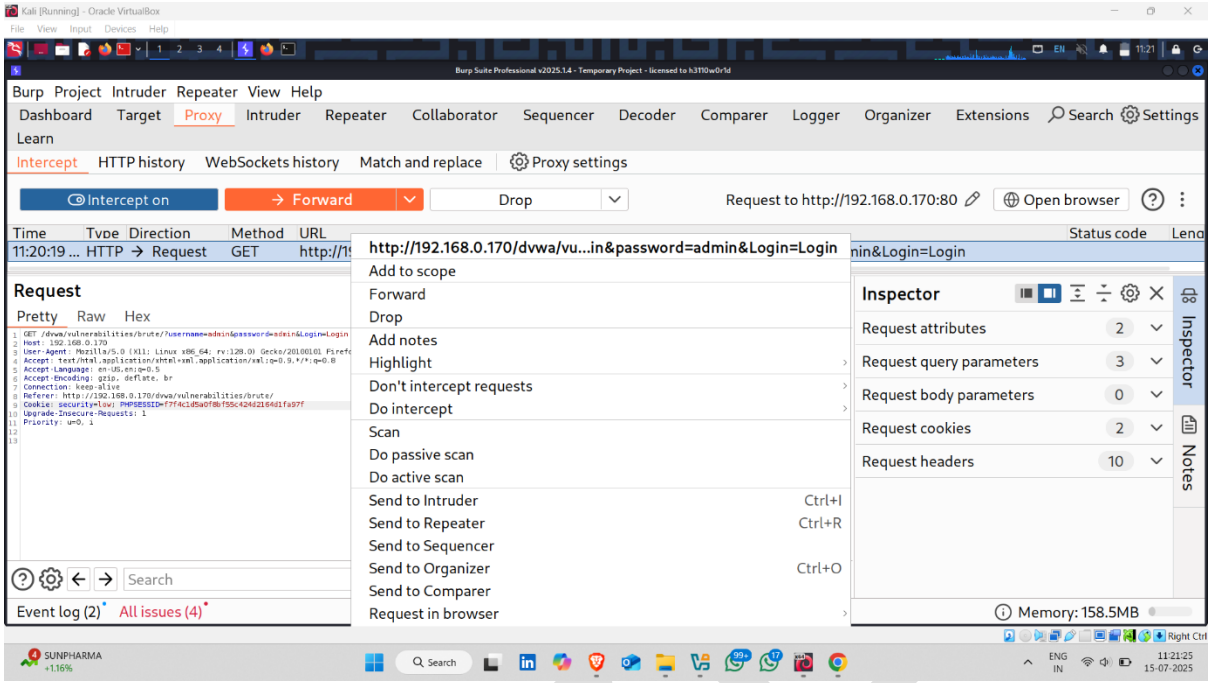
- Open DVWA Project , click on **Brute force**
- Enter **username as a admin** and enter **random password**



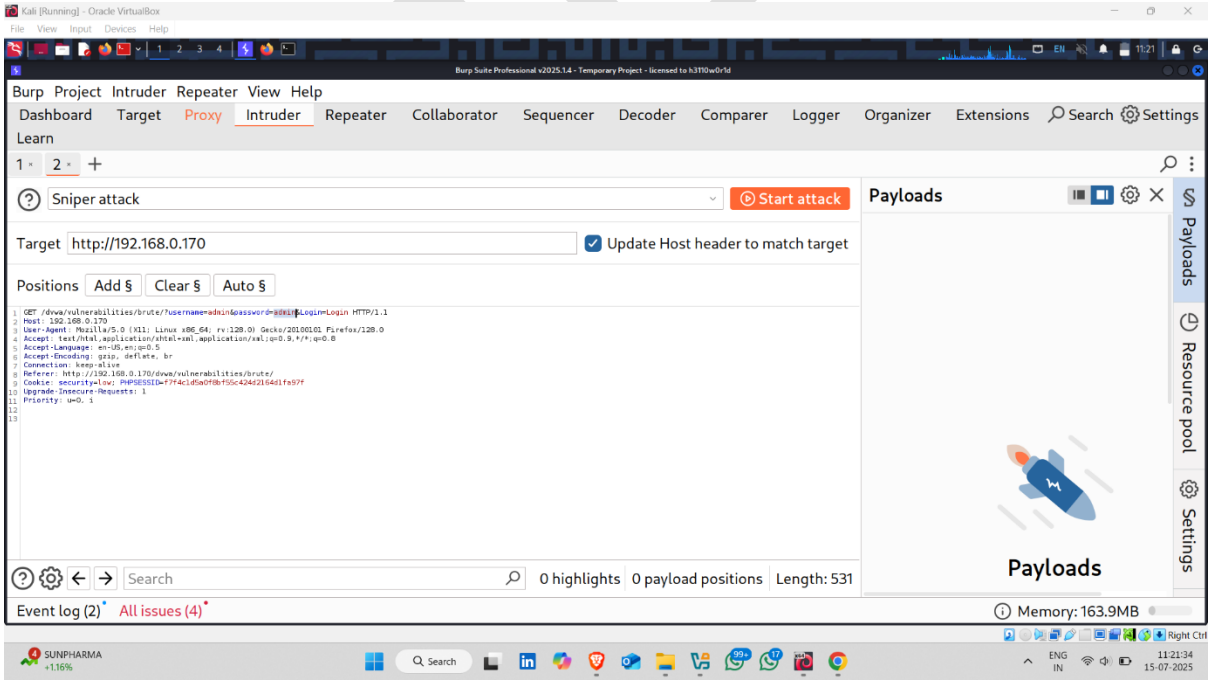
- Request intercept in burp suite



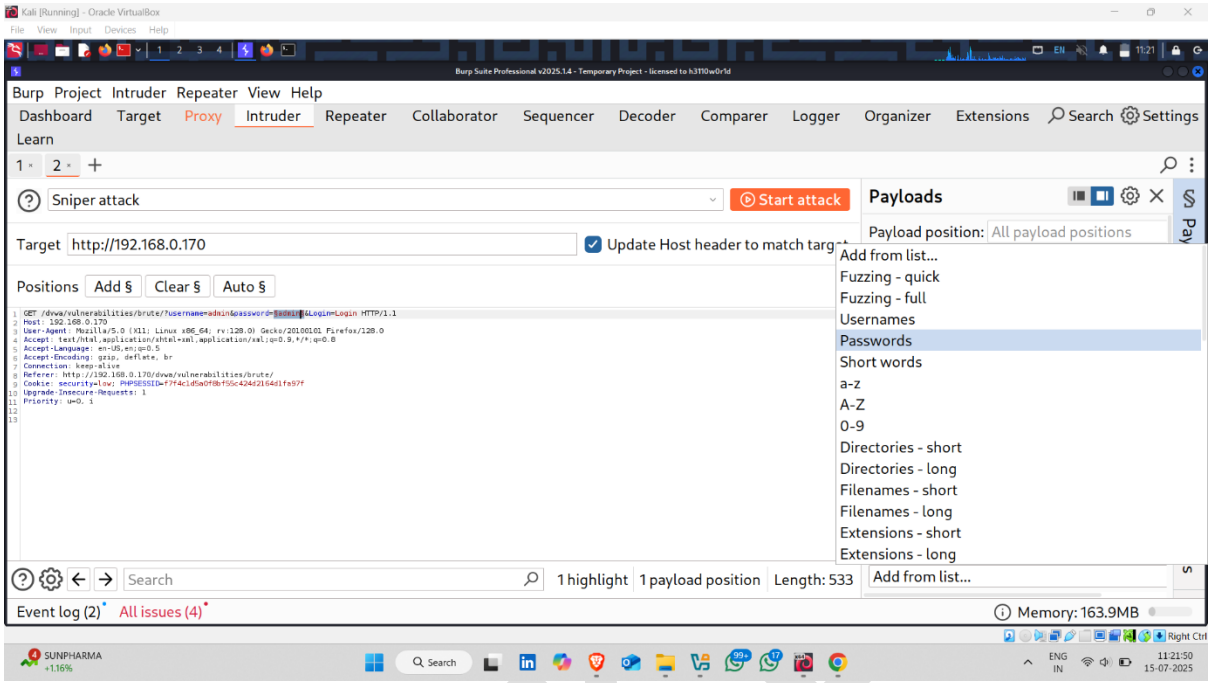
- Send this request to the **intruder**



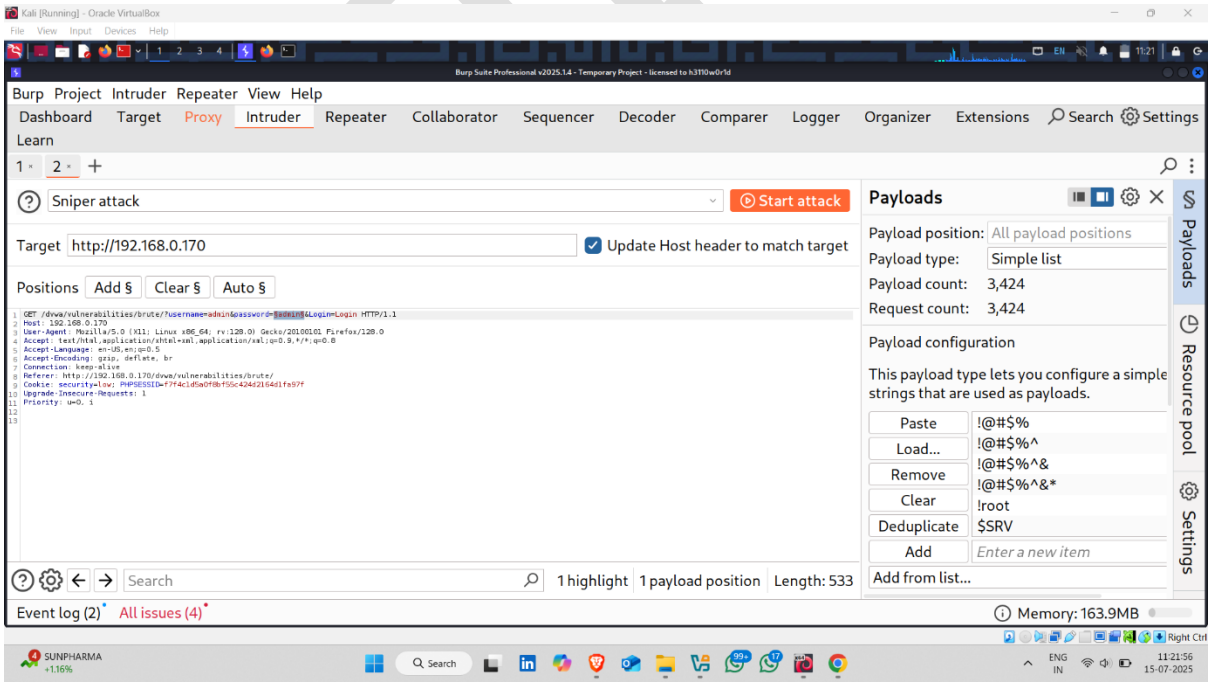
- Set payload position to password



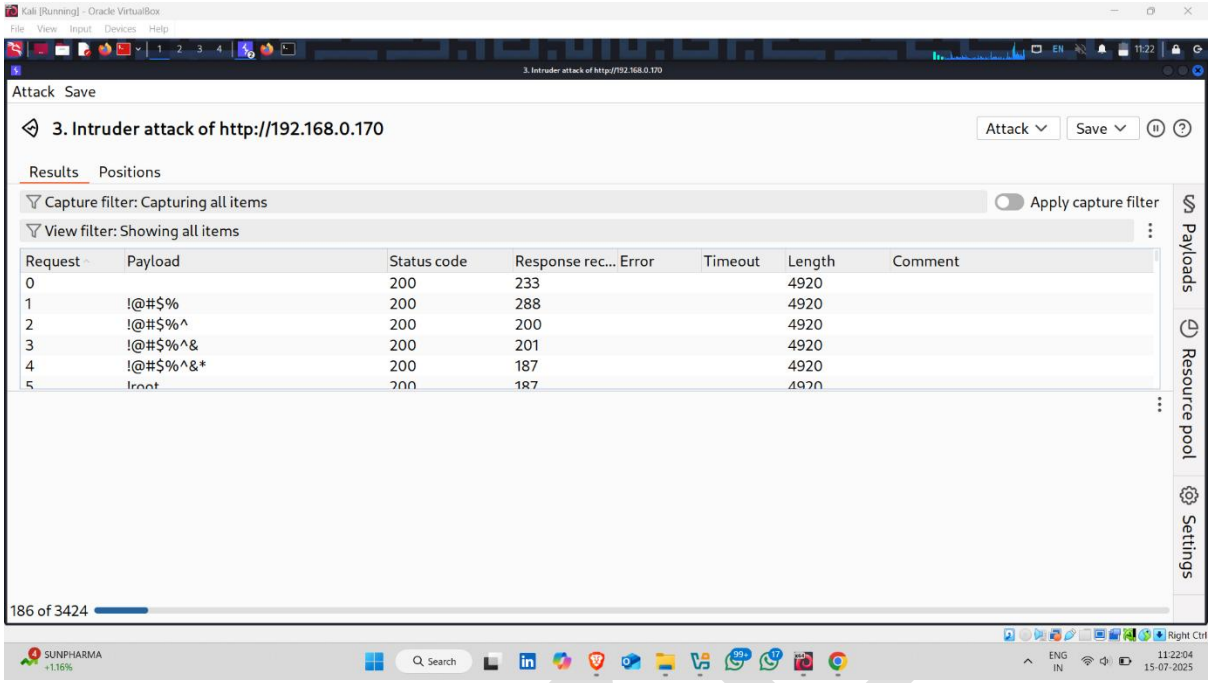
- And in payload section , click on Add from List and select **password**



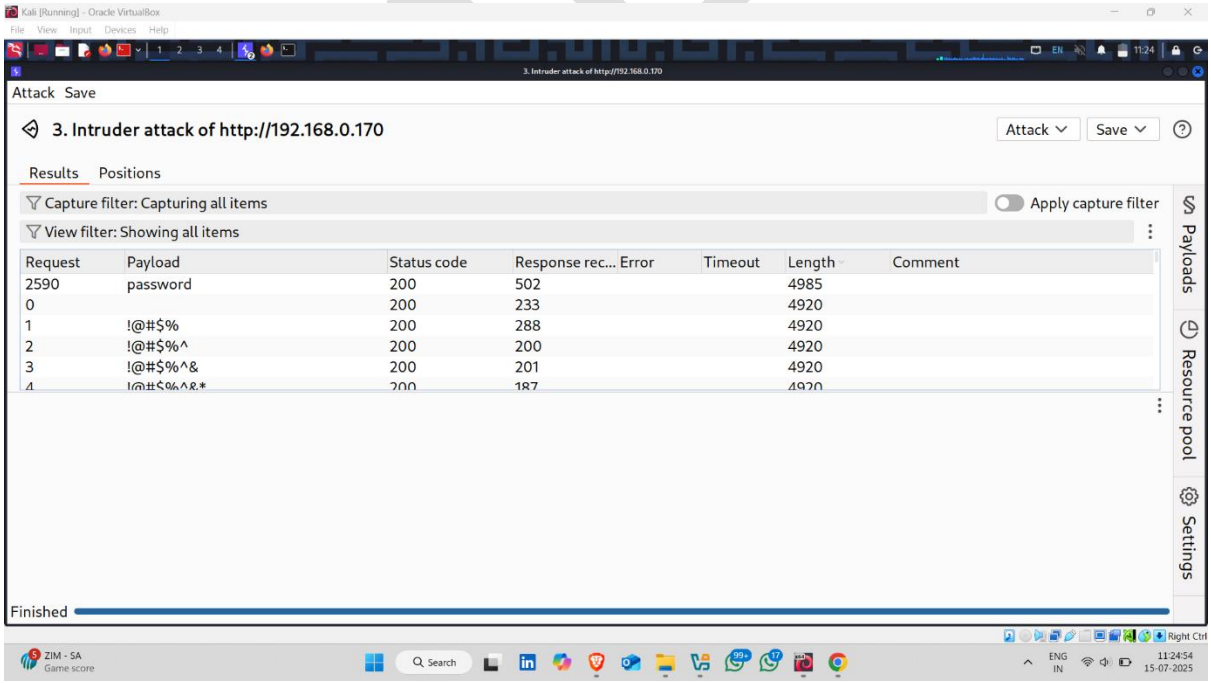
- Now click on **start attack**



- Attack Started 👇

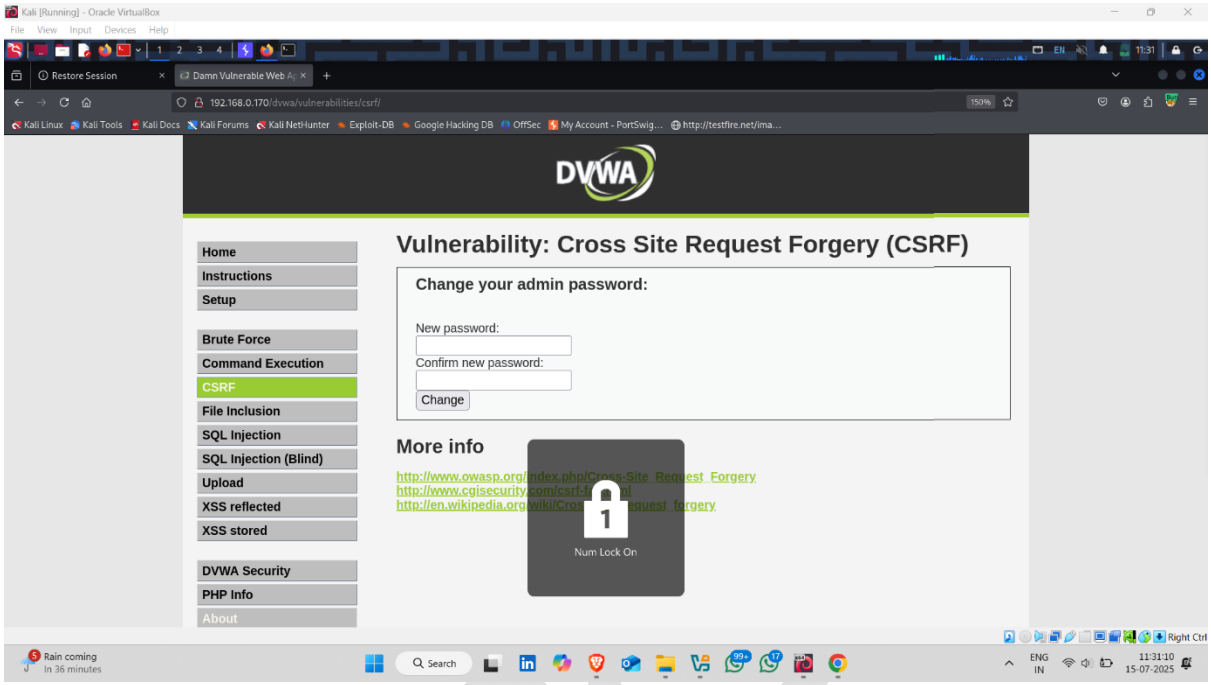


- Password find 👇✅

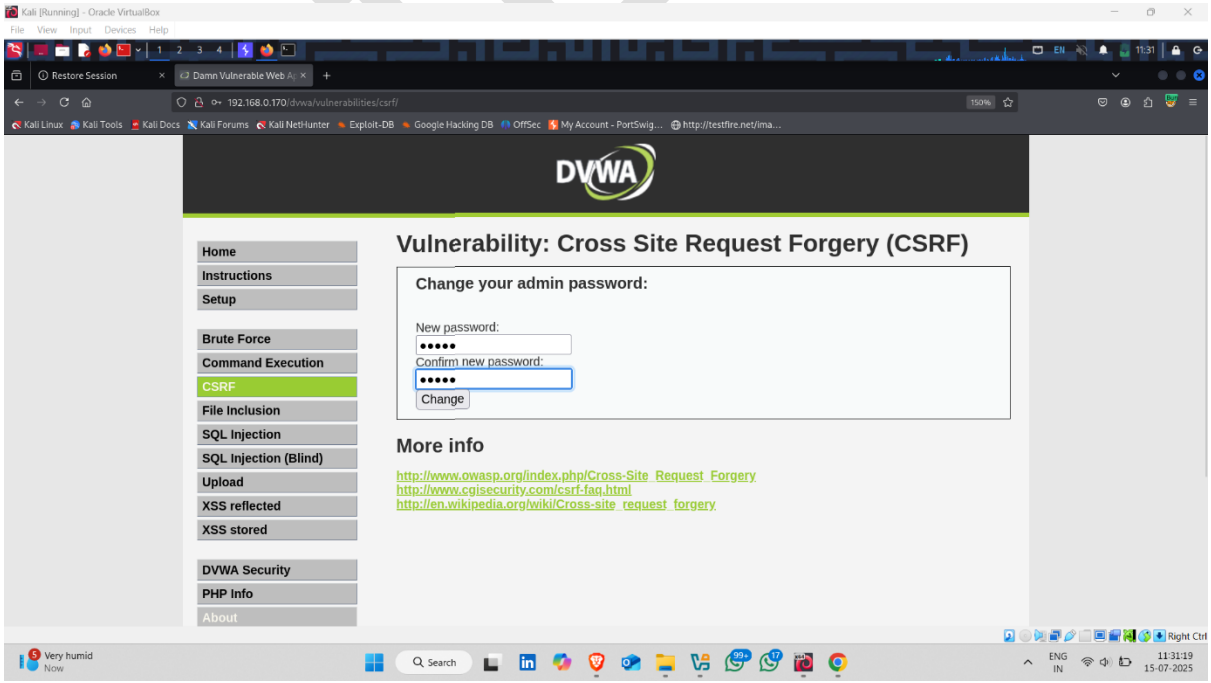


Task-2 – Cross Site Request Forgery (CSRF)

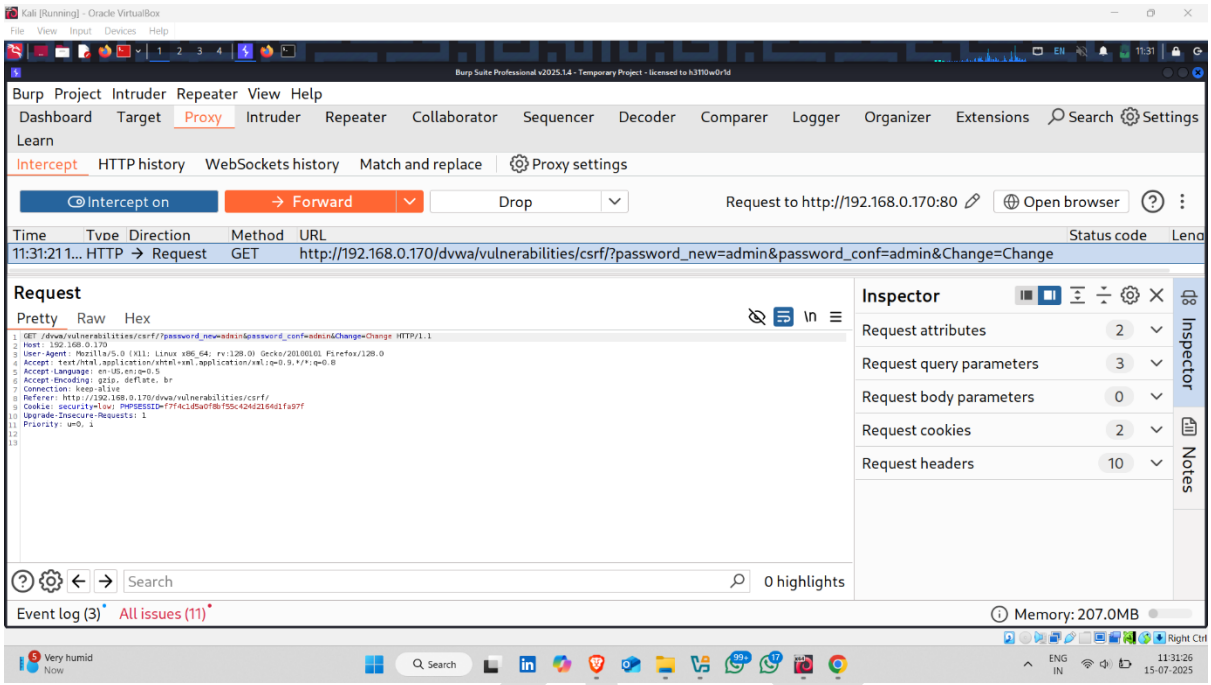
- Click on CSRF



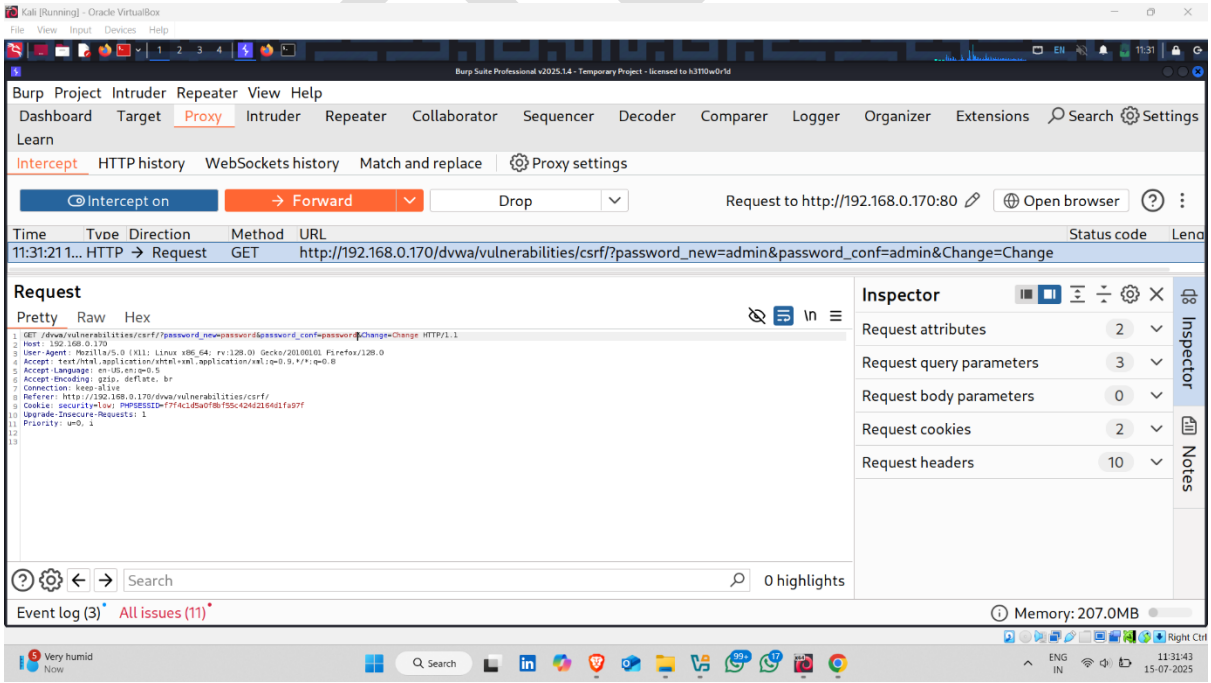
- Provide random password and confirm password



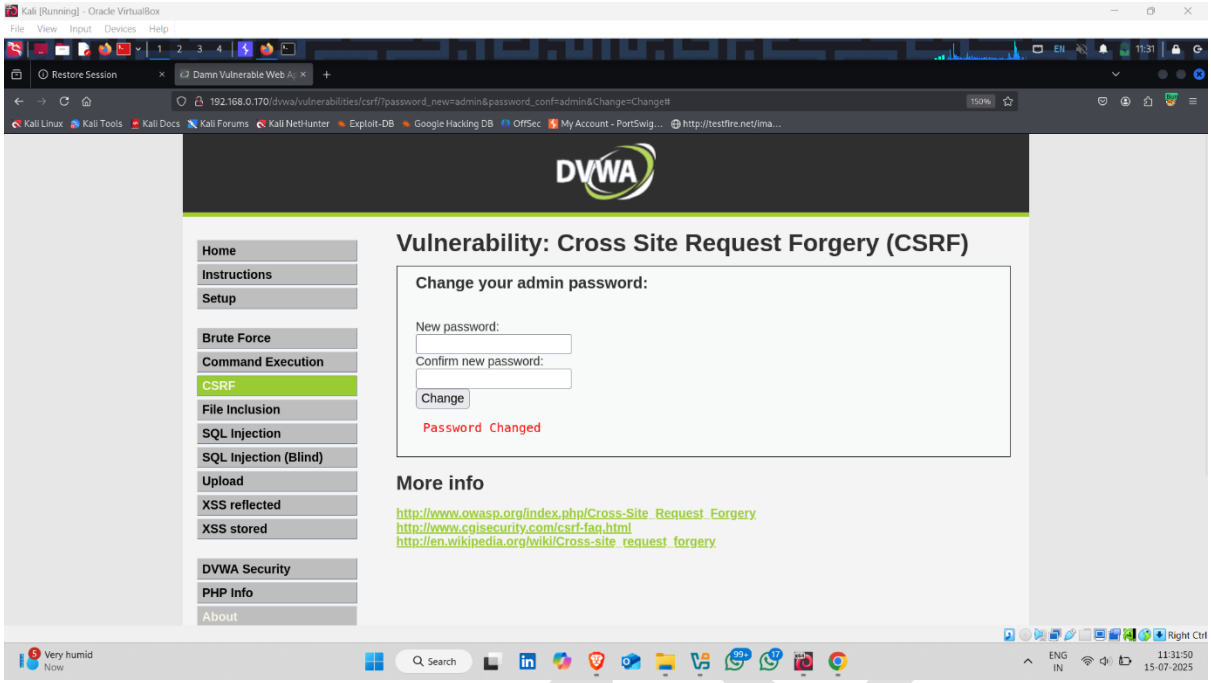
- Request Capture in burp suite 👇
- Now modify this request with your own password



- Forward this request

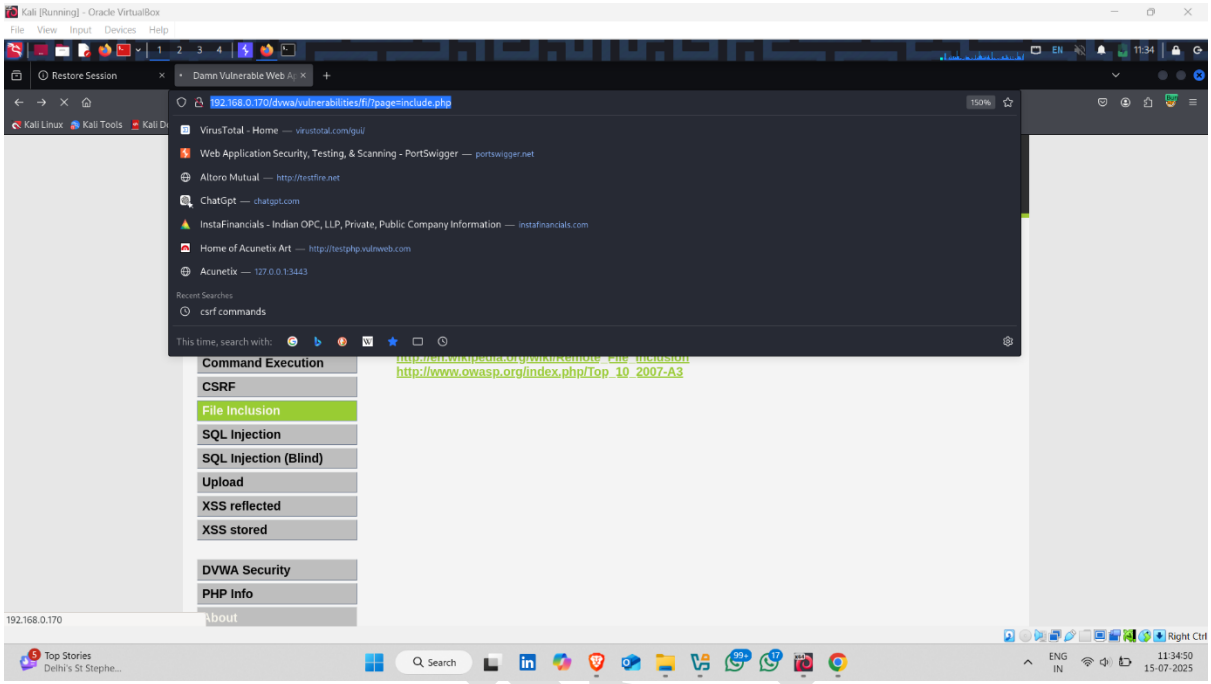


- Password changed 👇 ✅

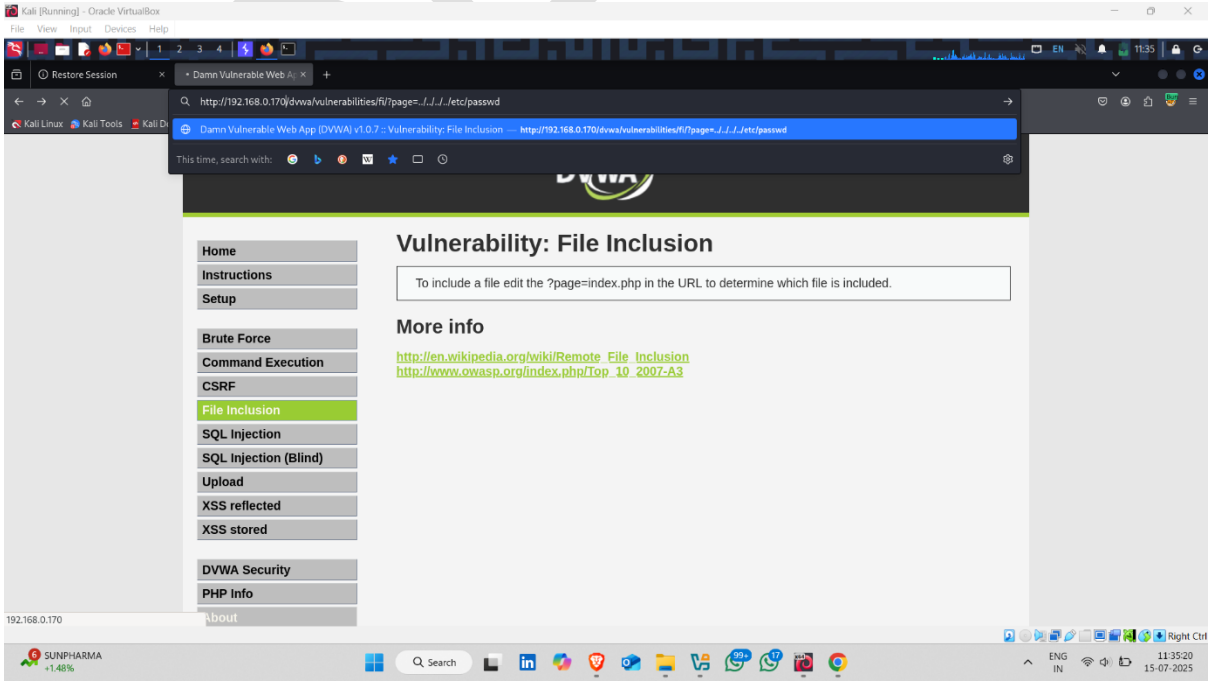


Task-3 – File Inclusion

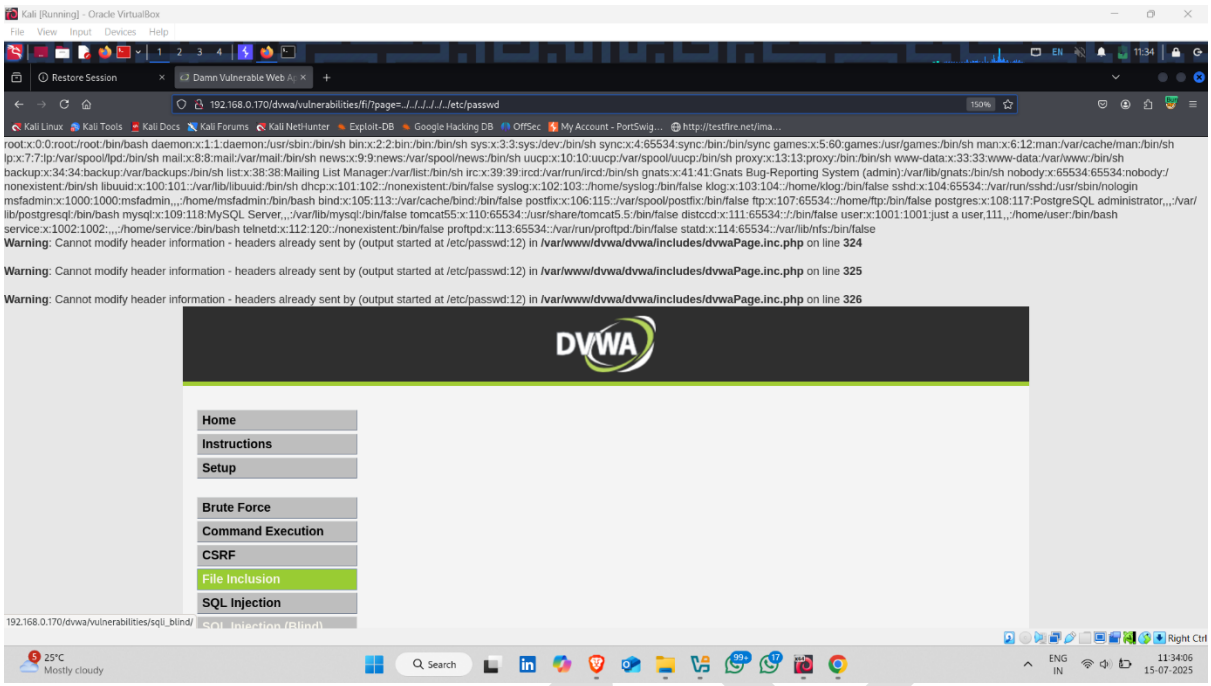
- Click on url section

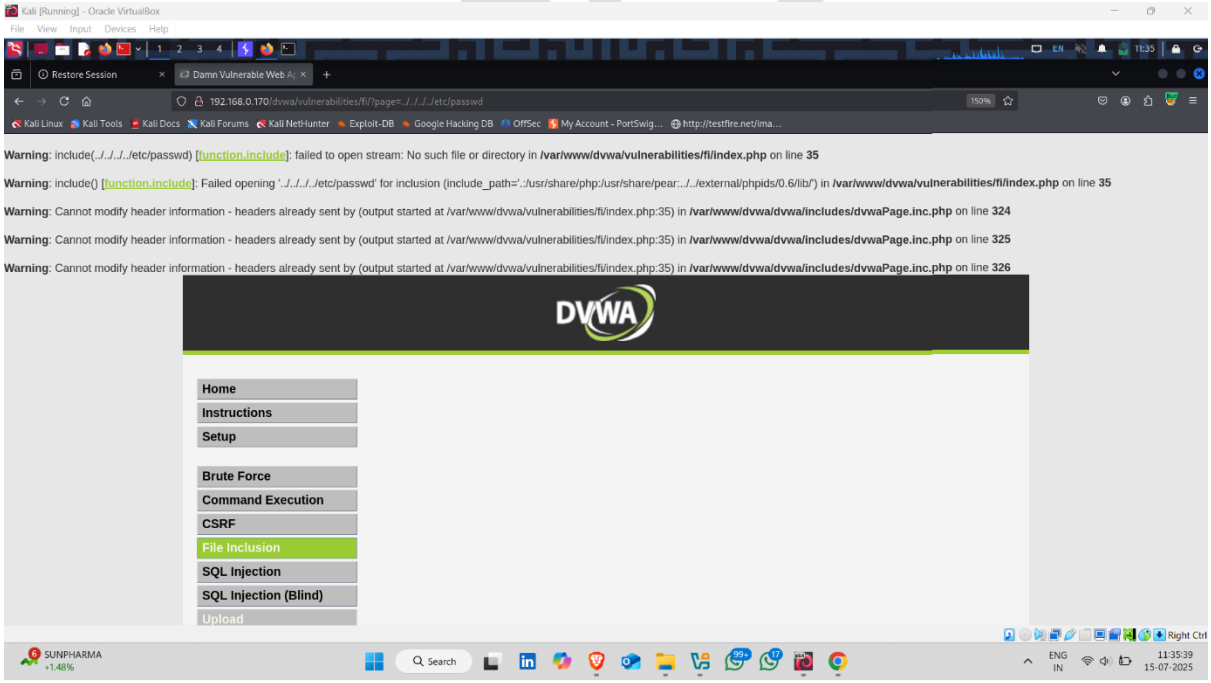


- And enter following url , To check Directory Transversal happen or not 👇✅



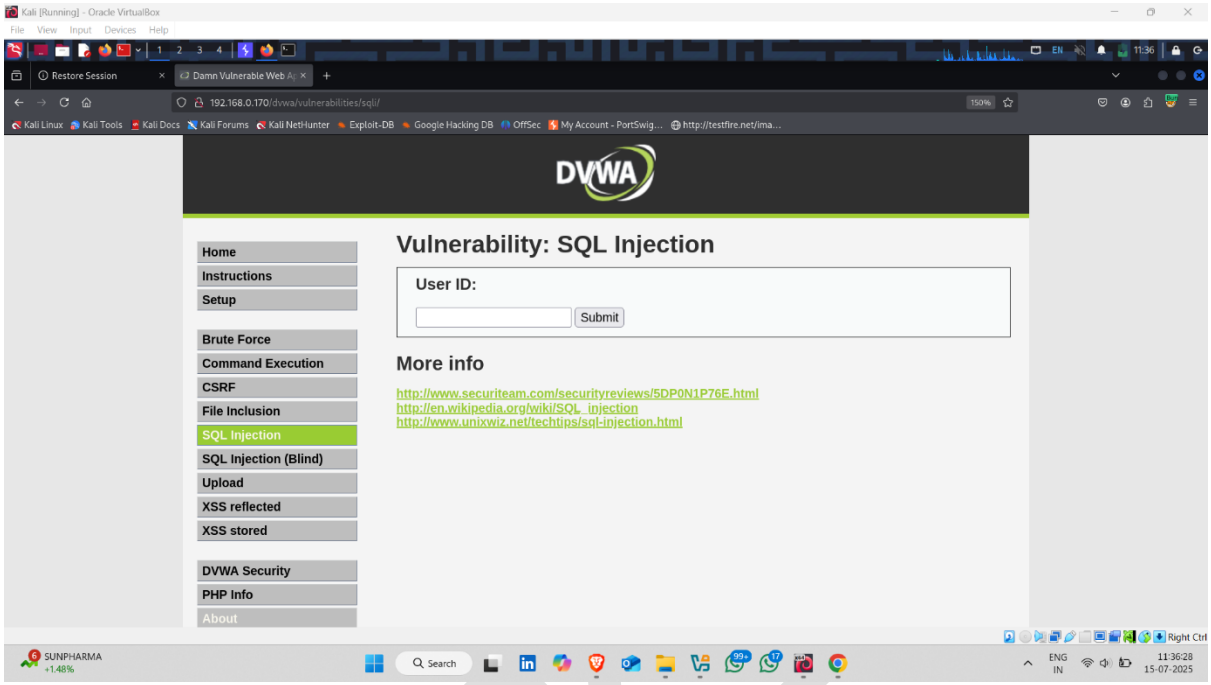
- Directory Transversal ✅ 👇



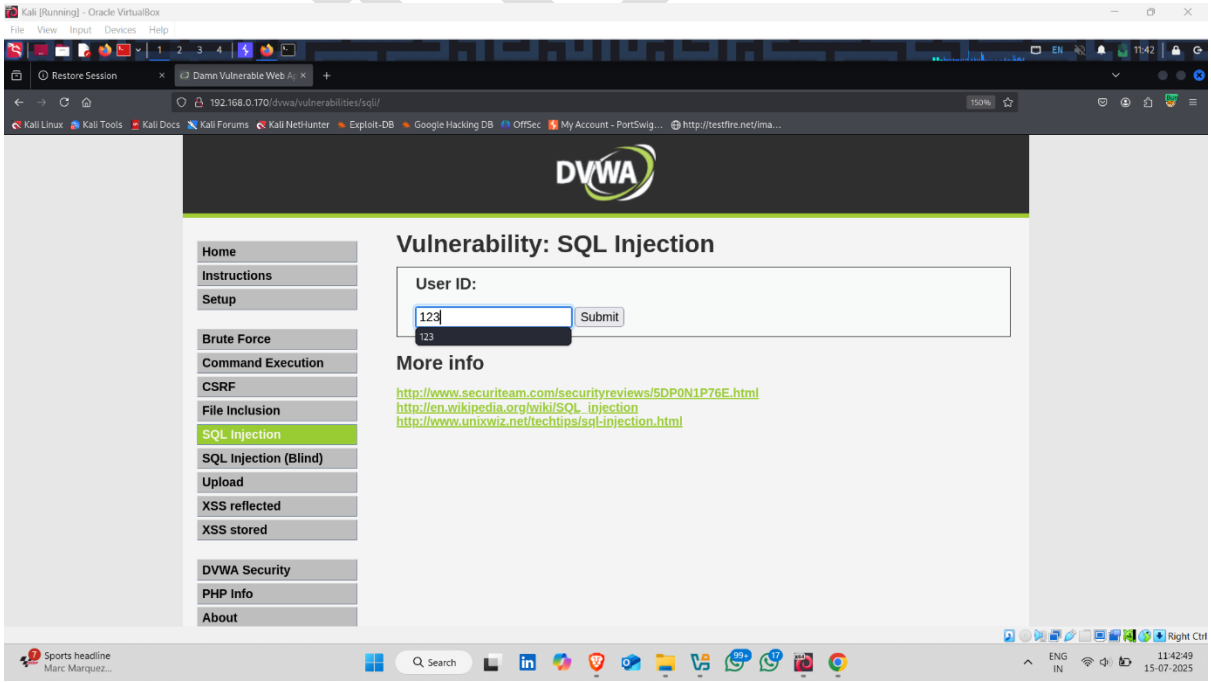


Task -4 – SQL Injection

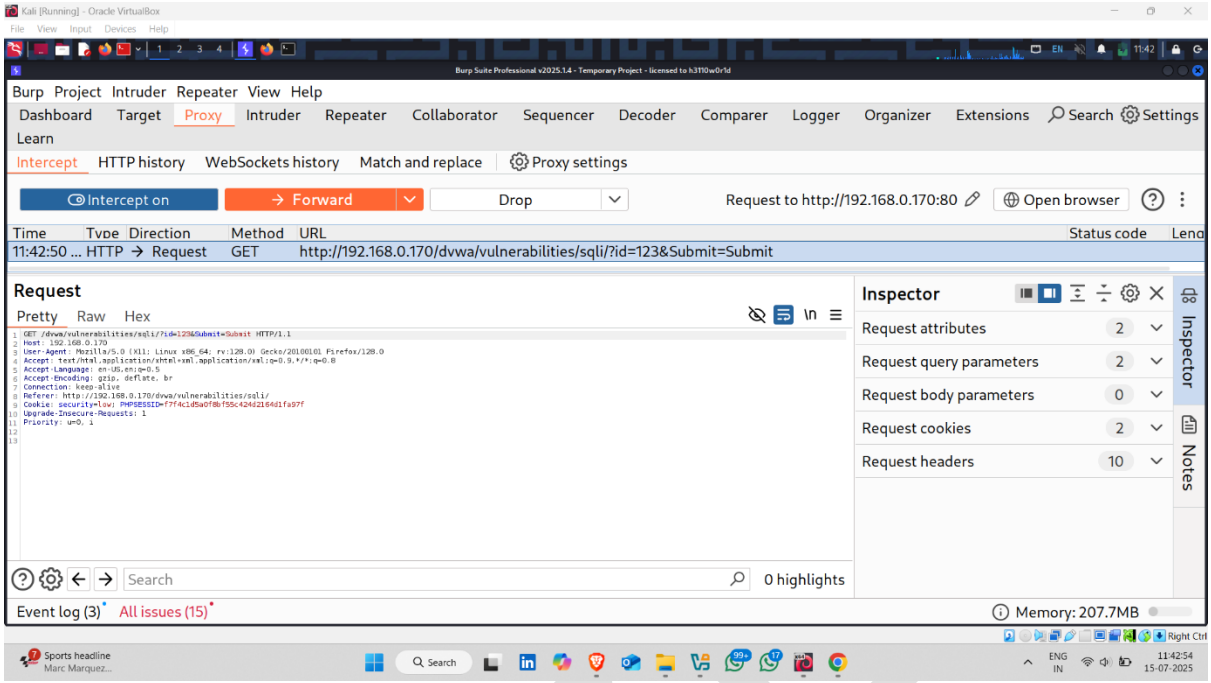
- Click on Sql Injection Section



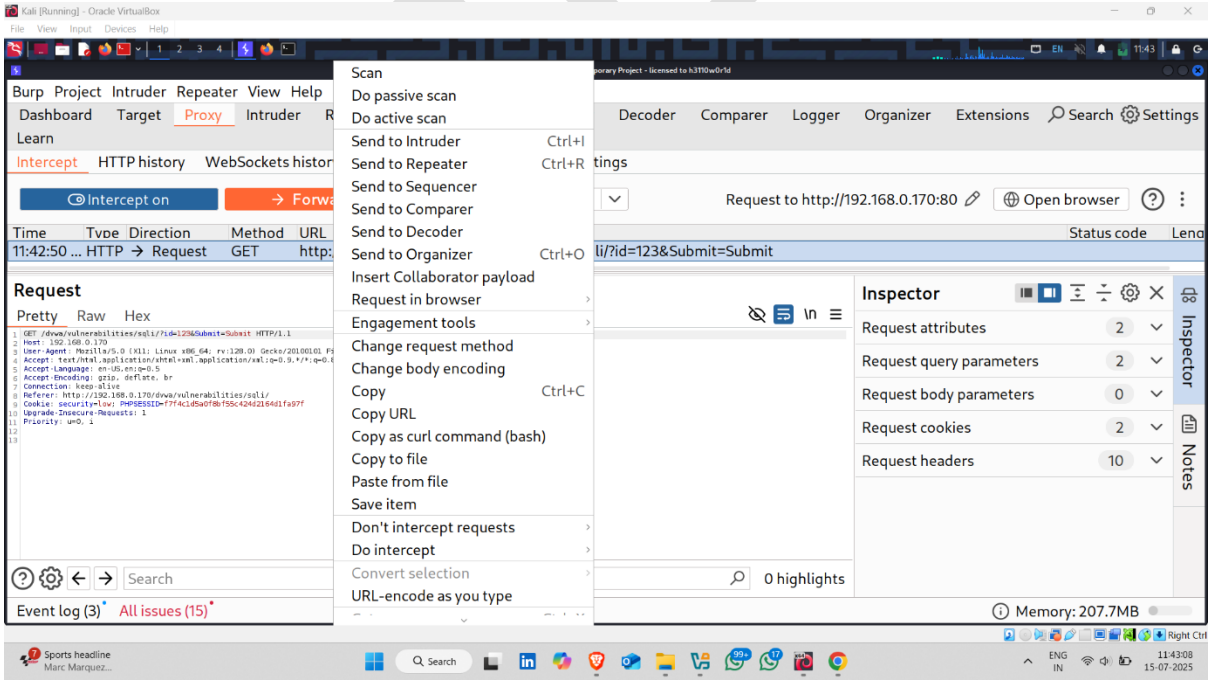
- Enter random Id and click on submit button



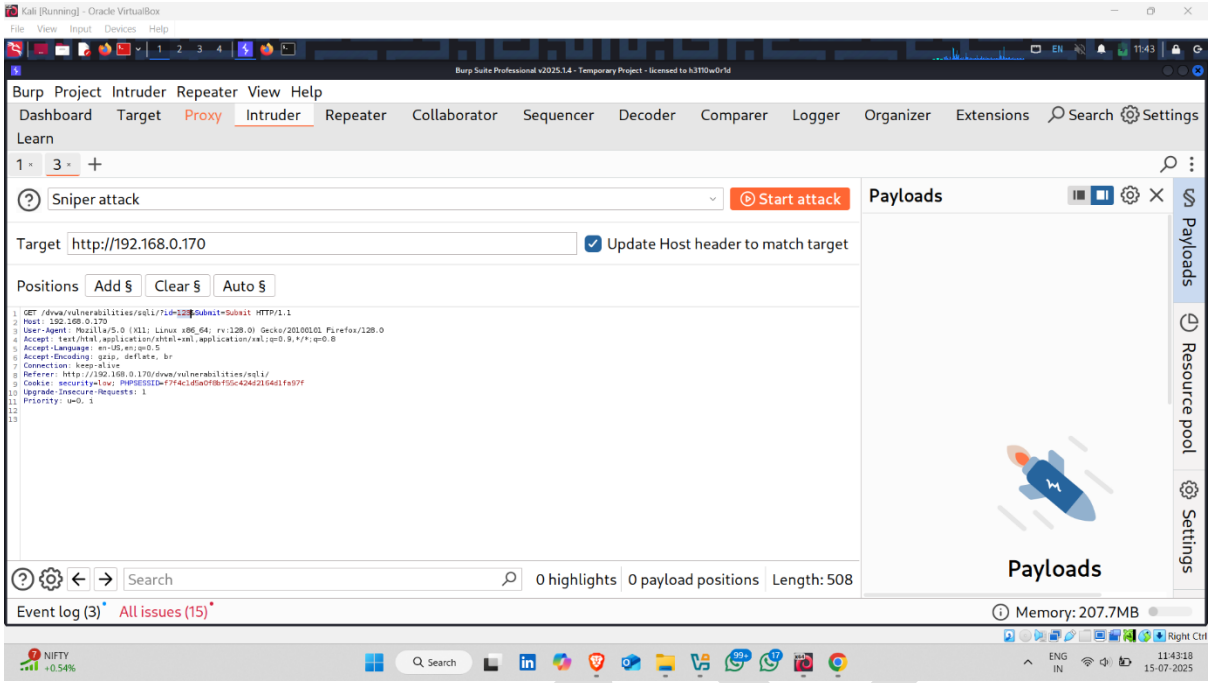
- Request intercept using burp suite



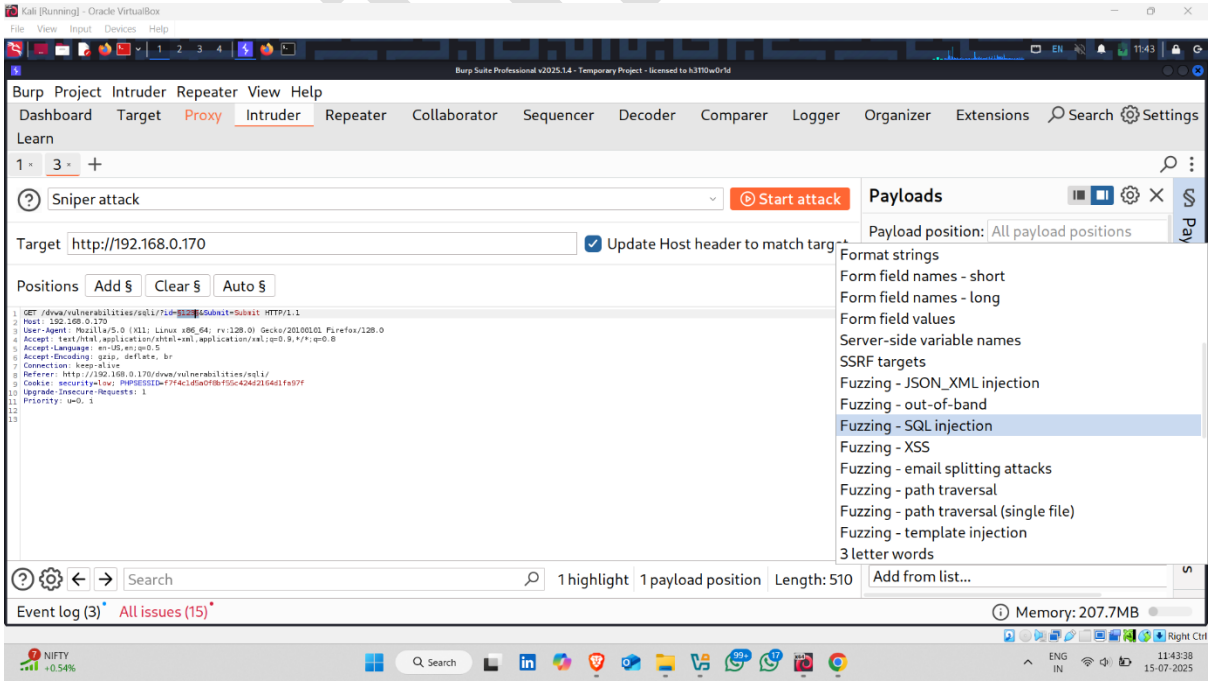
- Right click on request and send it to **intruder**



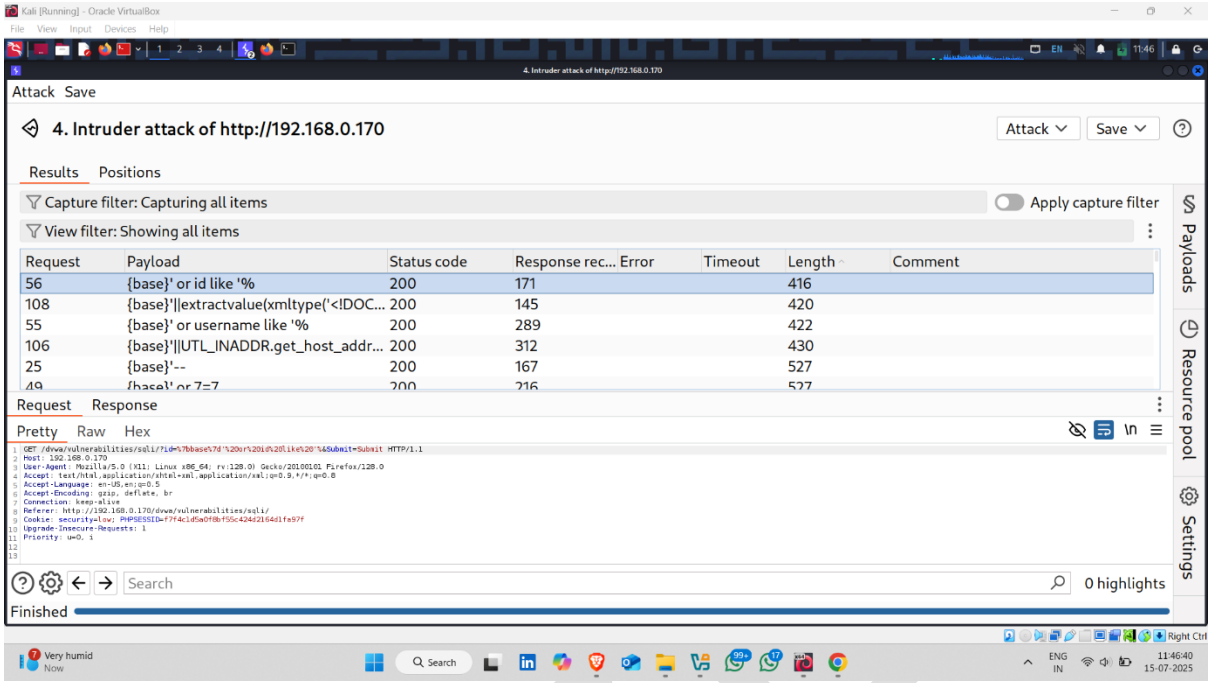
- Set payload position on **id section**



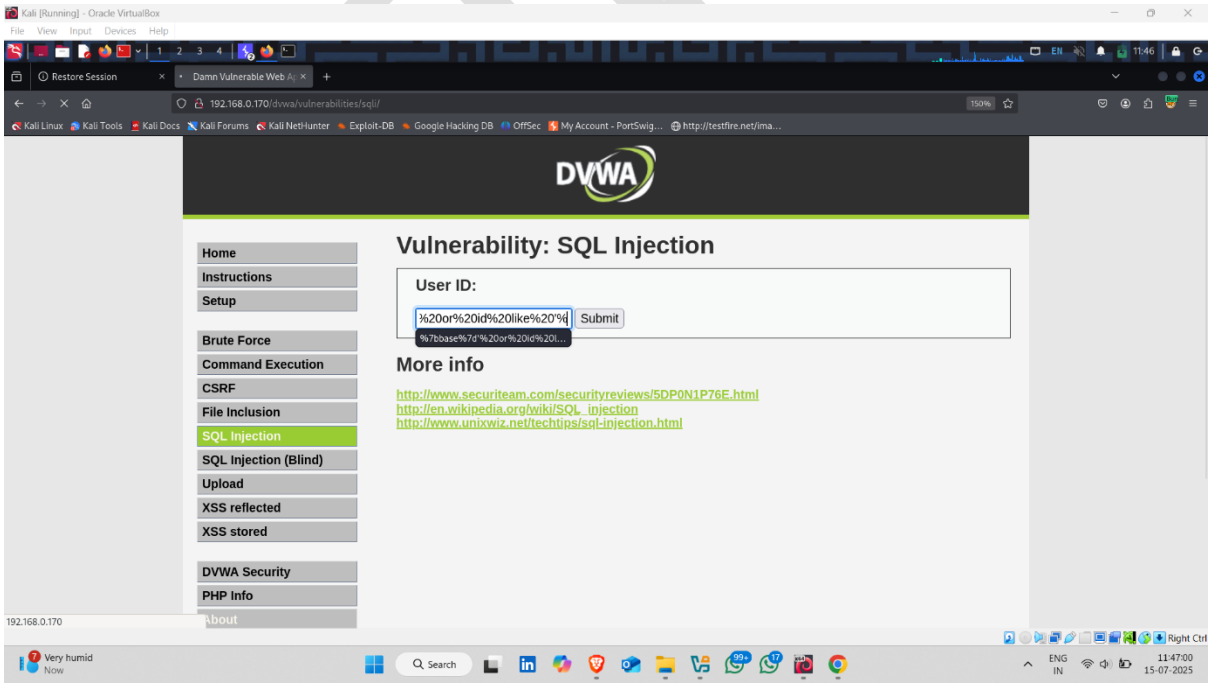
- Now click on **payload configuration** and then click on **Add from list** and select **fuzzing- SQL Injection**
- Click on **Start attack**



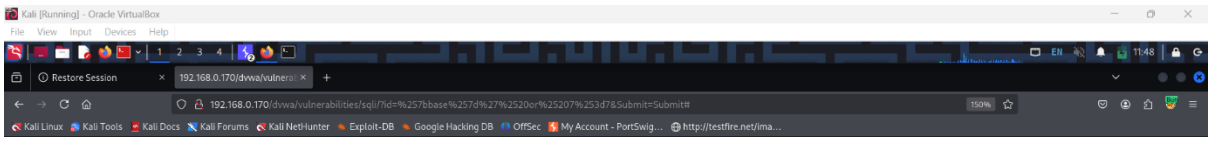
- Automated attack finished



- Now enter sql injection on input field for manual testing
- Click on submit

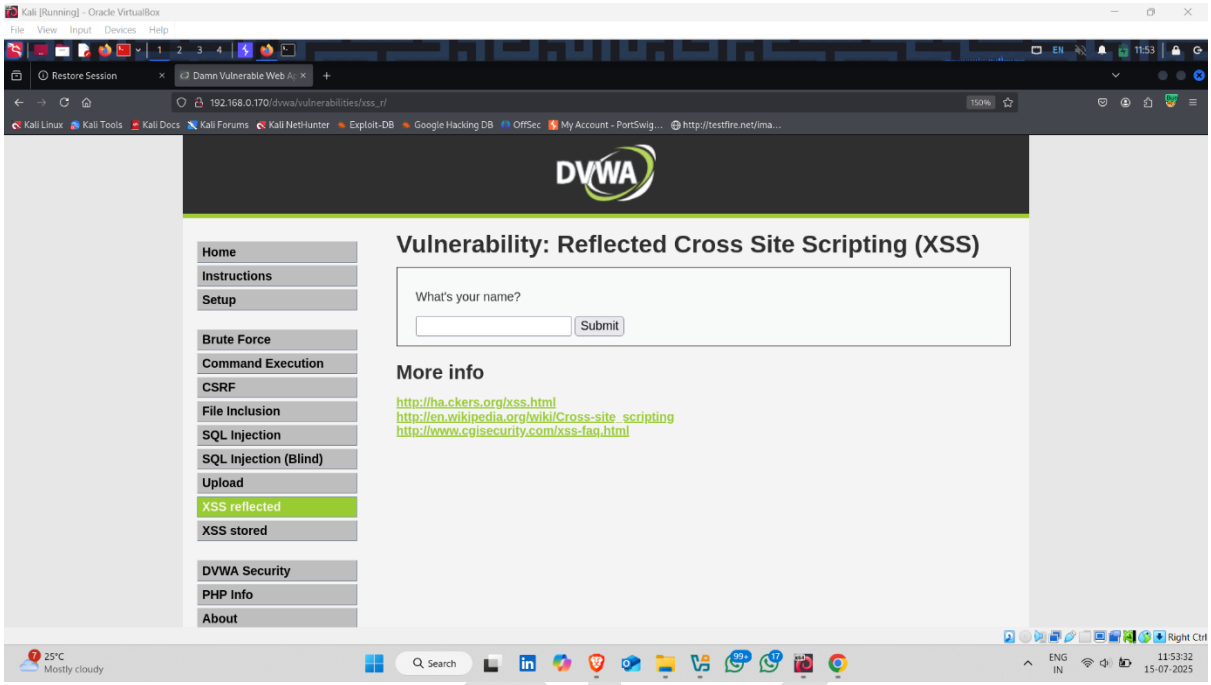


- Result 👇✅

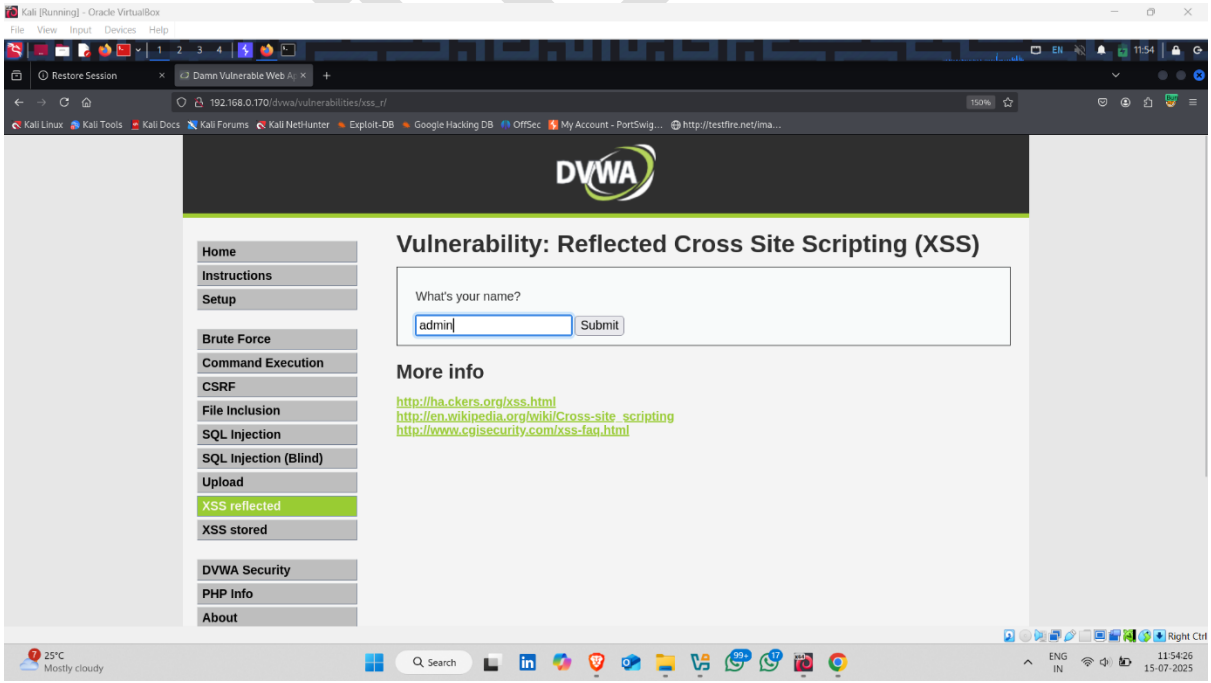


Task-5—Cross Site Scripting reflected (XSS)

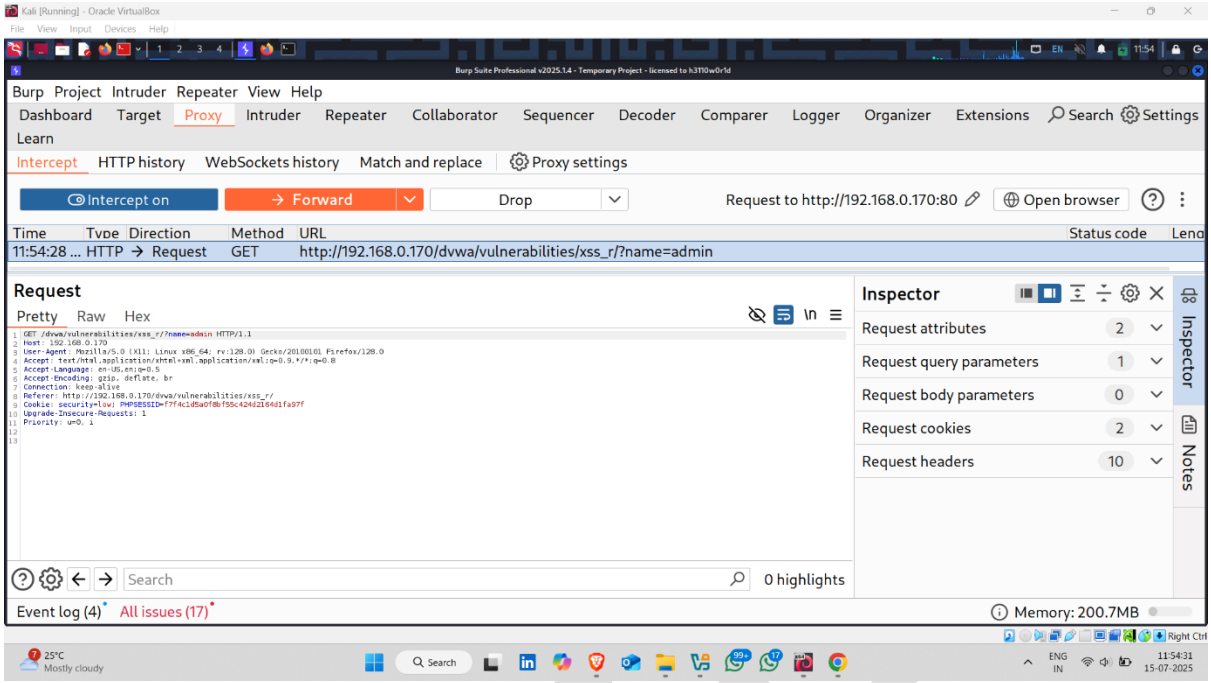
- Click on Xss Reflected Section



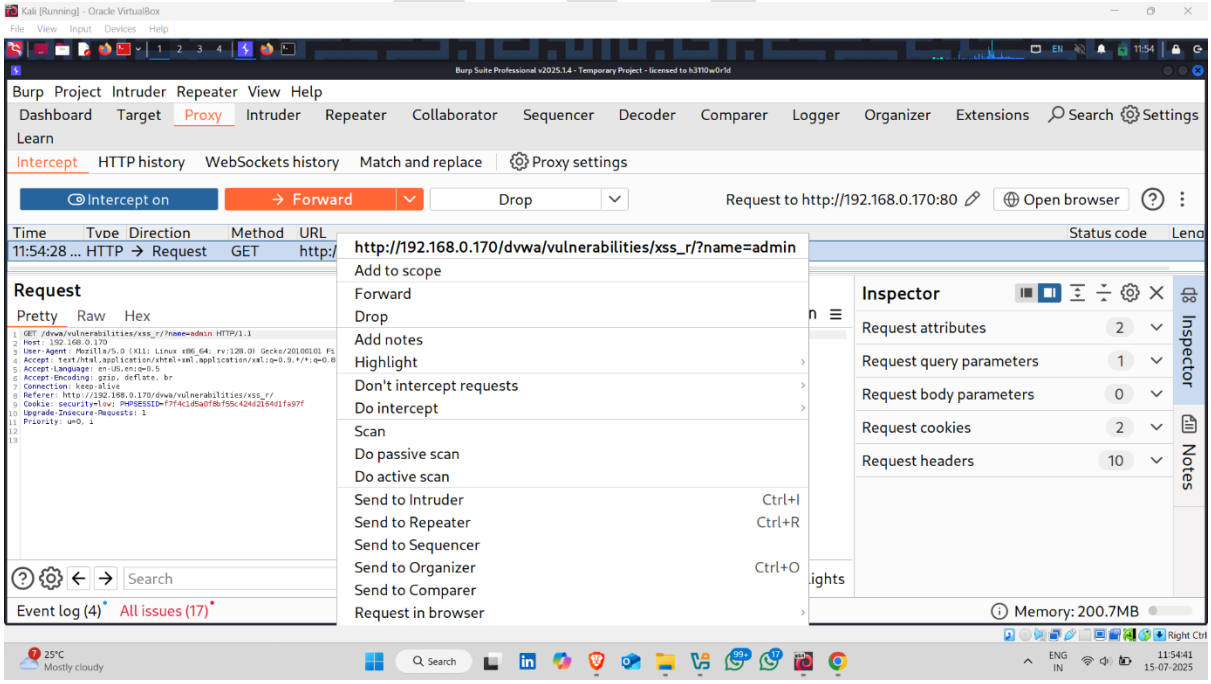
- Enter a string and click on submit



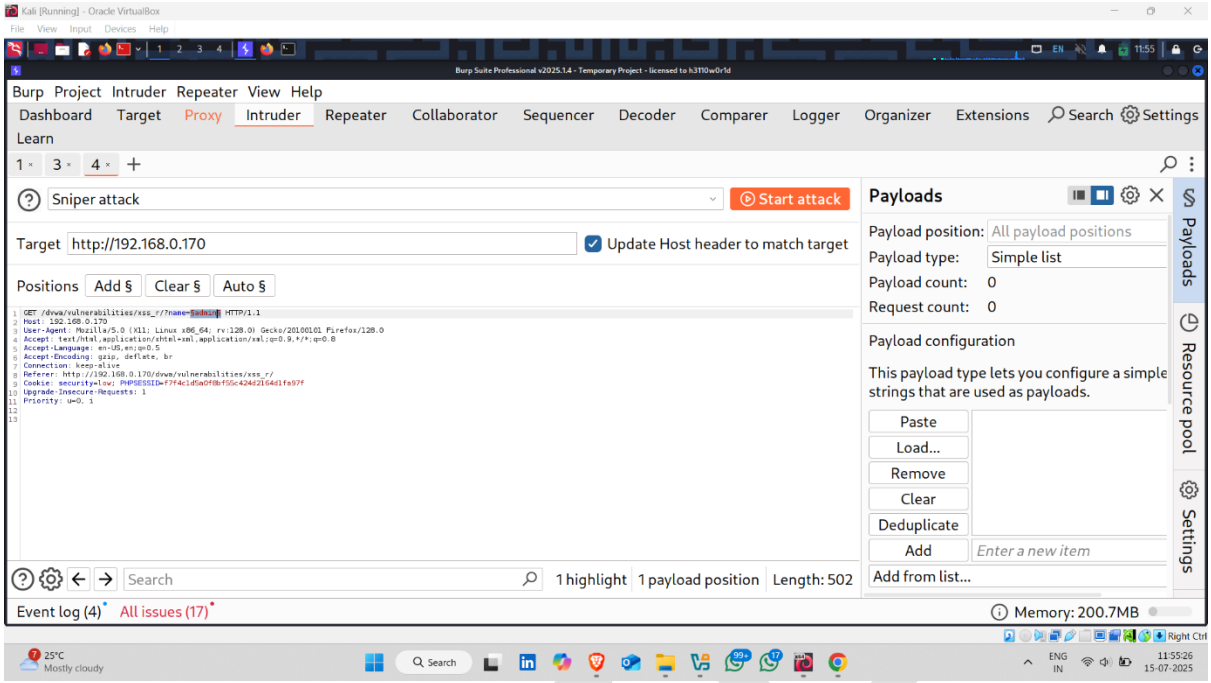
- Request intercepted



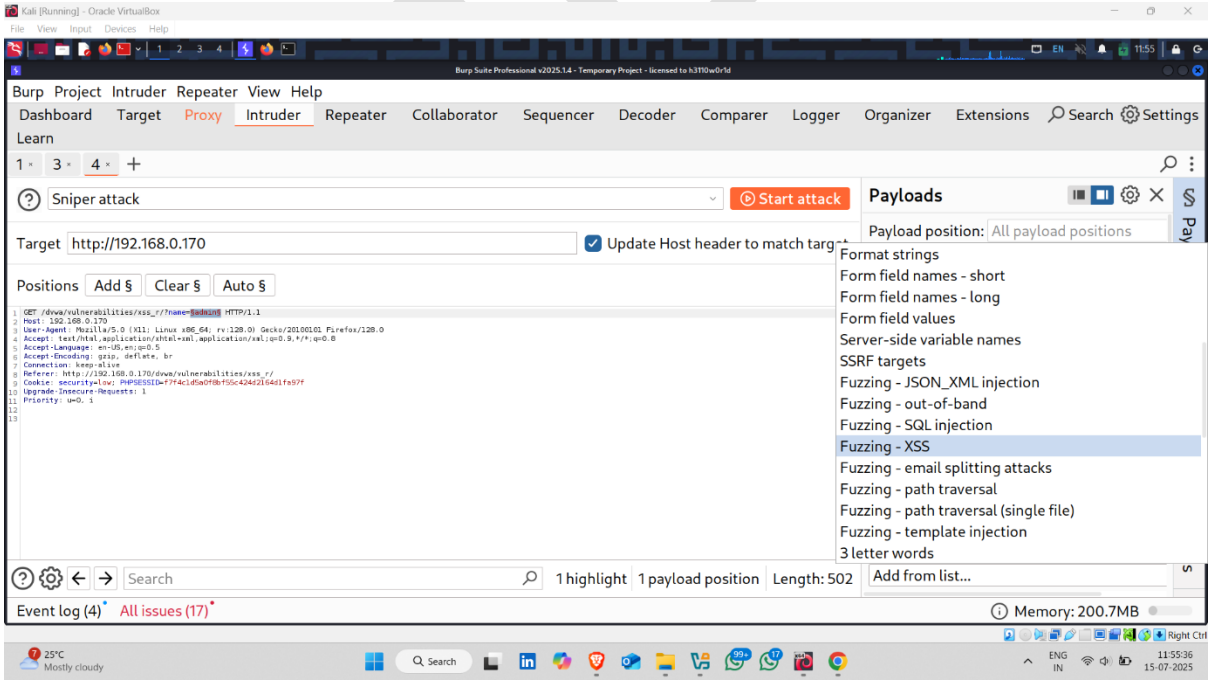
- Send this request to the **intruder**



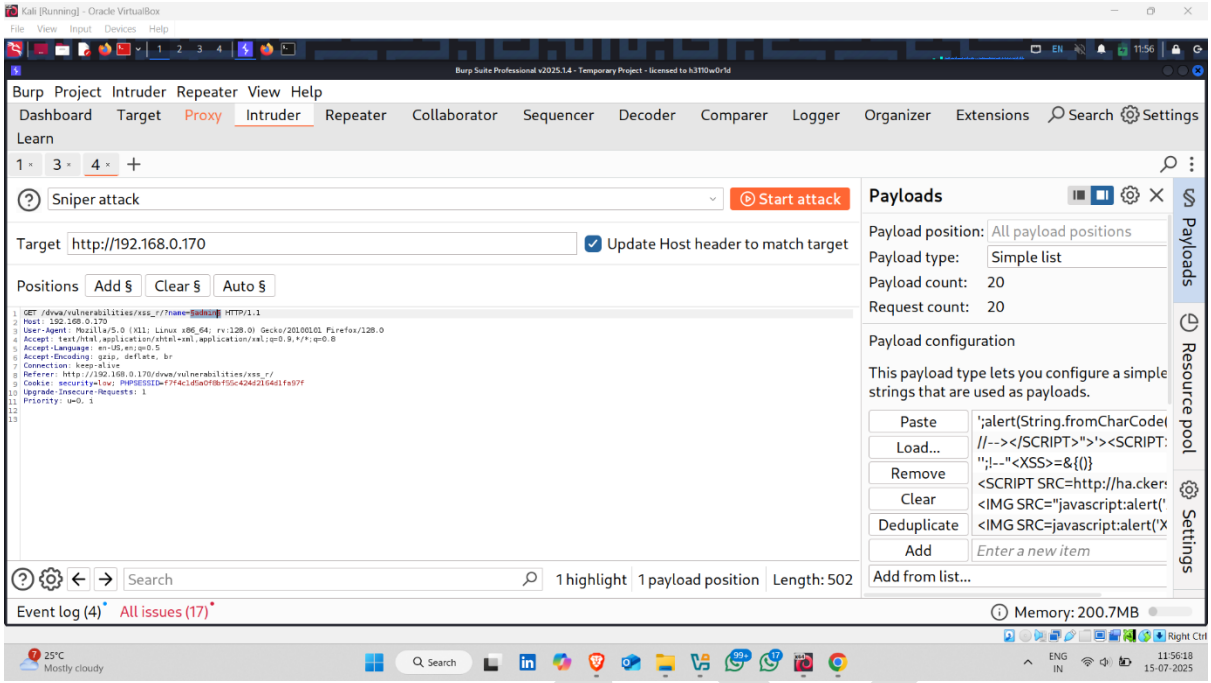
- Set position to **name** section



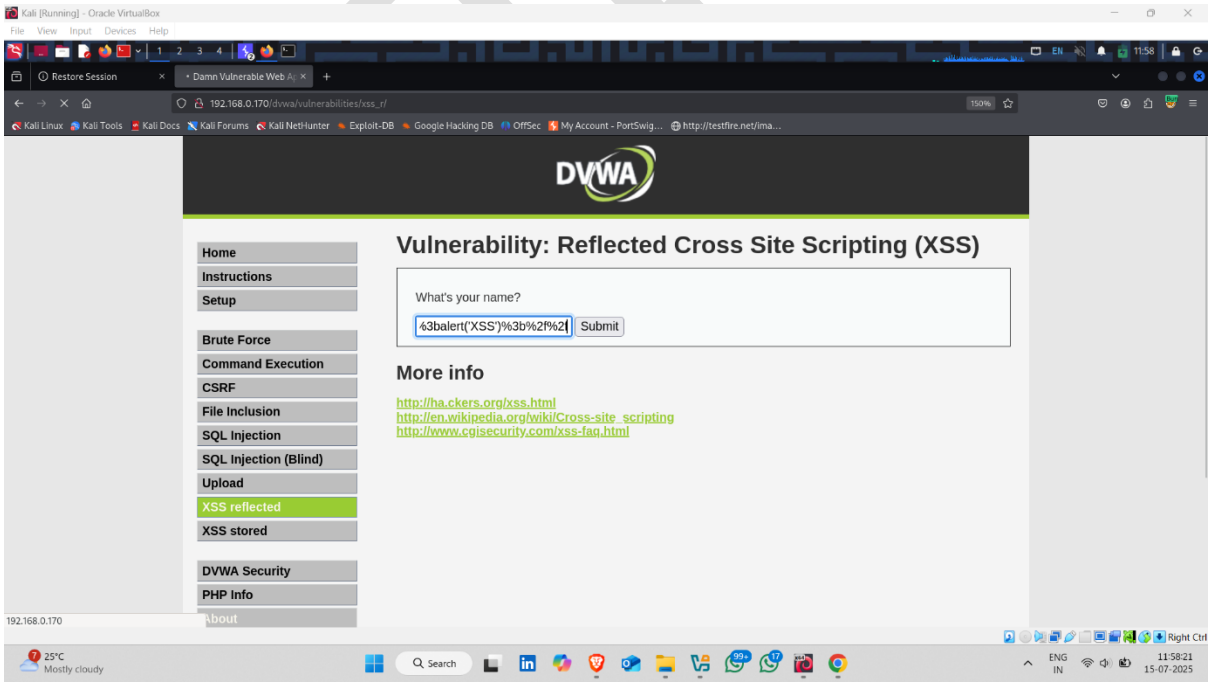
- Select **fuzzing-XSS**



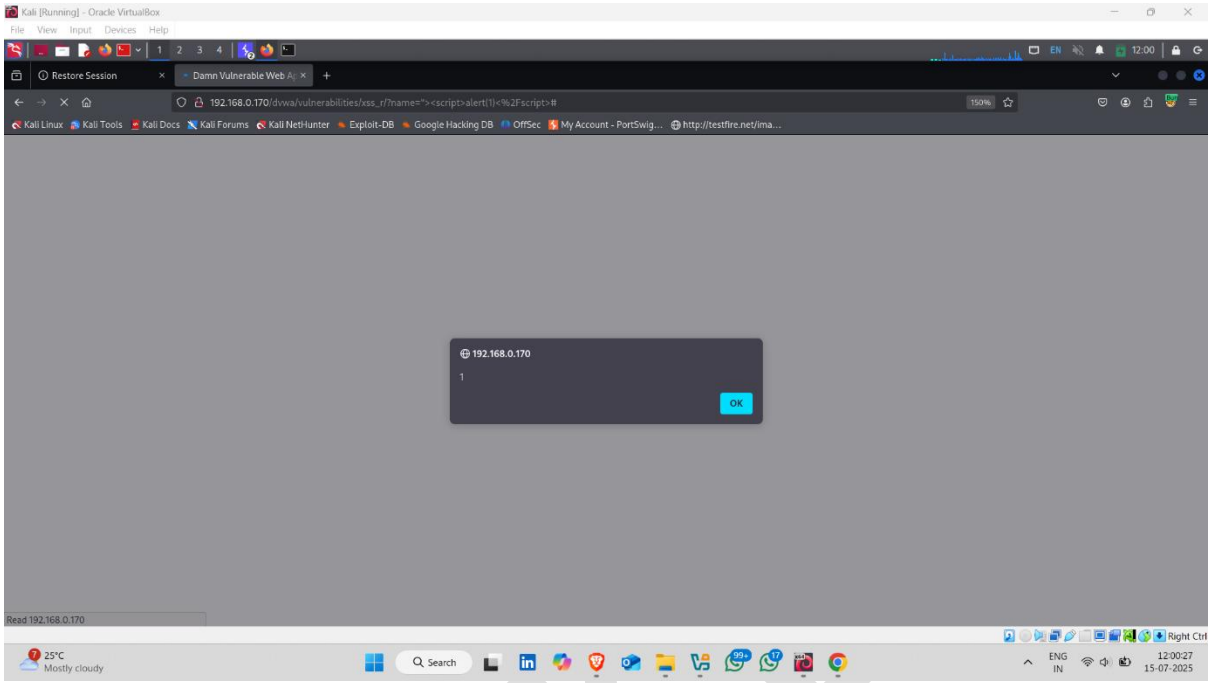
- Click on **start attack**



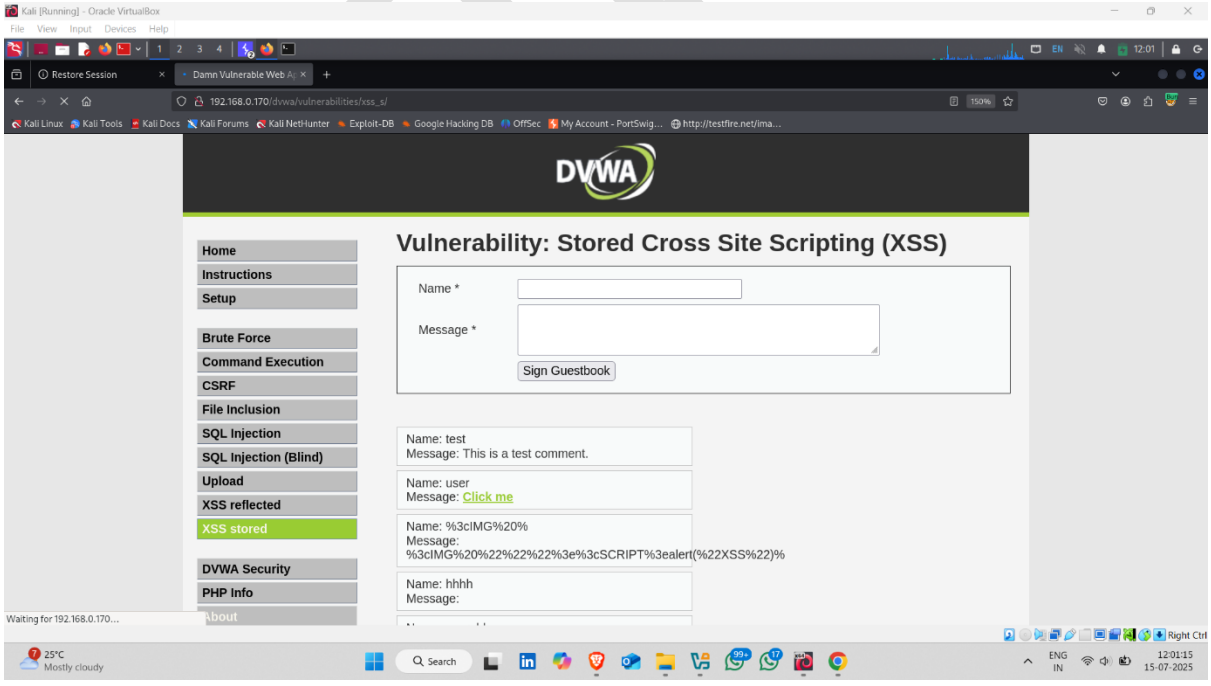
- Now enter string in input section for manual testing and click on submit button



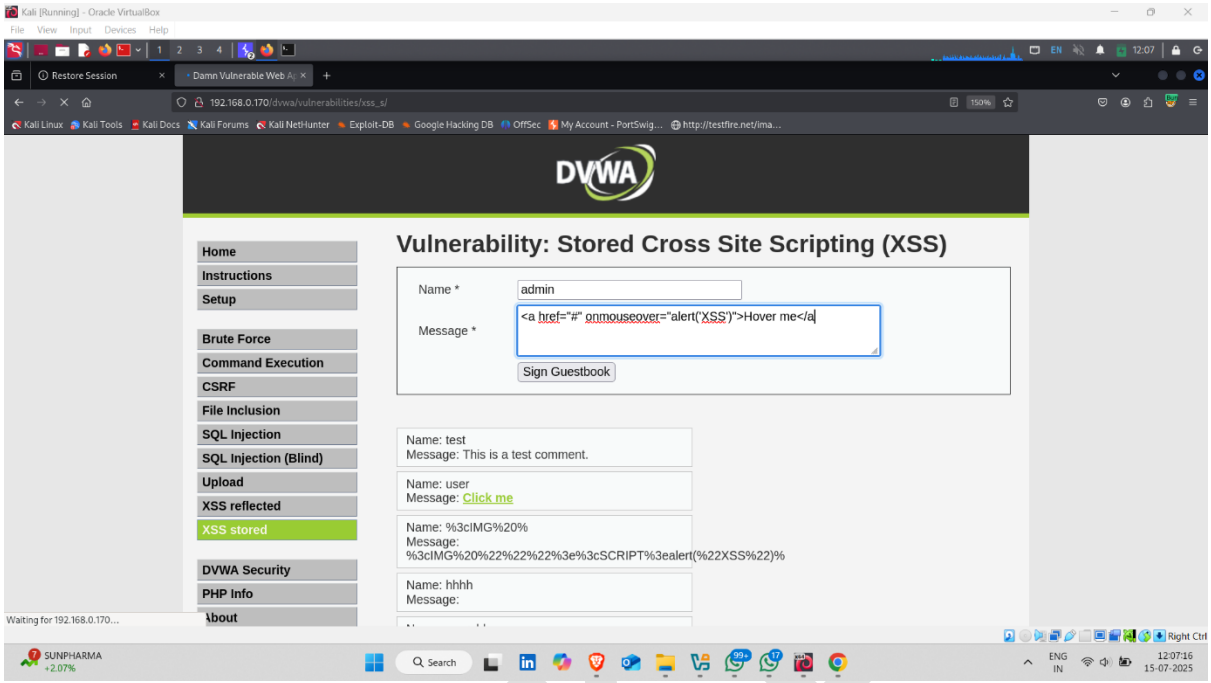
- Pop up appears , it means there is cross site scripting vulnerability available



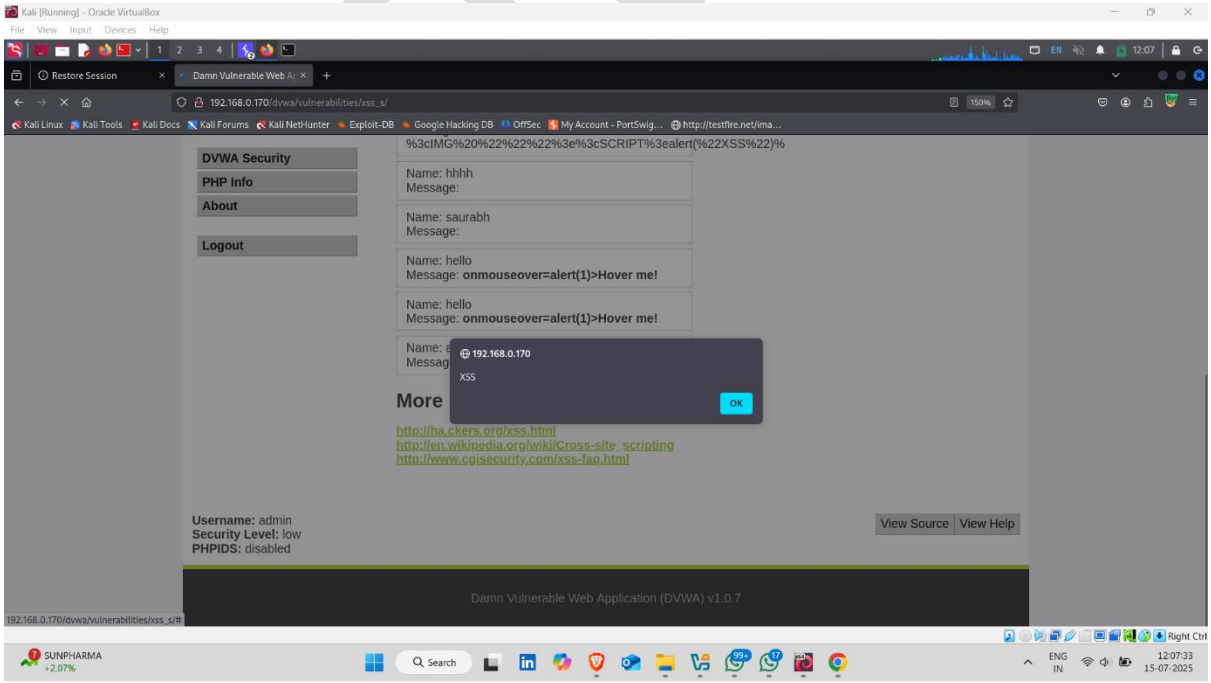
- Now click on Cross site scripting stored (XSS Stored)



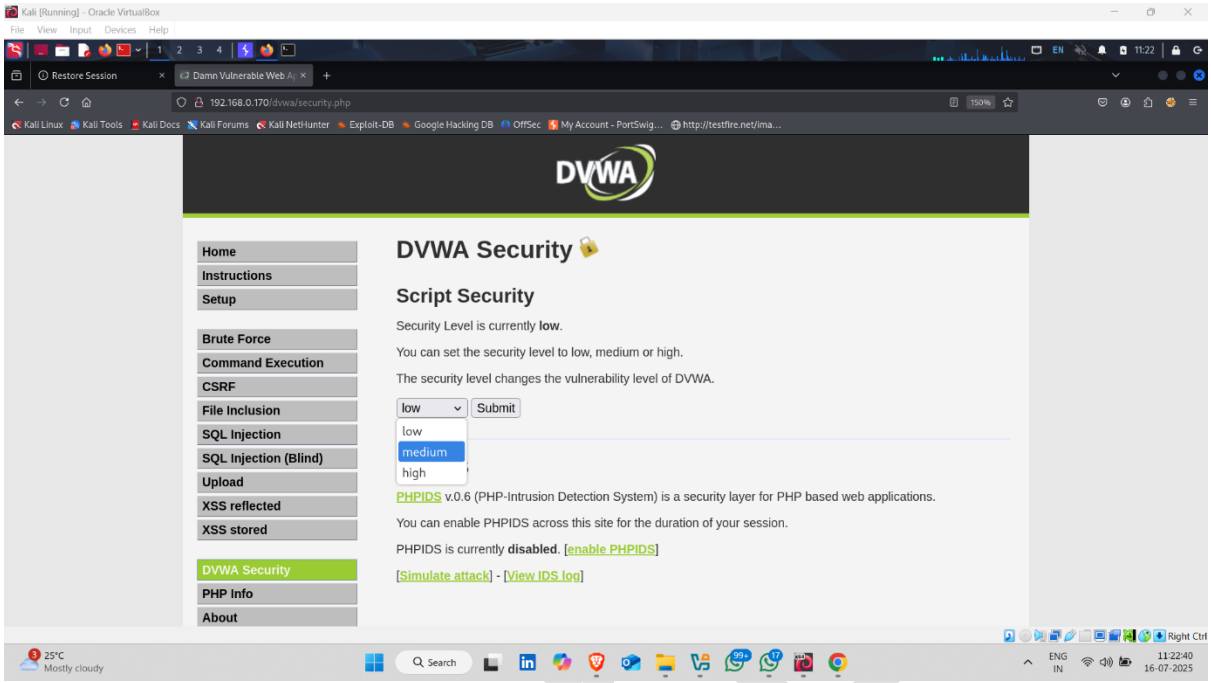
- Enter string in name section and enter script in message box and then click on **sign guestbook**



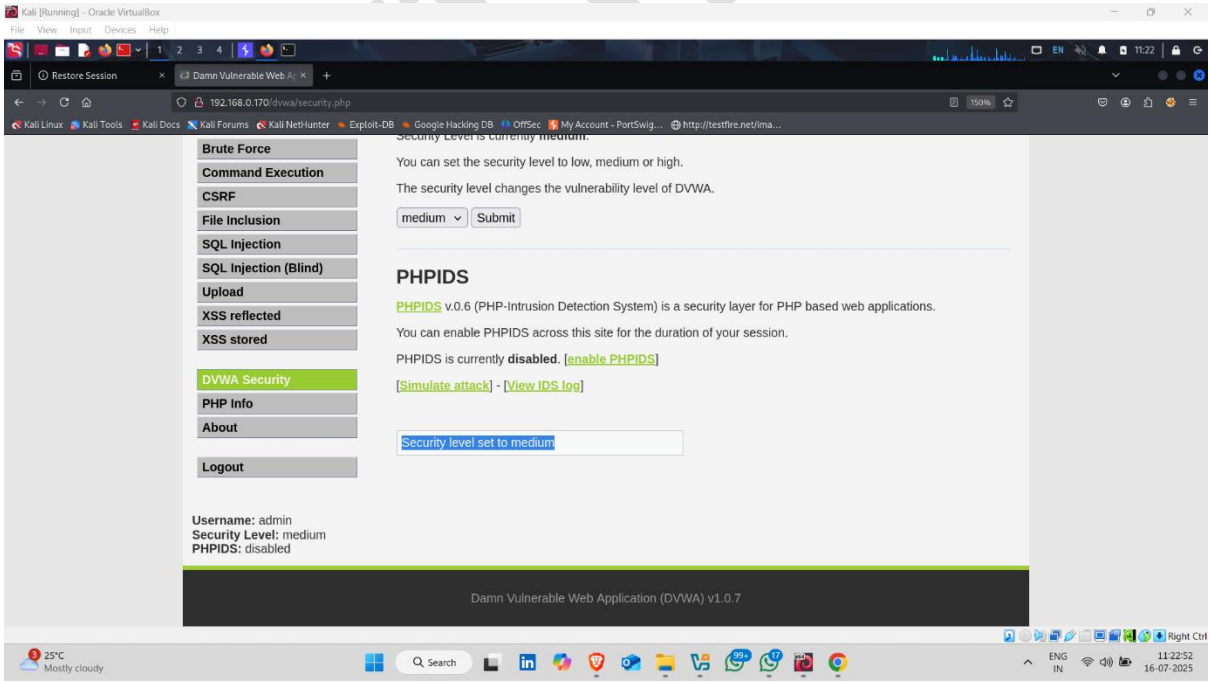
- Pop up appears ✅ 👇



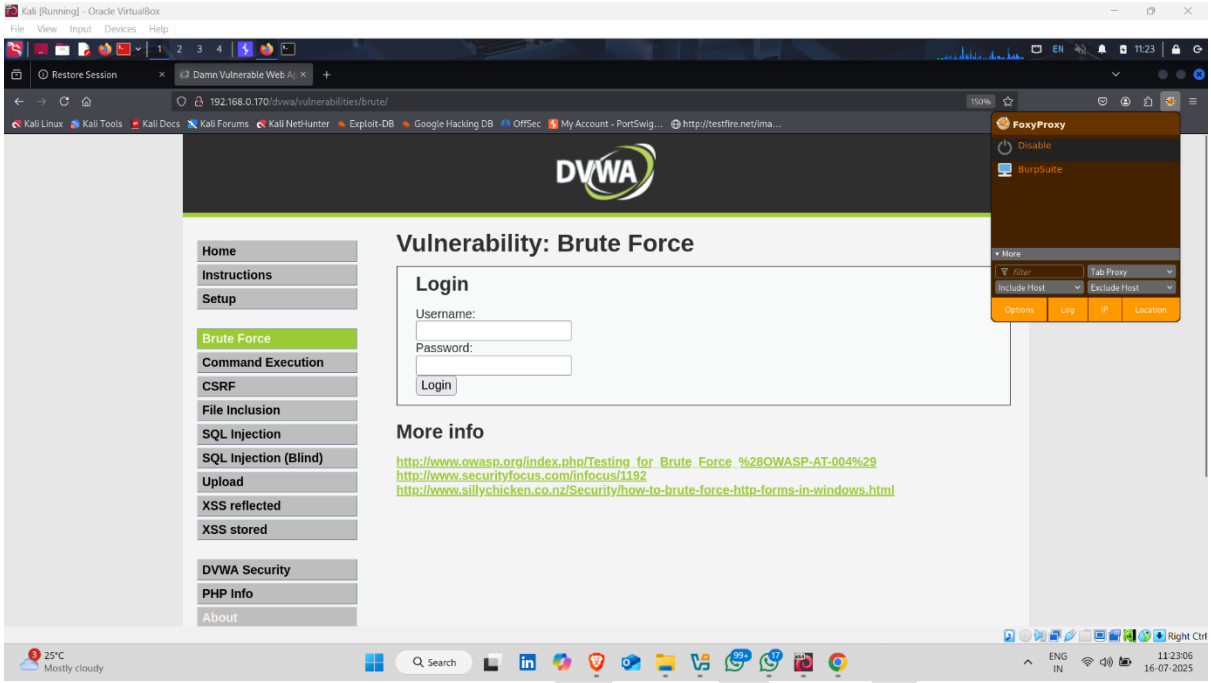
Security Level -: Medium



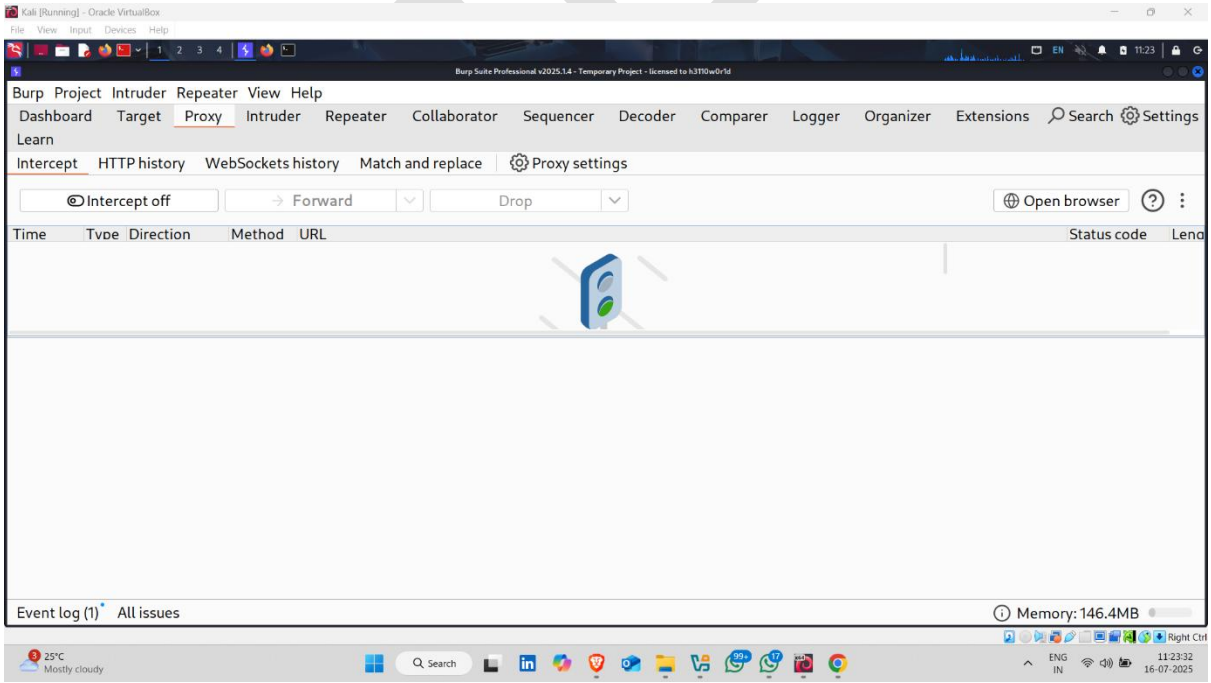
- Security Level Set 👇✅



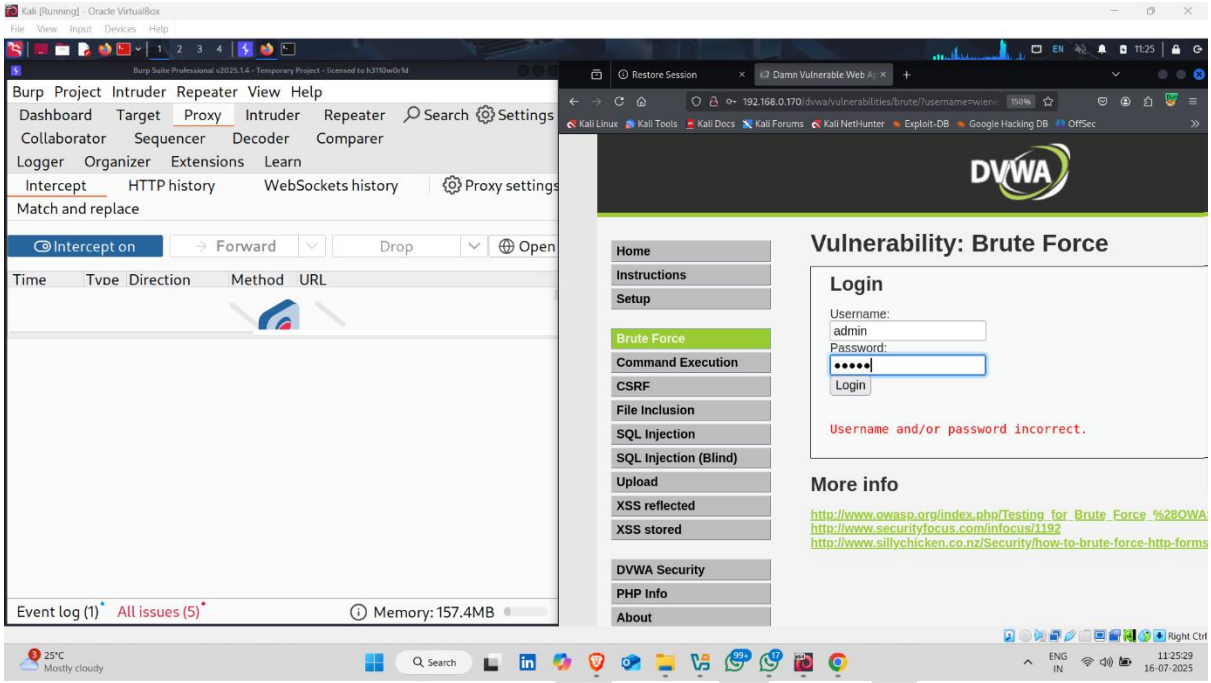
- Set up proxy



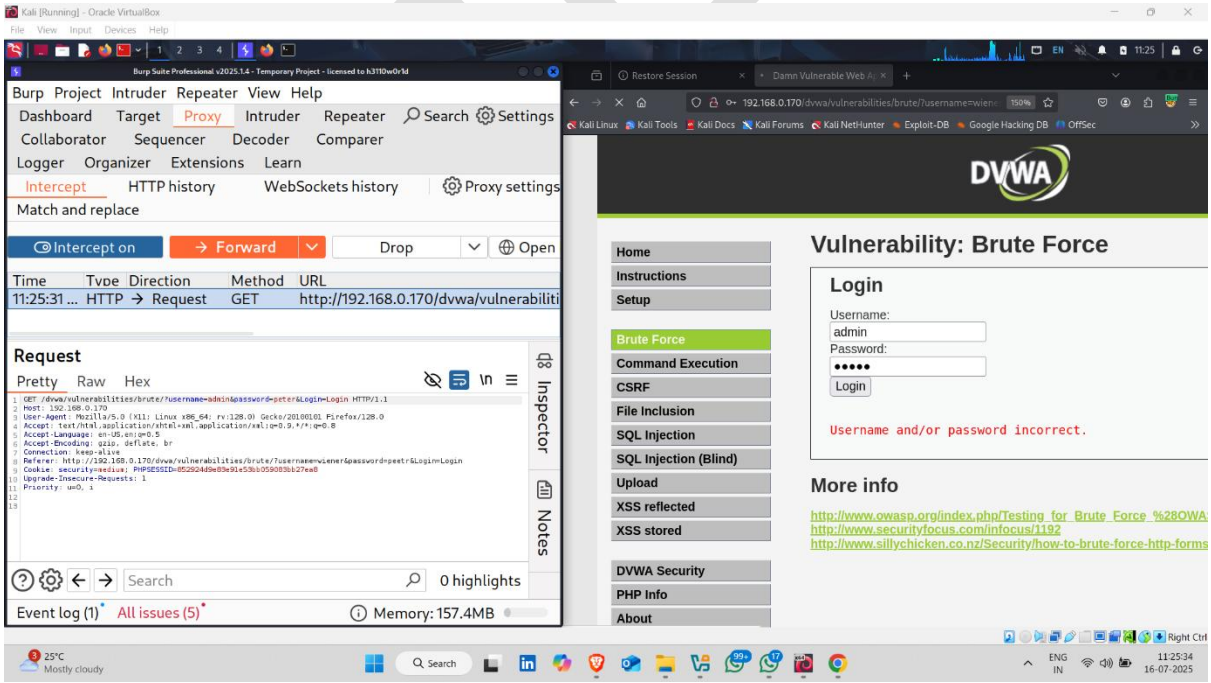
- Turn on Interception



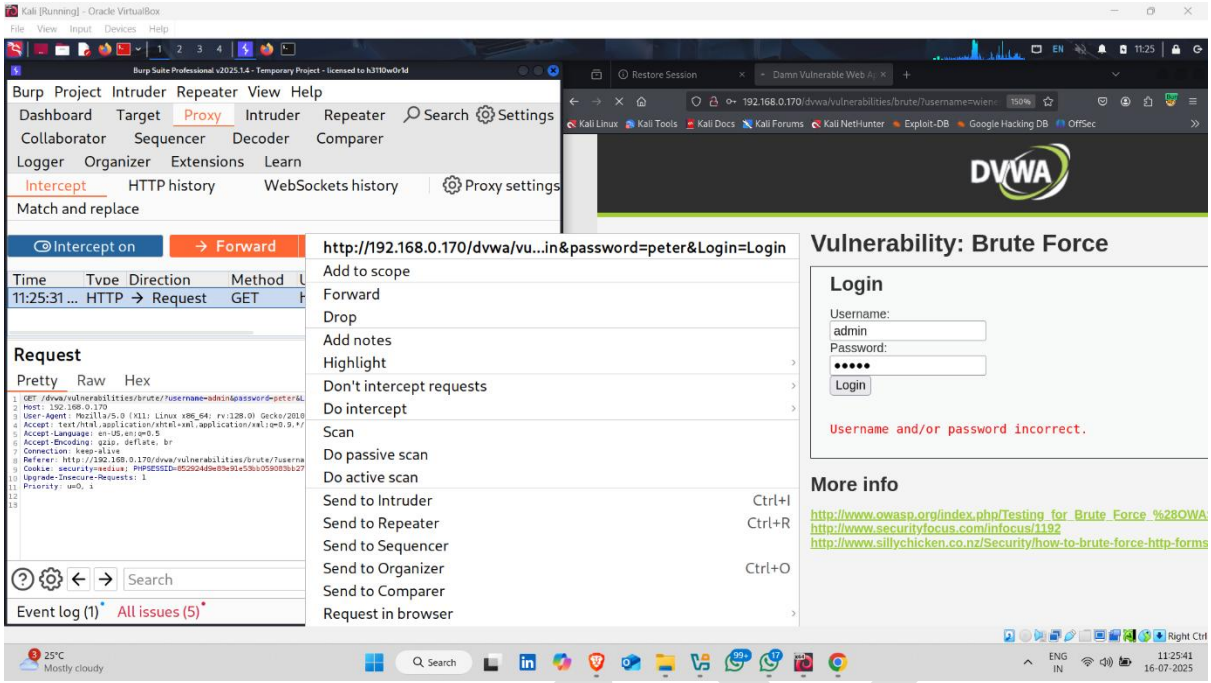
- Enter admin as a username and enter random password



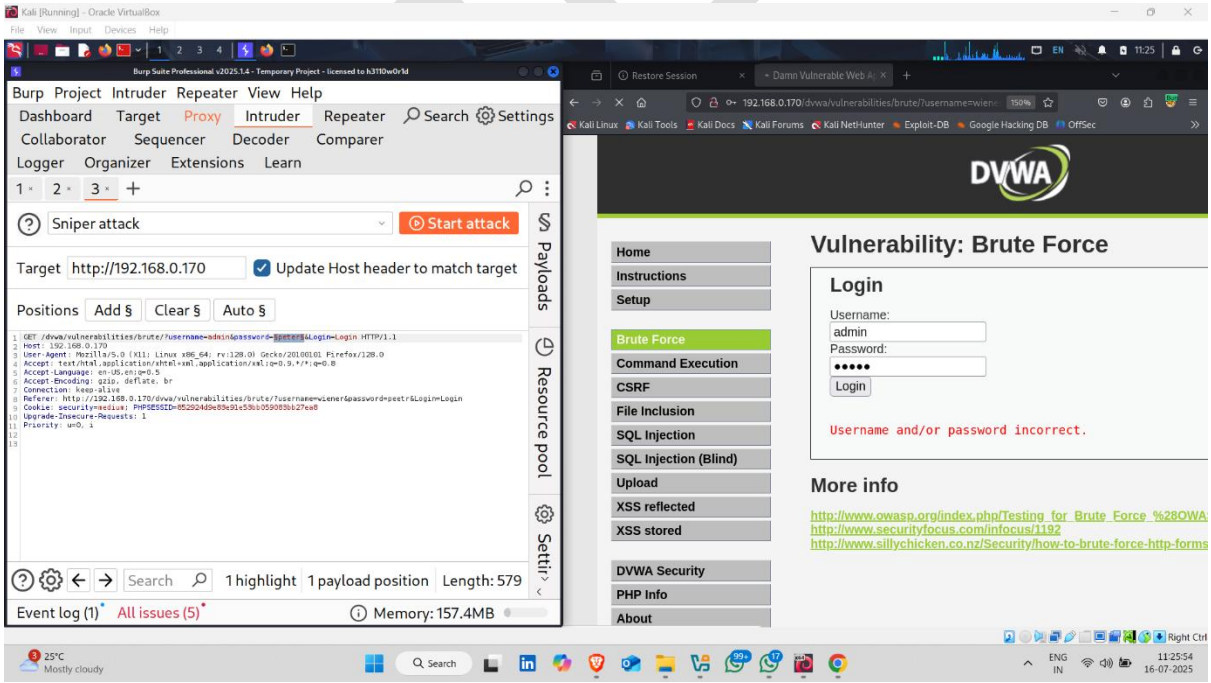
- Request intercept



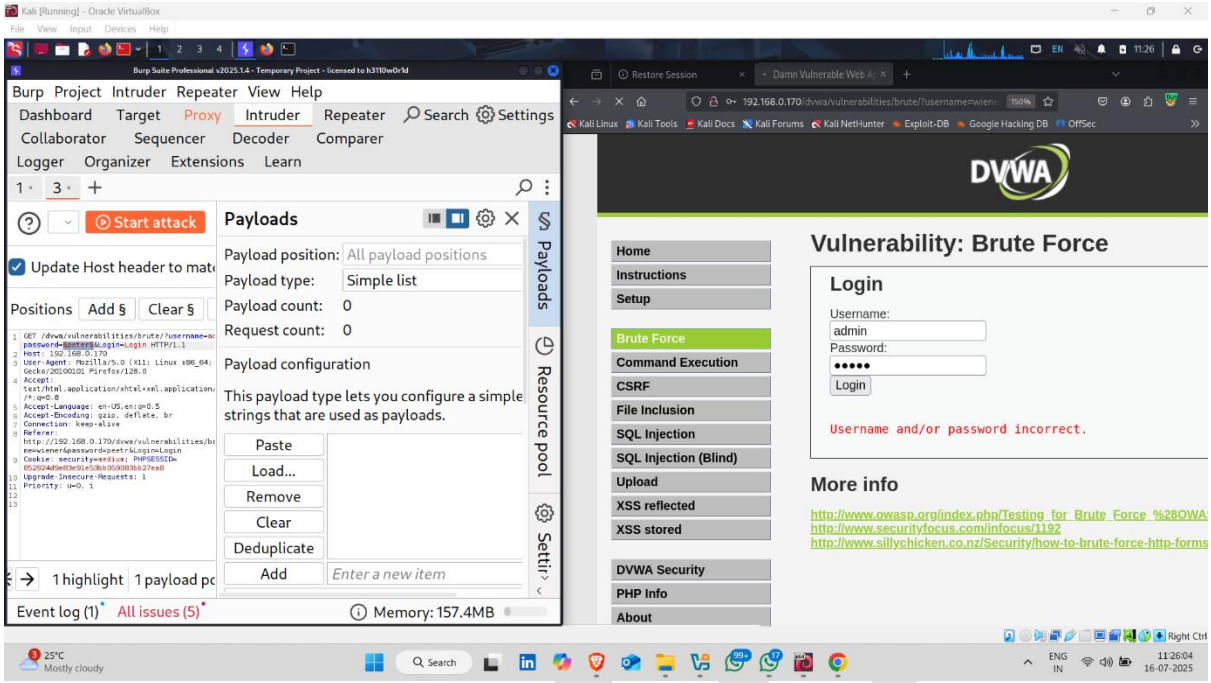
- Right click on request and send it to the **intruder**



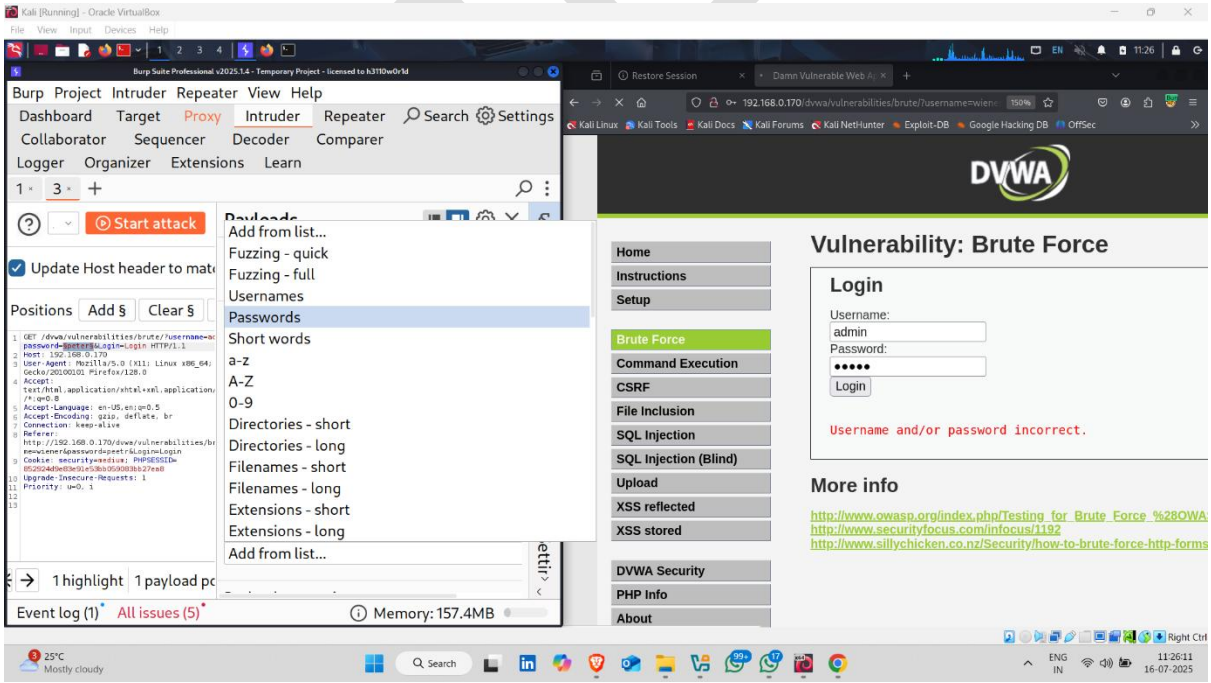
- Set payload position to password input



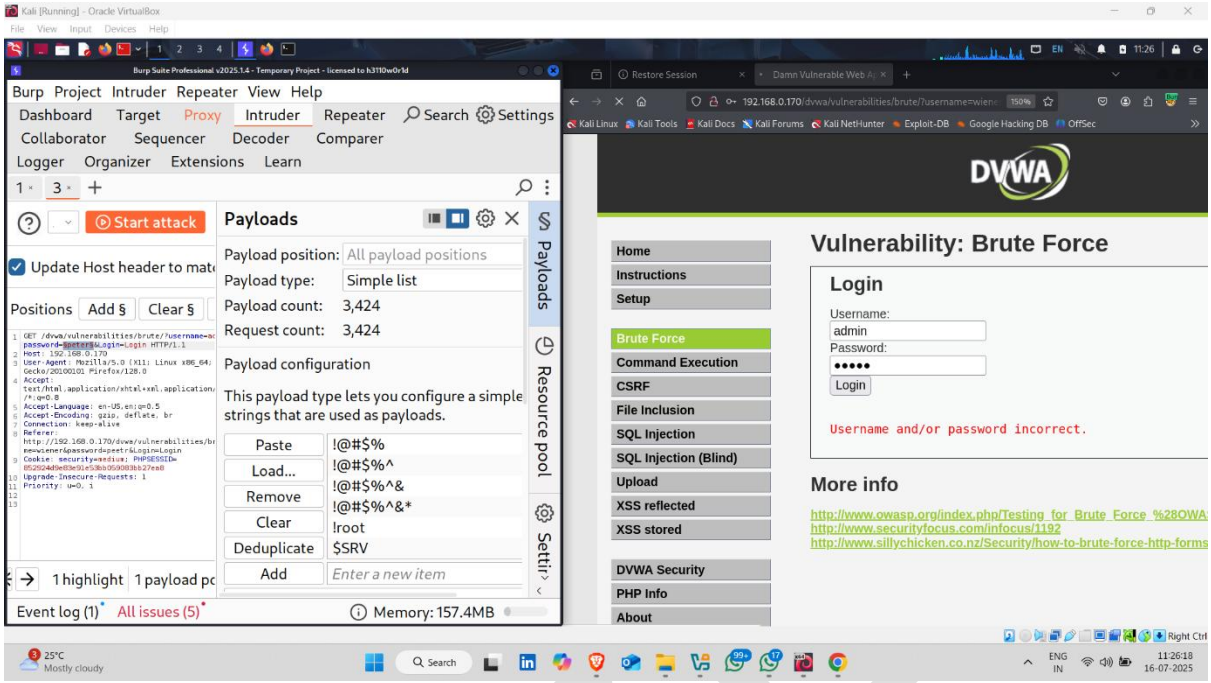
- Click on **payload tab**



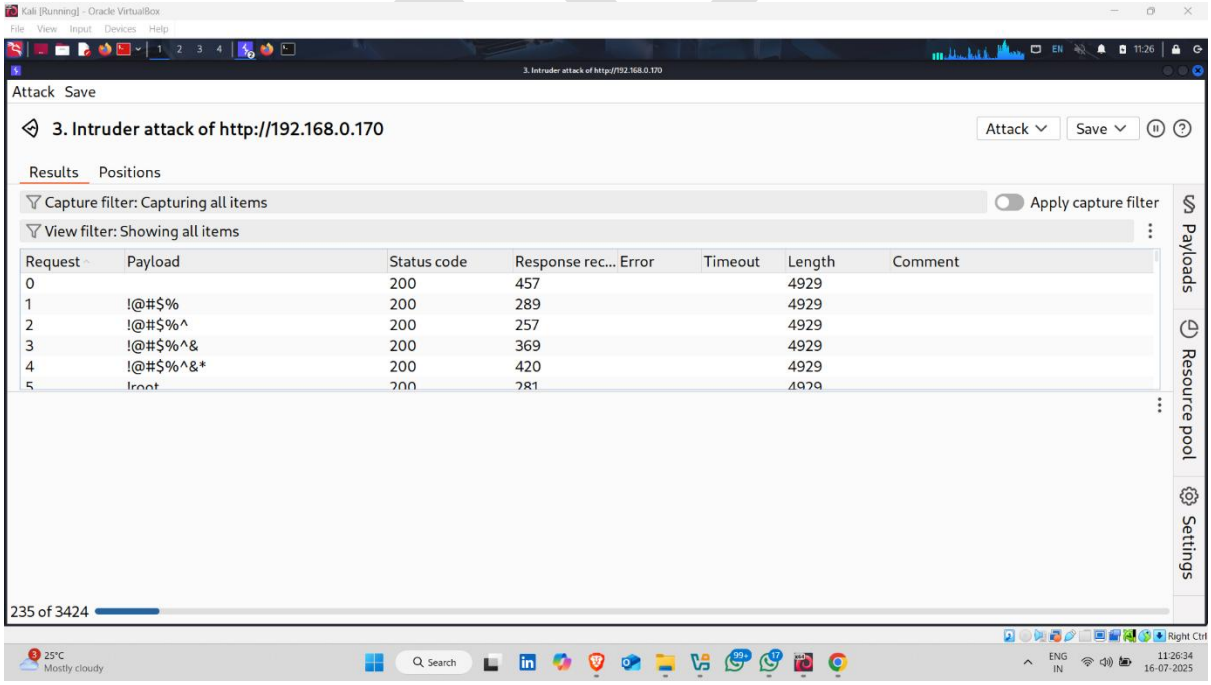
- Now click on **add from list** and select **password**



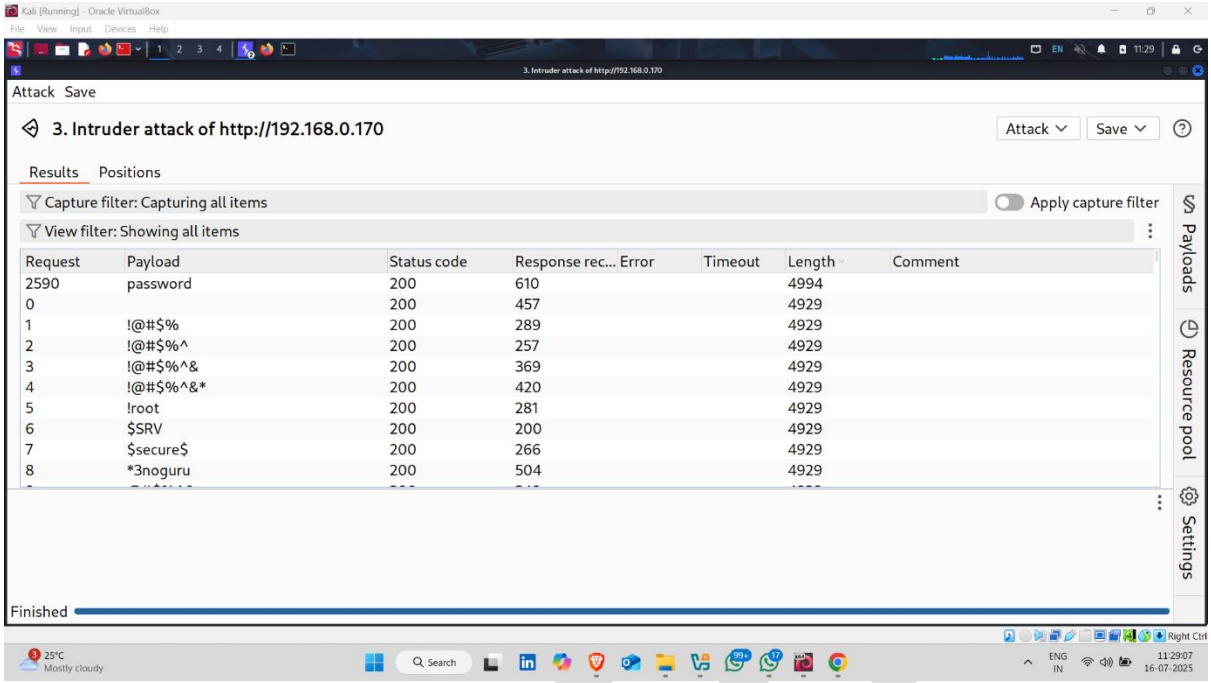
- Now start attack



- attack started



- **password found**



Perform Brute force attack using Hydra

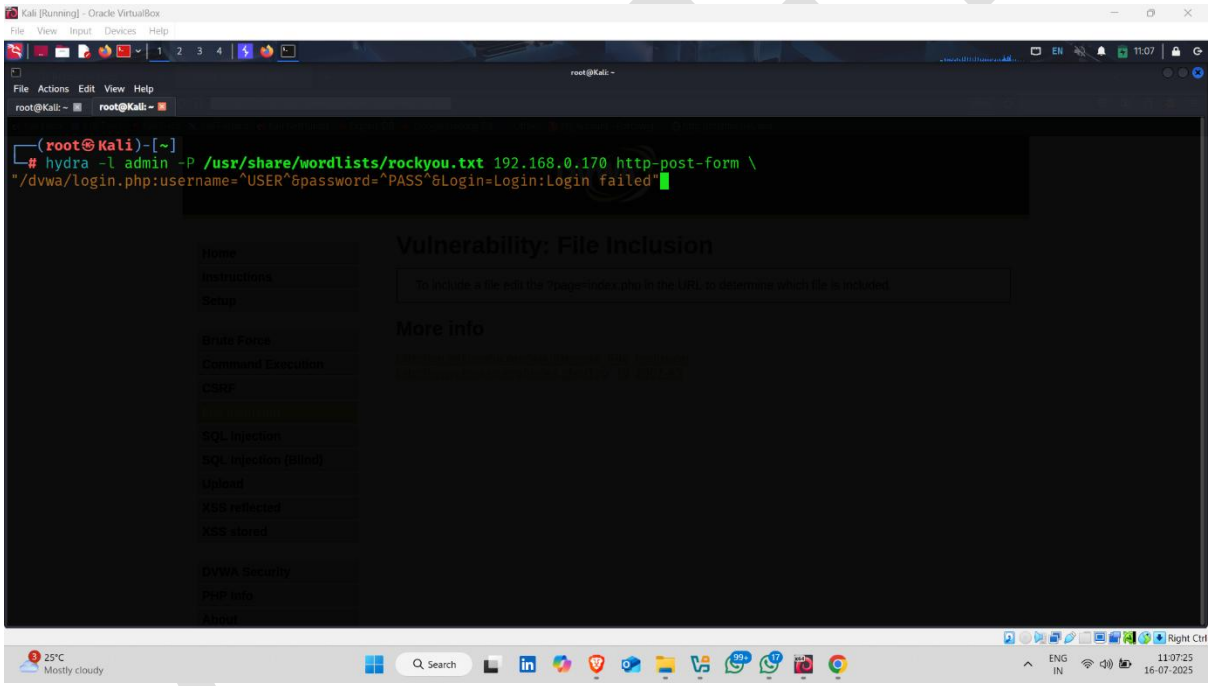
- enter following command for brute force using hydra

command -: hydra -l admin -P /usr/share/wordlists/rockyou.txt 192.168.0.170 http-post-form "/dvwa/login.php:username=USER&password=PASS&Login=Login:Login failed"

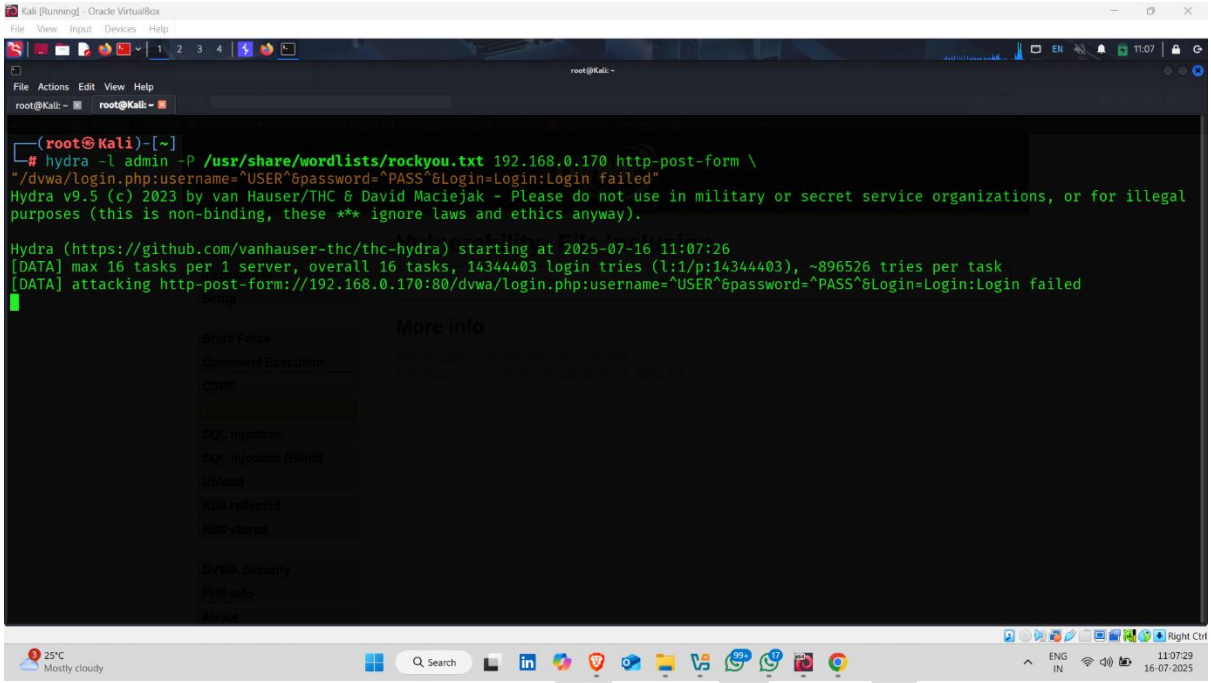
Explanation -:

Part	**Description**
hydra	Hydra is a fast and flexible password-cracking tool used to perform brute force attacks.
-l admin	Specifies the login username (admin).

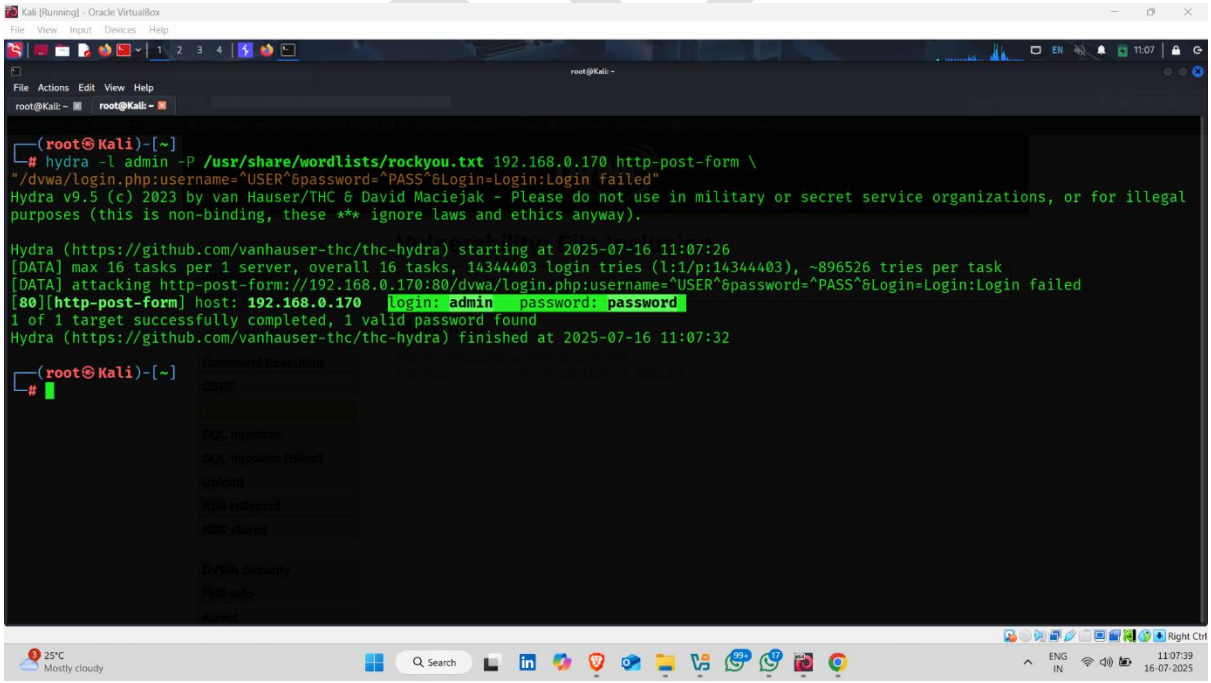
Part	Description
-P /usr/share/wordlists/rockyou.txt	Uses the rockyou.txt wordlist to test passwords. This is a common and popular list of passwords.
192.168.0.170	Target IP address where DVWA is hosted (likely in your local VM or lab).
http-post-form	Tells Hydra that it's targeting a web form using the HTTP POST method.



- attack started



- password found ✅ 👇



Perform Brute Force Using Medusa

- enter following command

command -: medusa -h 192.168.0.170 -u admin -P /usr/share/wordlists/rockyou.txt \

-M http -m DIR:/dvwa/login.php -m FORM:"username=USER&password=PASS&Login=Login" \

-m DENY:"Login failed"

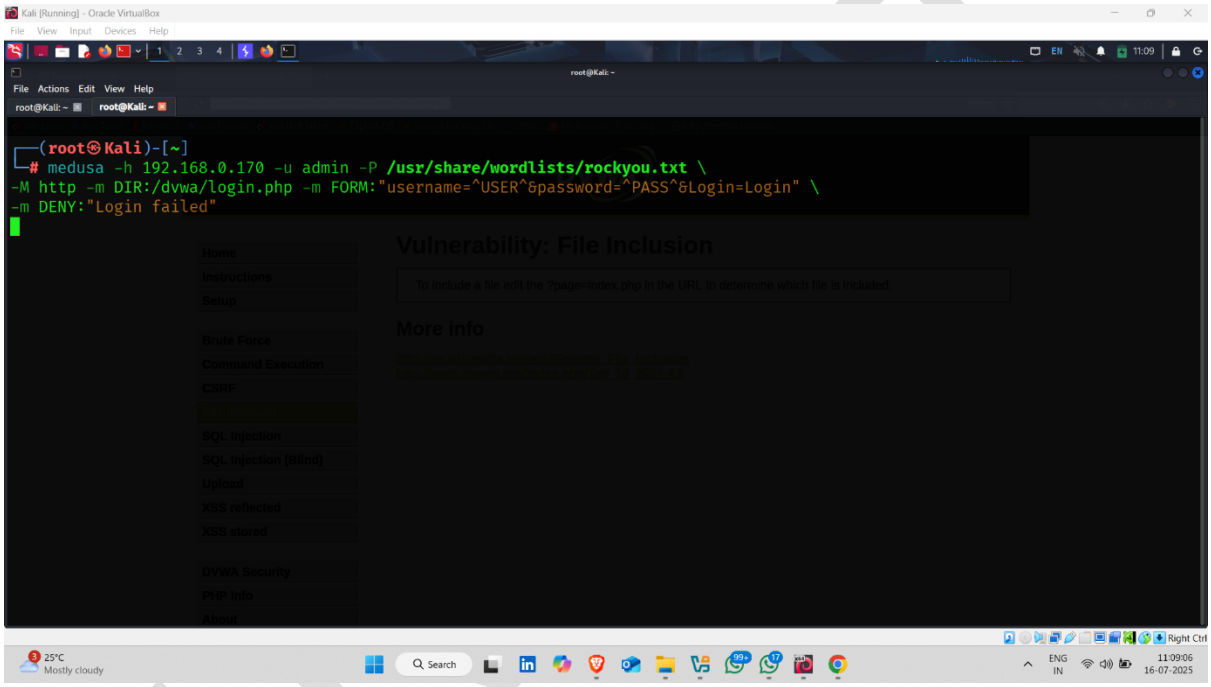
Explanation -:

Part	**Meaning**
medusa	Launches the Medusa login brute-forcing tool.
-h 192.168.0.170	Specifies the **target host IP address** (where DVWA is running).
-u admin	Sets the **username** to admin.
-P /usr/share/wordlists/rockyou.txt	Provides the **password wordlist** (RockYou).

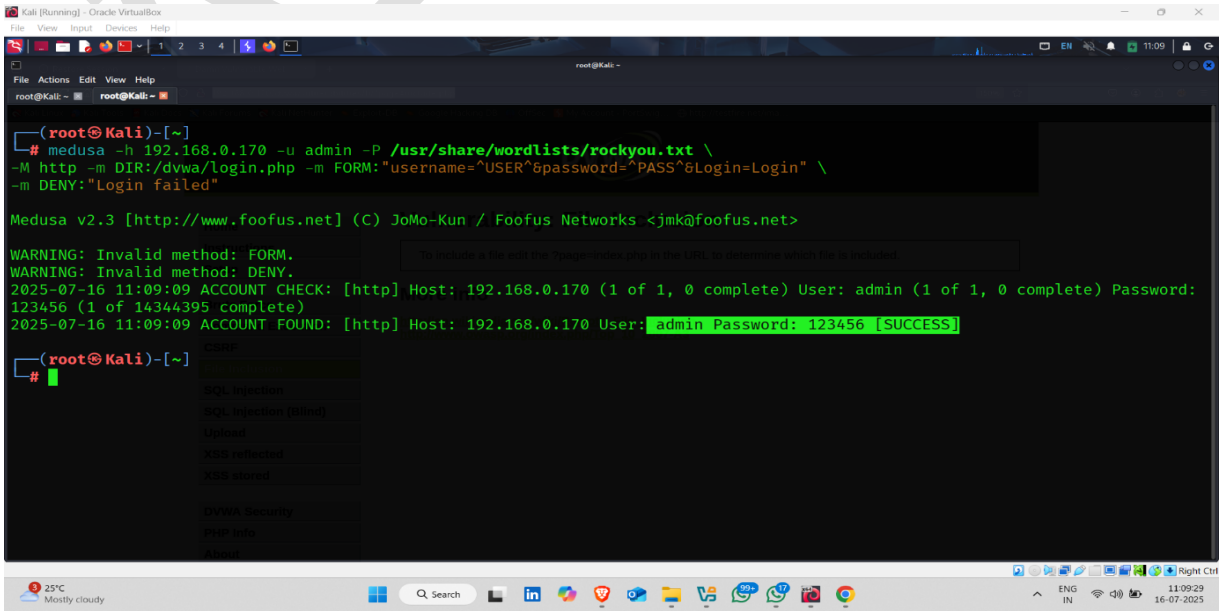
🔍 **Module-Specific Options:**

Option	**Description**
-M http	Specifies the **HTTP module** for web-based login brute-force.
-m DIR:/dvwa/login.php	Sets the **login URL path**.
-m FORM:"username=USER&password=PASS&Login=Login"	Defines the **POST parameters** with placeholders USER and PASS.

Option	Description
-m DENY:"Login failed"	Tells Medusa to detect **failed login attempts** by matching the "**Login failed**" response string. If not found, it assumes login was successful.

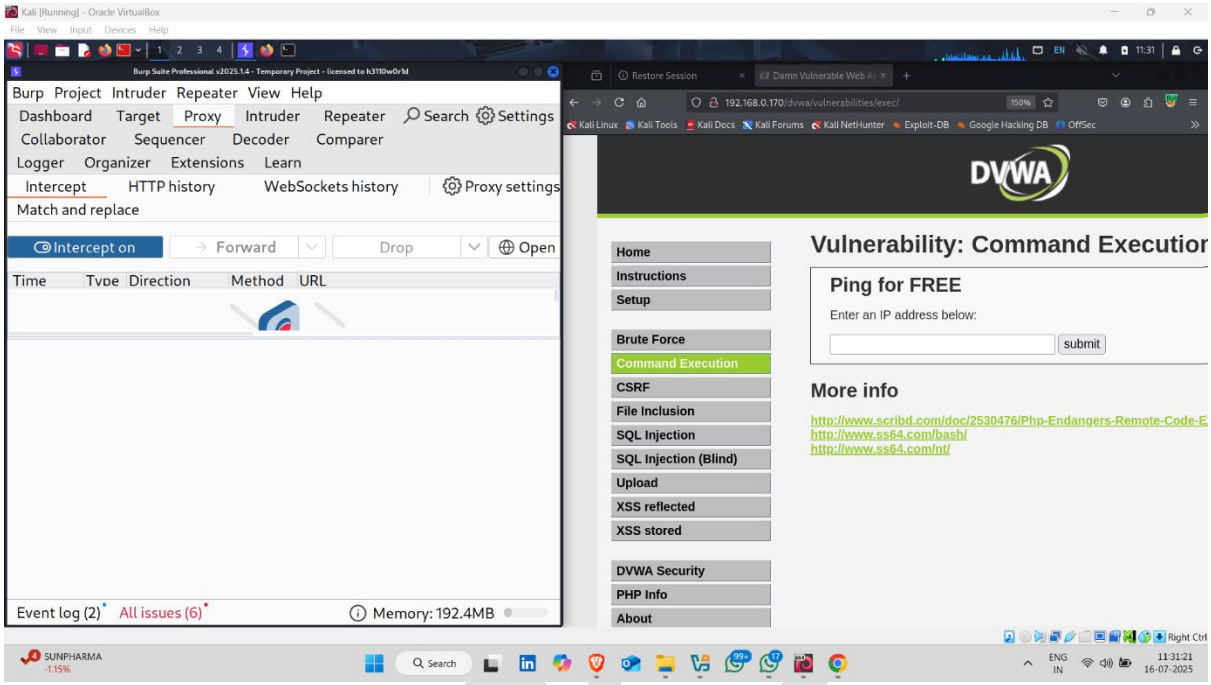


- Password Found 👇✅



Task-2 –Command Execution

- Click on command Execution tab



- Open terminal and type following command

Command -: curl -X POST "http://192.168.0.170/dvwa/vulnerabilities/exec/" \

--cookie "PHPSESSID=852924d9e83e91e53bb059083bb27ea8; security=medium" \

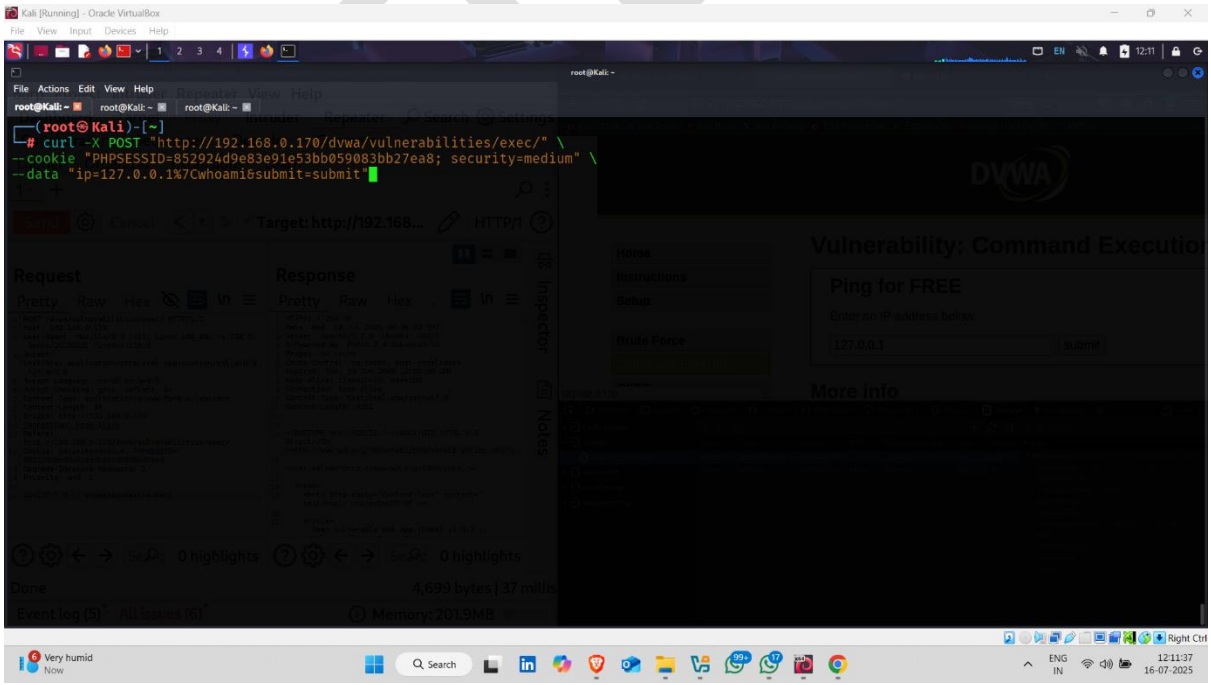
--data "ip=127.0.0.1%7Cwhoami&submit=submit"

Explanation-:

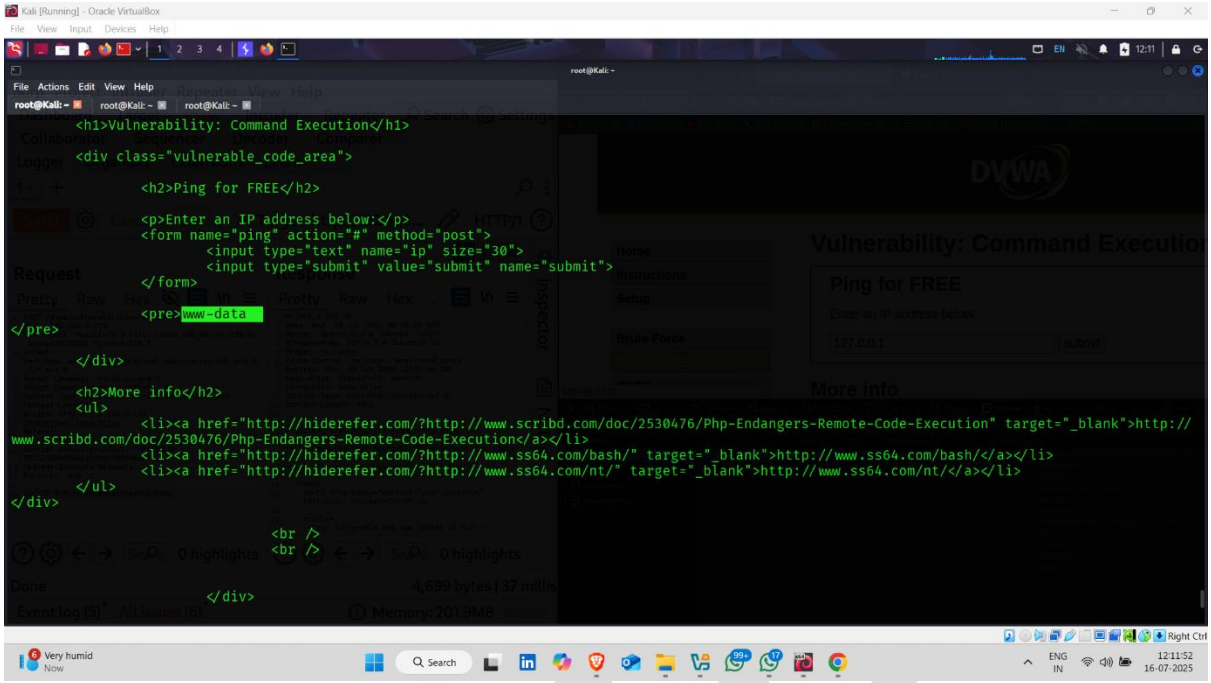
Part	**Description**
curl	Command-line tool to send HTTP requests.
-X POST	Specifies the **POST method** (used to submit form data).

Part	Description
"http://192.168.0.170/dvwa/vulnerabilities/exec/"	Target URL of the **Command Execution** vulnerability in DVWA.
--cookie	Sends **session cookie** to stay authenticated:

- PHPSESSID=852924... is your active login session.
- security=medium is the DVWA security level. |
  | --data | Sends POST form fields:
- ip=127.0.0.1%7Cwhoami → The %7C is **URL-encoded pipe symbol (|)**, used to **inject a shell command** (whoami).
- submit=submit → Mimics clicking the submit button. |



- retrieve data



- Now use next command

Command -: curl -X POST "http://192.168.0.170/dvwa/vulnerabilities/exec/" \

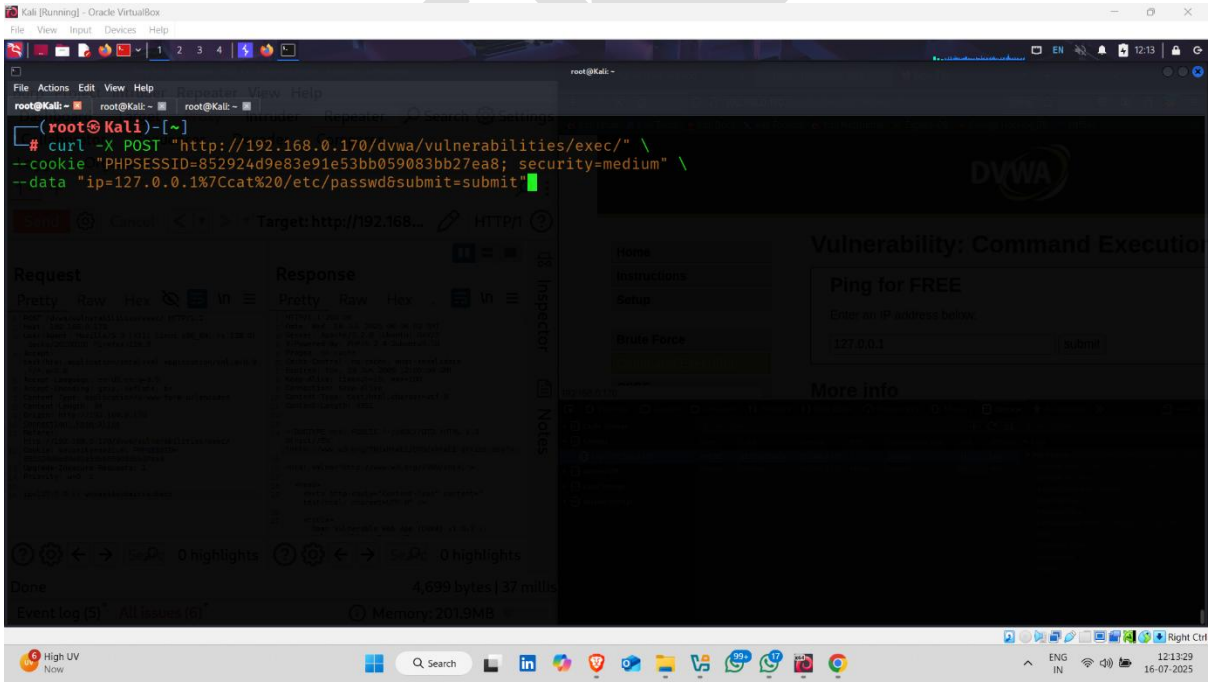
--cookie "PHPSESSID=852924d9e83e91e53bb059083bb27ea8; security=medium" \

--data "ip=127.0.0.1%7Ccat%20/etc/passwd&submit=submit"

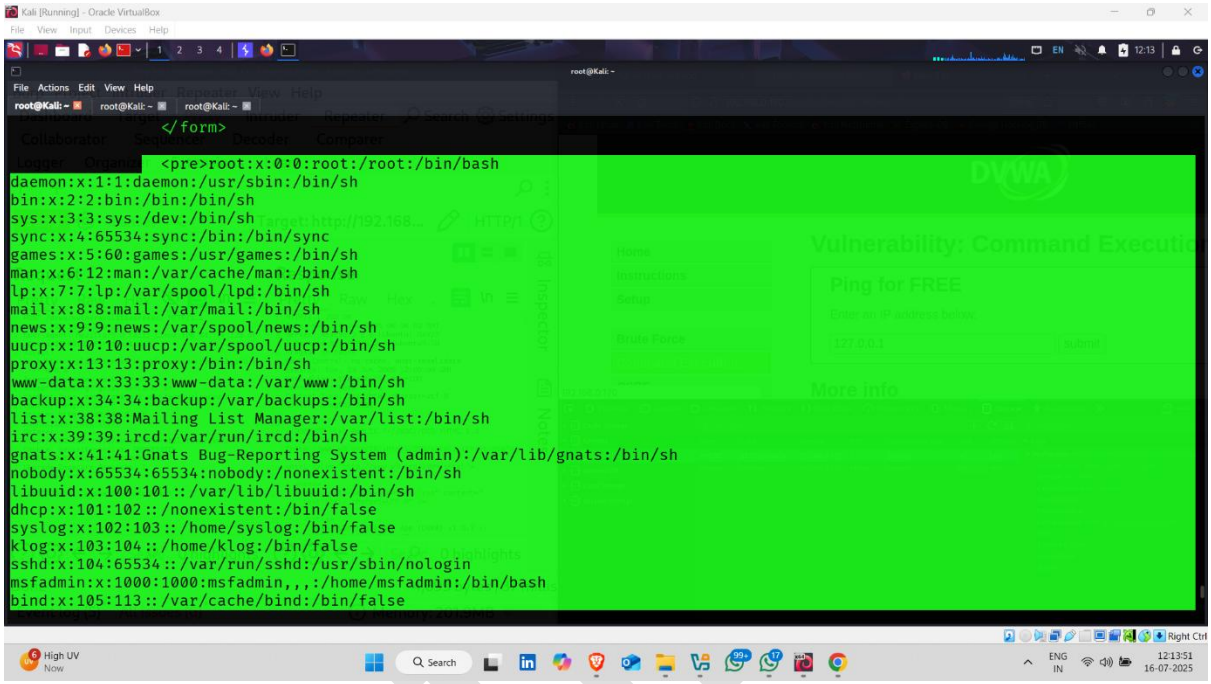
Explanation -:

Part	**Purpose**
curl -X POST	Sends a **POST** request (mimicking a form submission).
http://192.168.0.170/dvwa/vulnerabilities/exec/	The **Command Injection vulnerable endpoint** in DVWA.

Part	Purpose
--cookie ...	Supplies the **session cookie** and **security level**.
--data "ip=127.0.0.1%7Ccat%20/etc/passwd&submit=submit"	The POST payload:
127.0.0.1%7Ccat%20/etc/passwd	This gets URL-decoded to:
`127.0.0.1	cat /etc/passwd`
➡️ The pipe (`)	lets you inject the cat /etc/passwd` command into the backend.

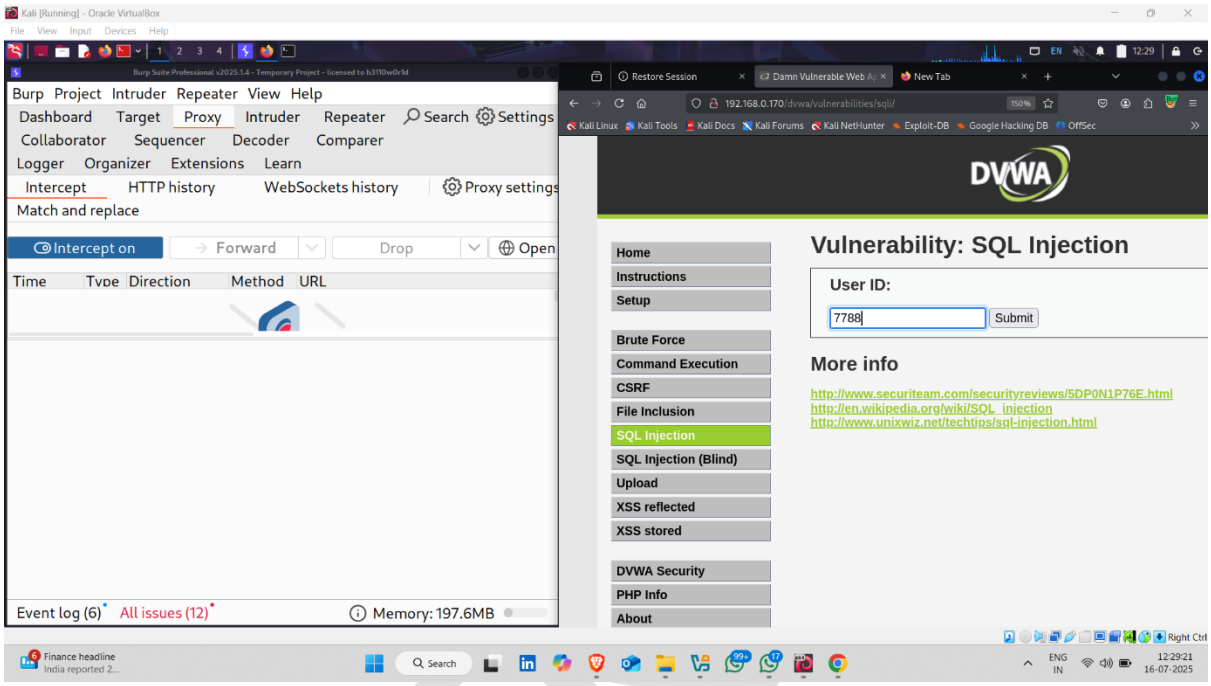


- Result ✅ 👇
- This is the **output of the /etc/passwd file**, a key system file on Linux that contains **user account information** (not passwords — those are stored in /etc/shadow).

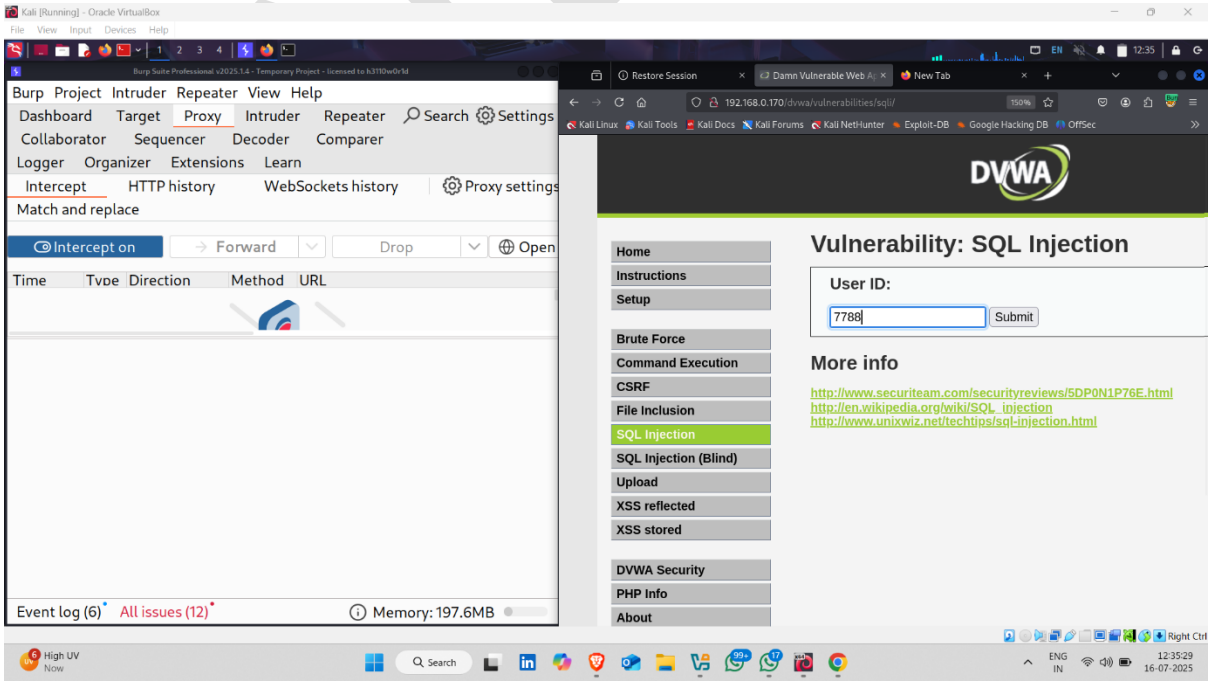


Task -3 – Sql injection

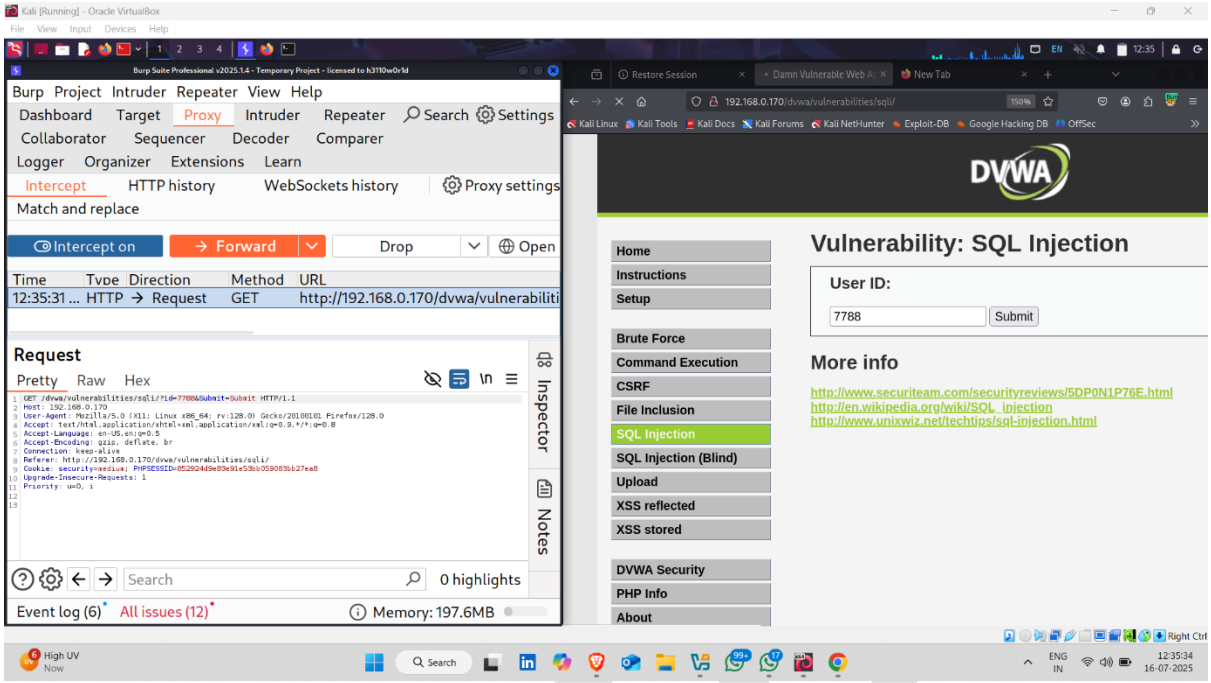
- Click on SQL Injection
- Enter a random **User Id**



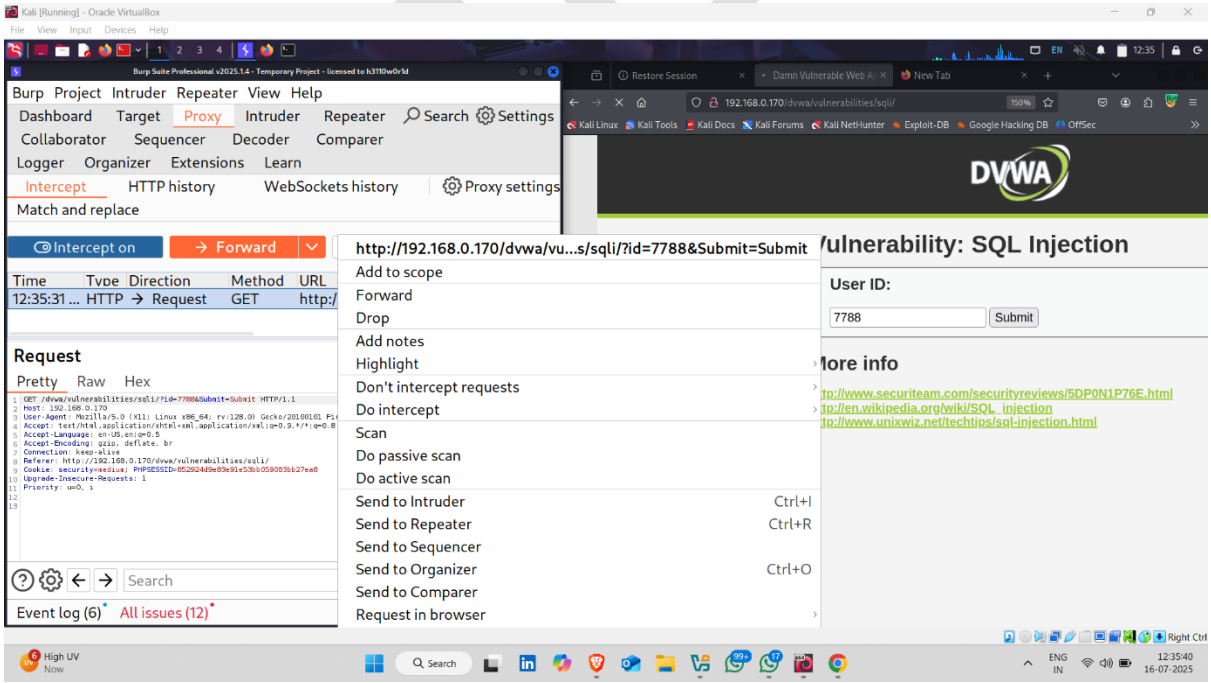
- Click on **submit**



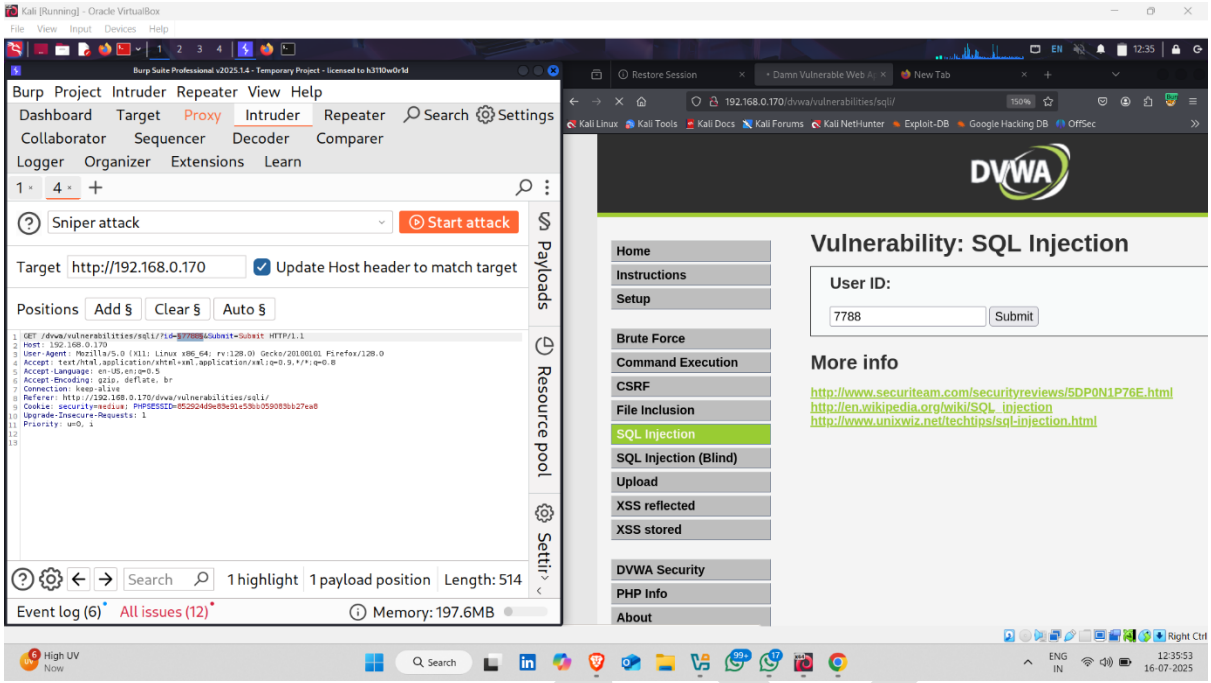
- Request intercepted



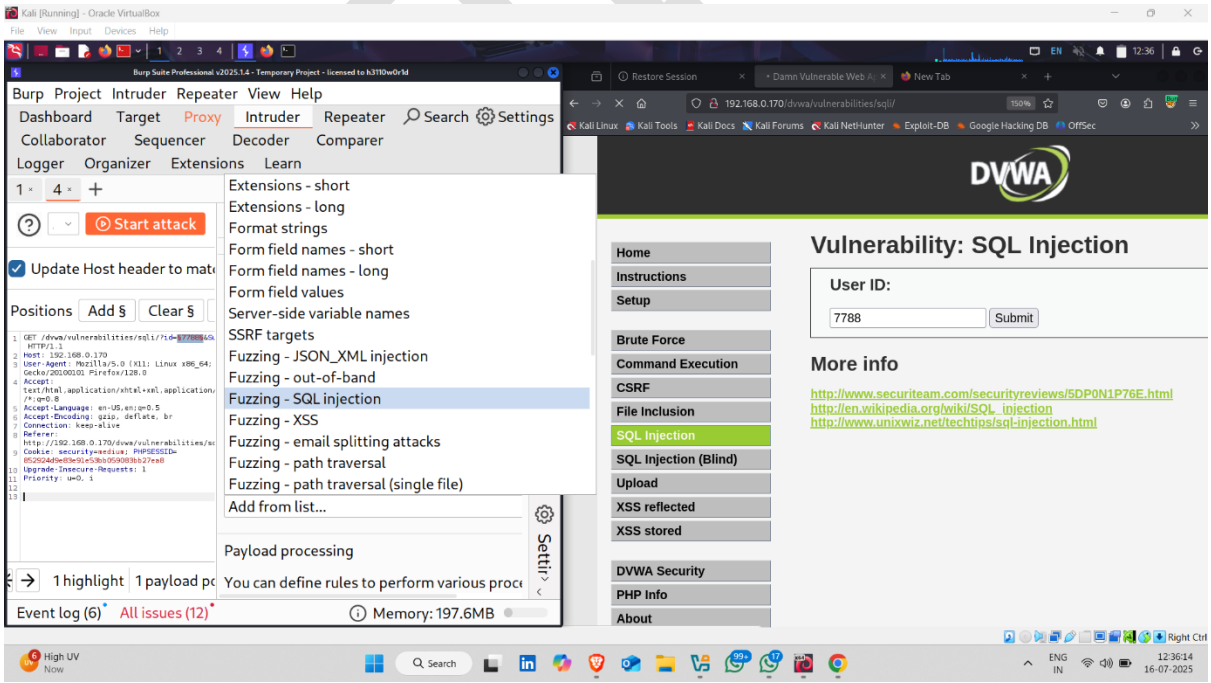
- Right click on request and send it to **intruder**



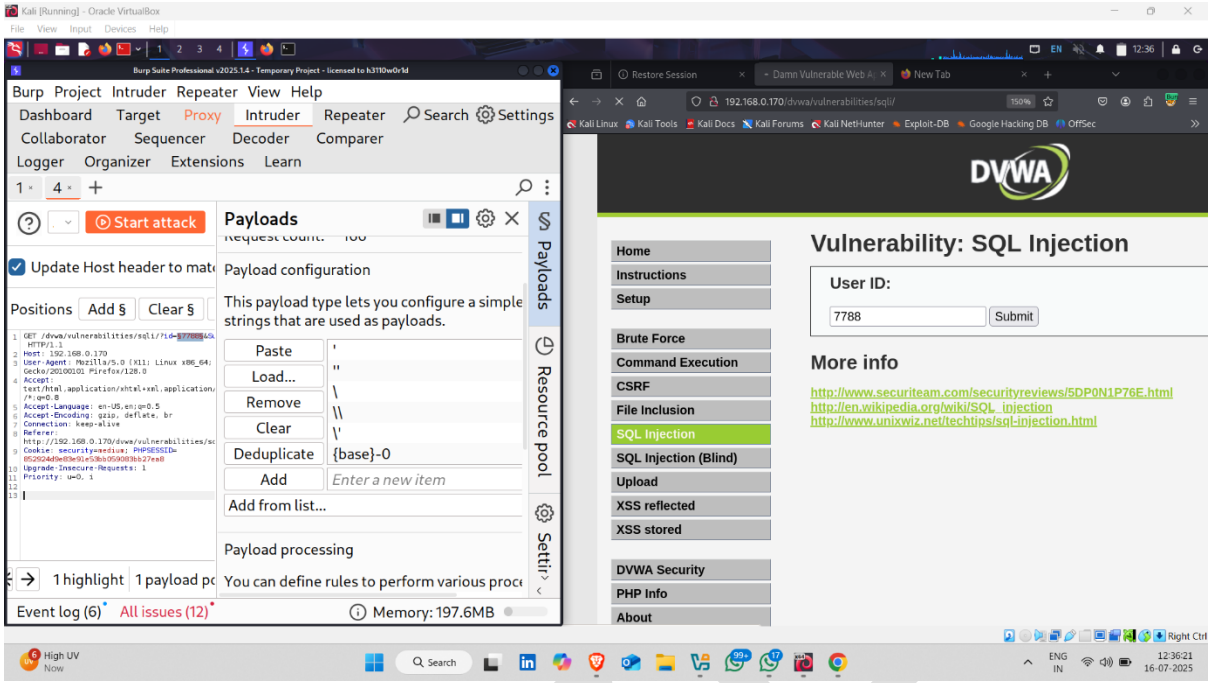
- In intruder , select payload position to **id section**



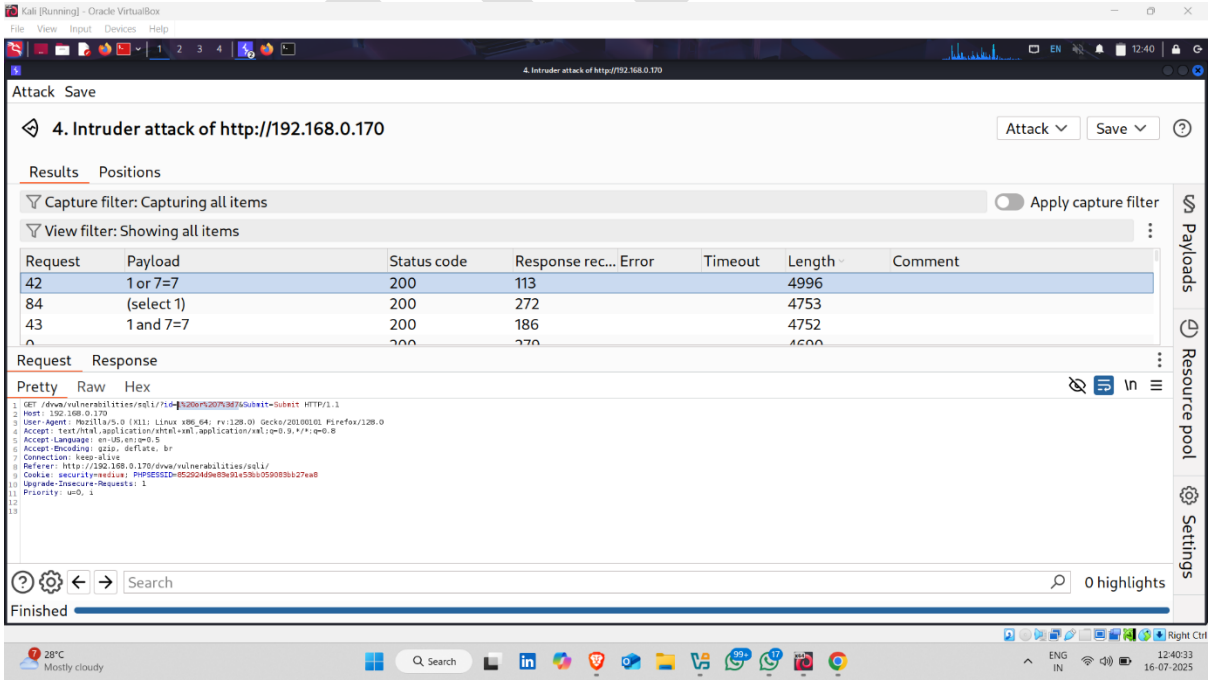
- Click on **payload** section , then click on **Add From List** and select **Fuzzing-SQL Injection**



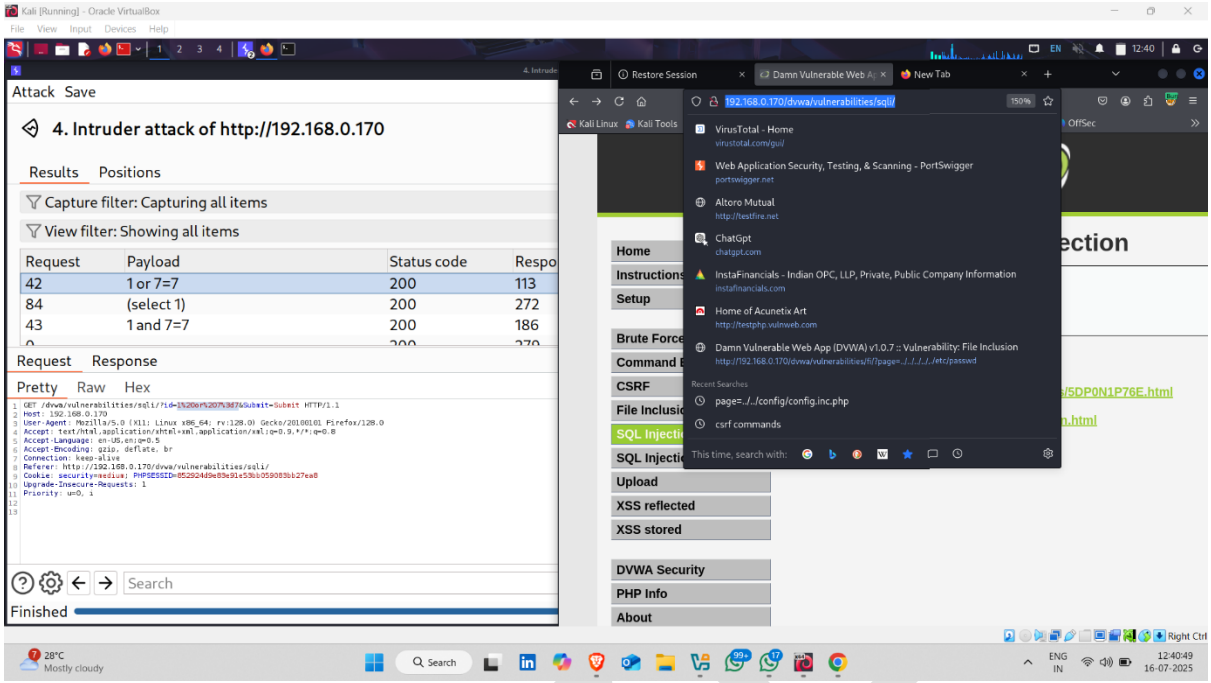
- Click on **start attack**



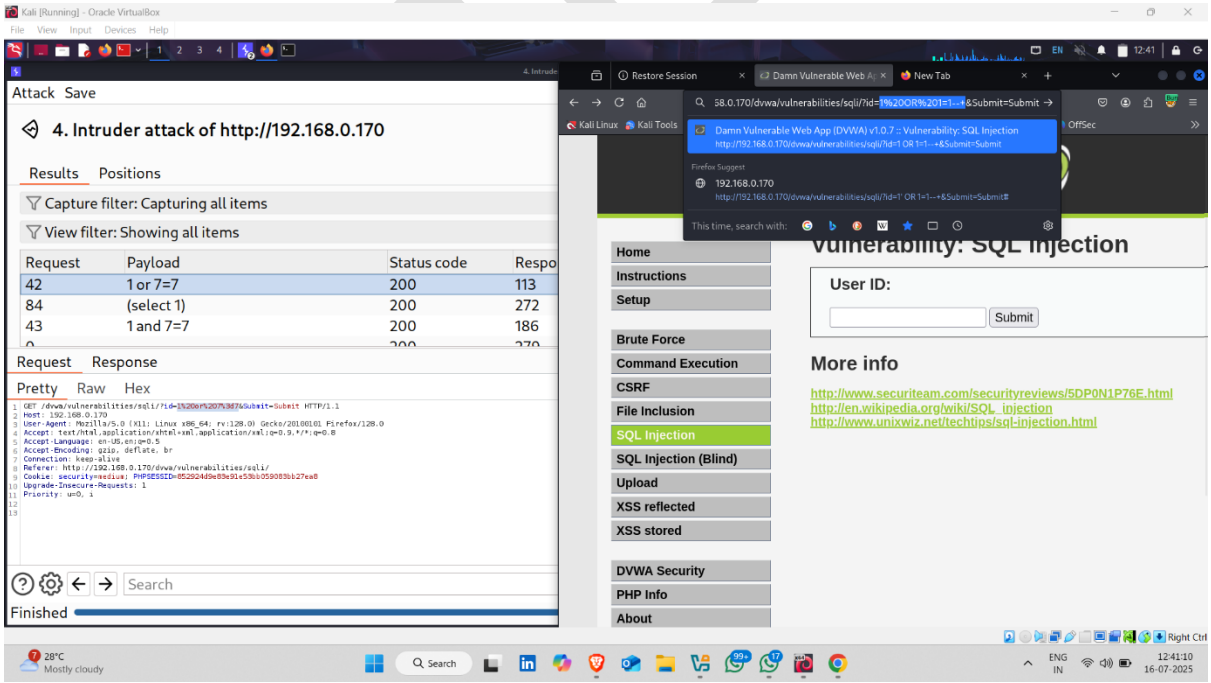
- Attack finished

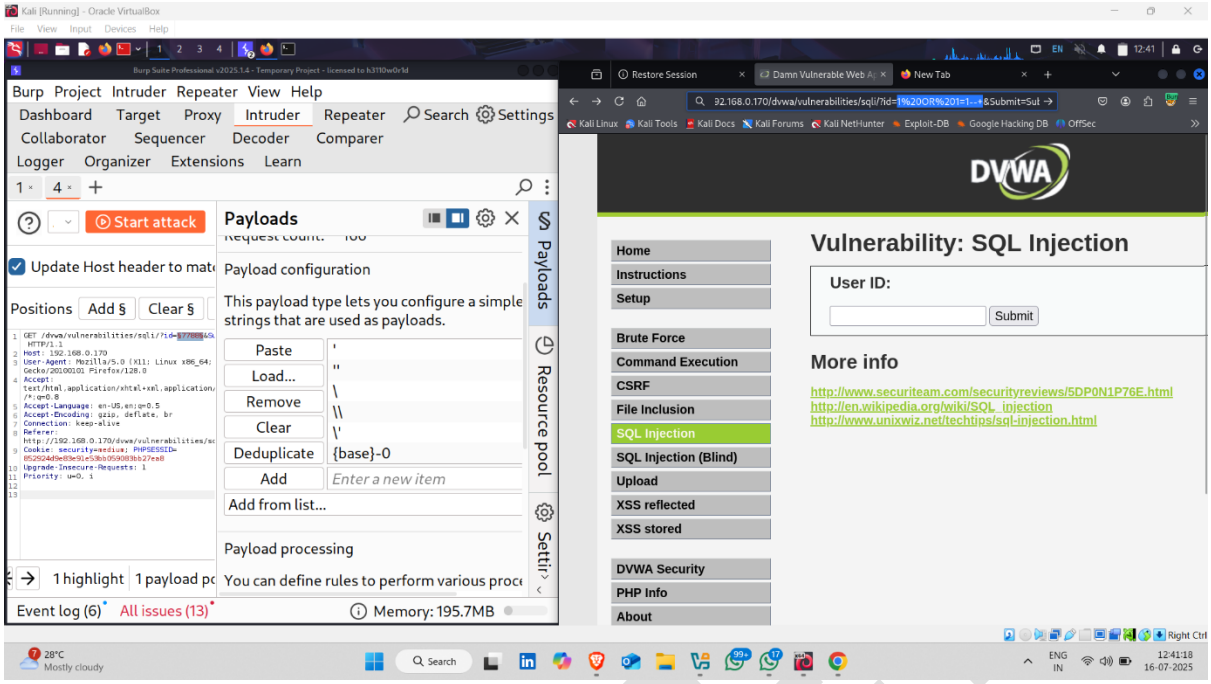


- Now go to DVWA, click on URL Section

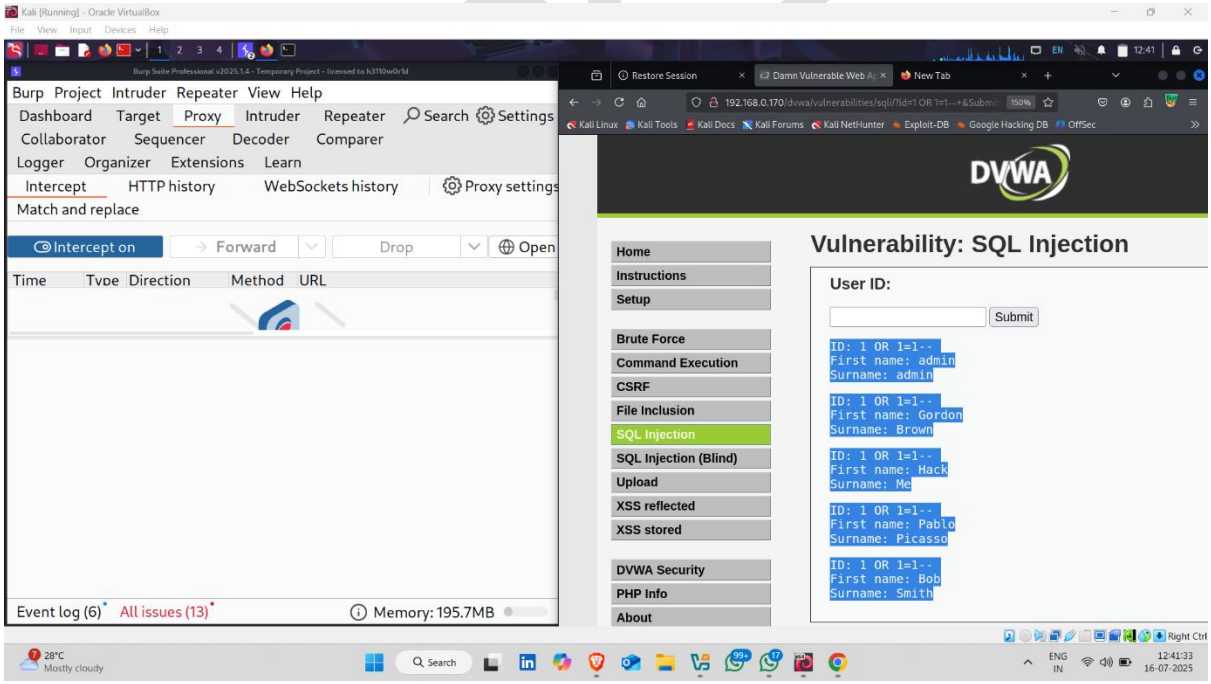


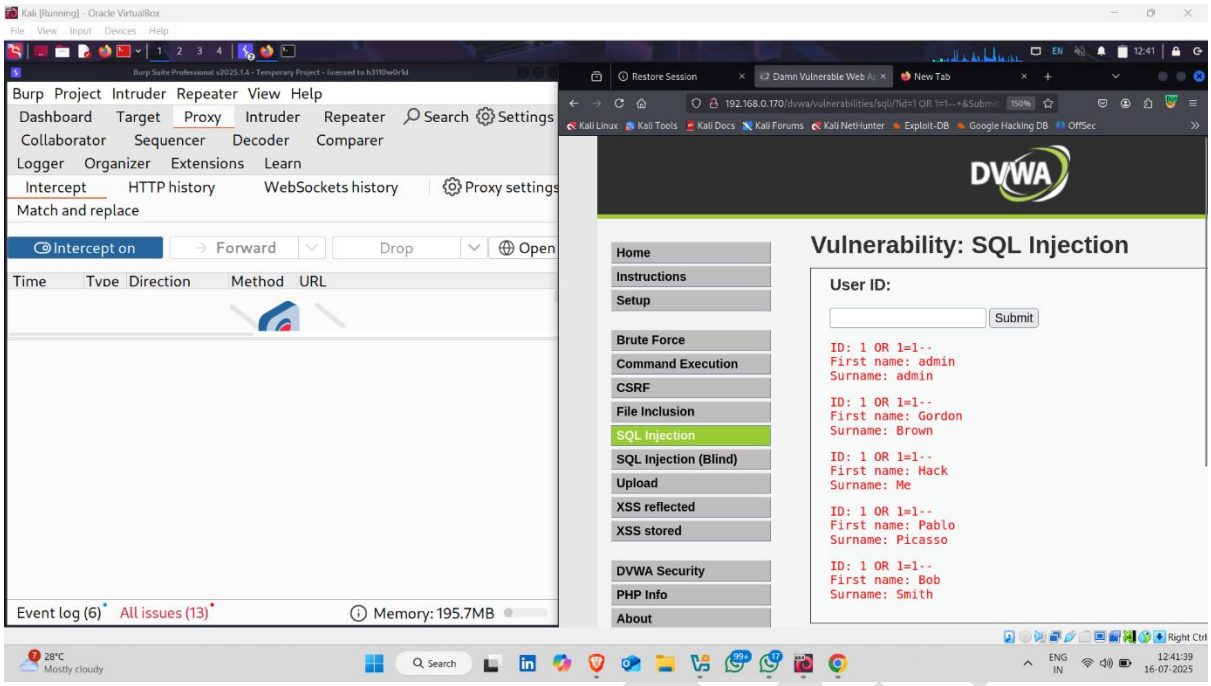
- And enter sql injection code for manual testing

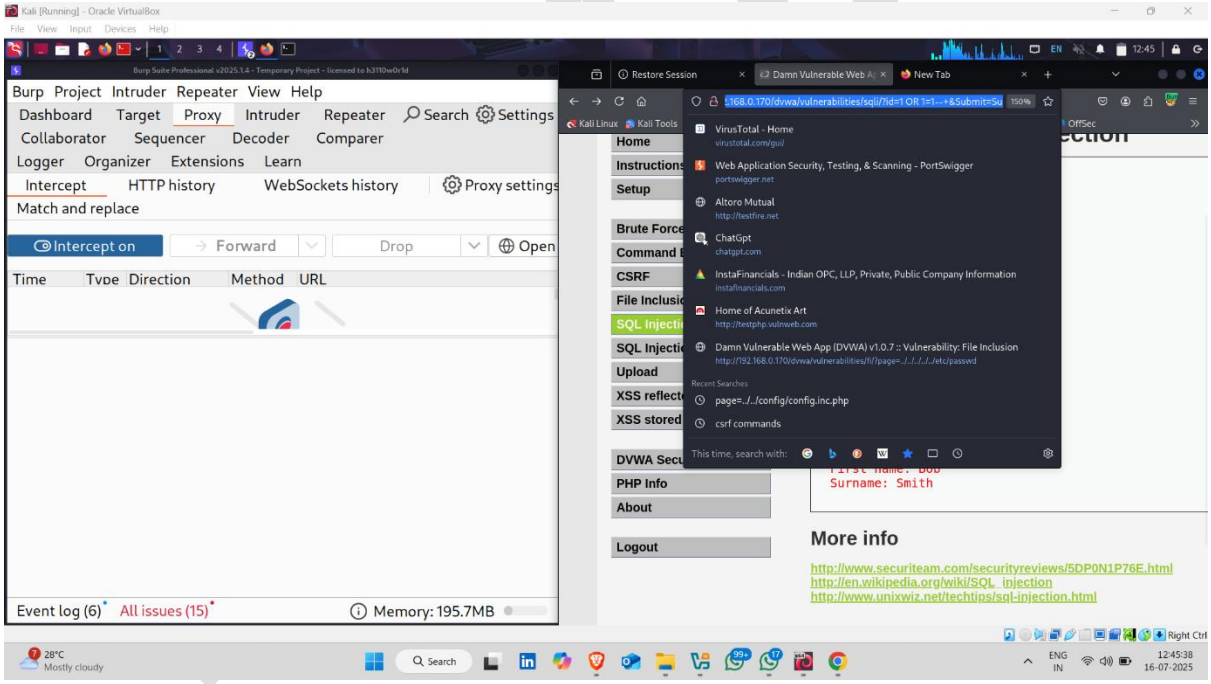




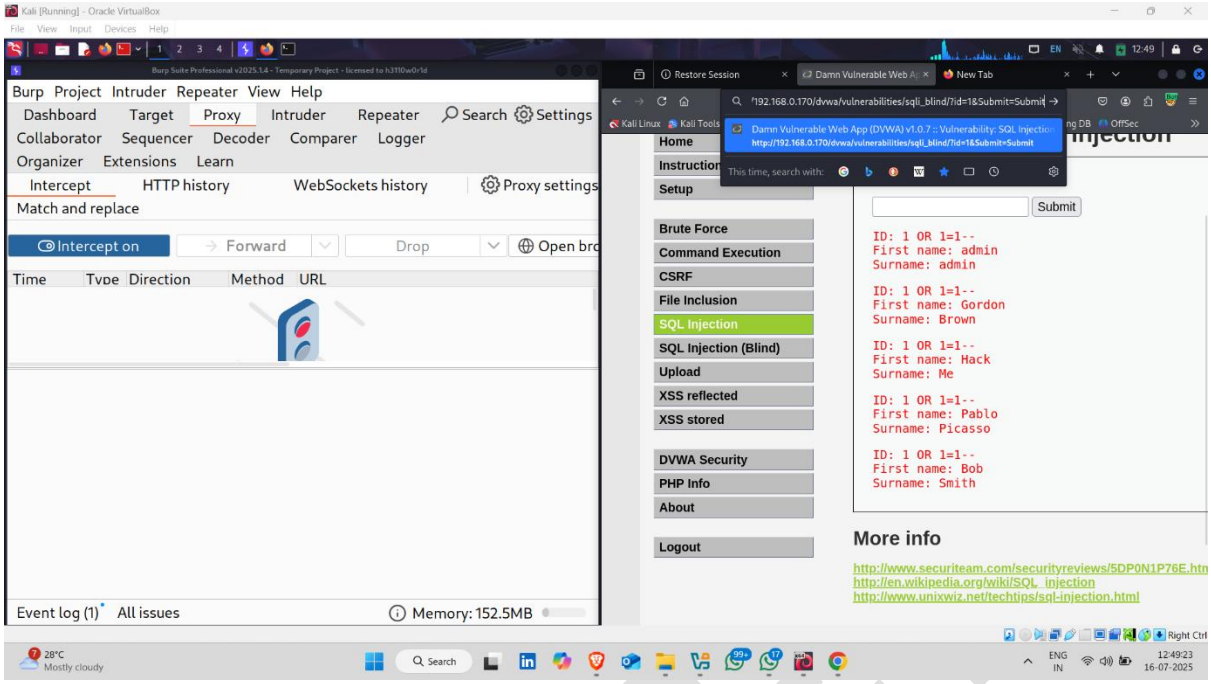
- Data access using Sql injection



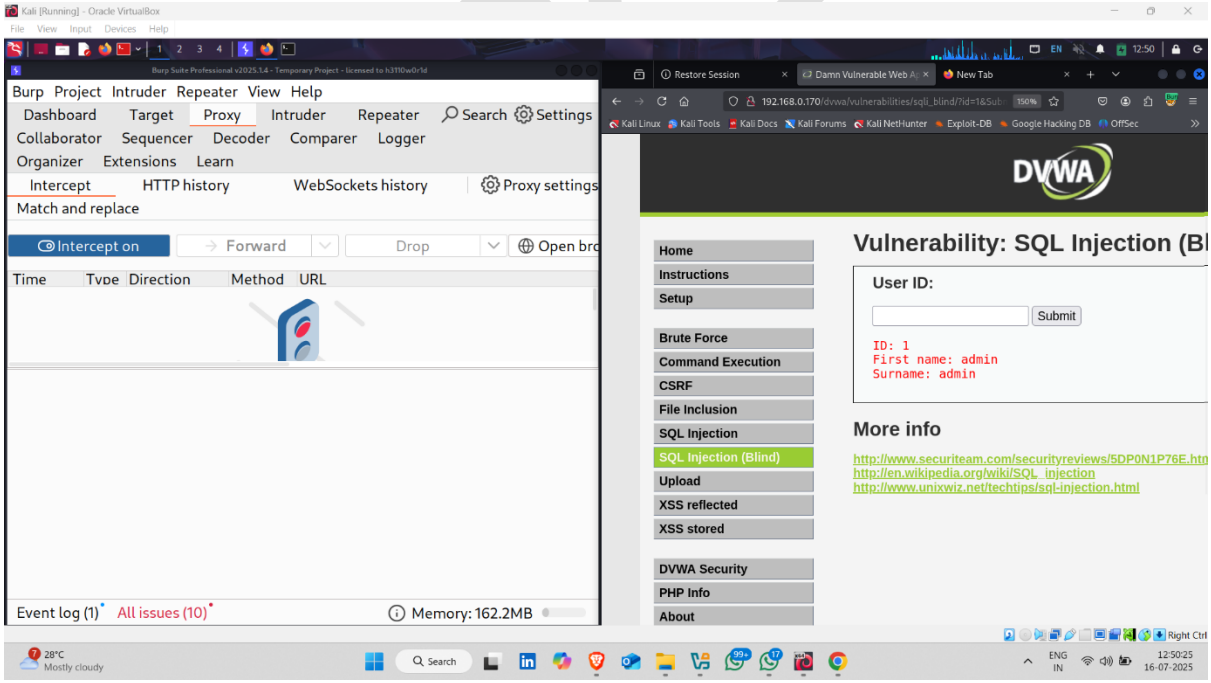




- Now click on sql injection (Blind)
- Click on url section and change sql malicious code

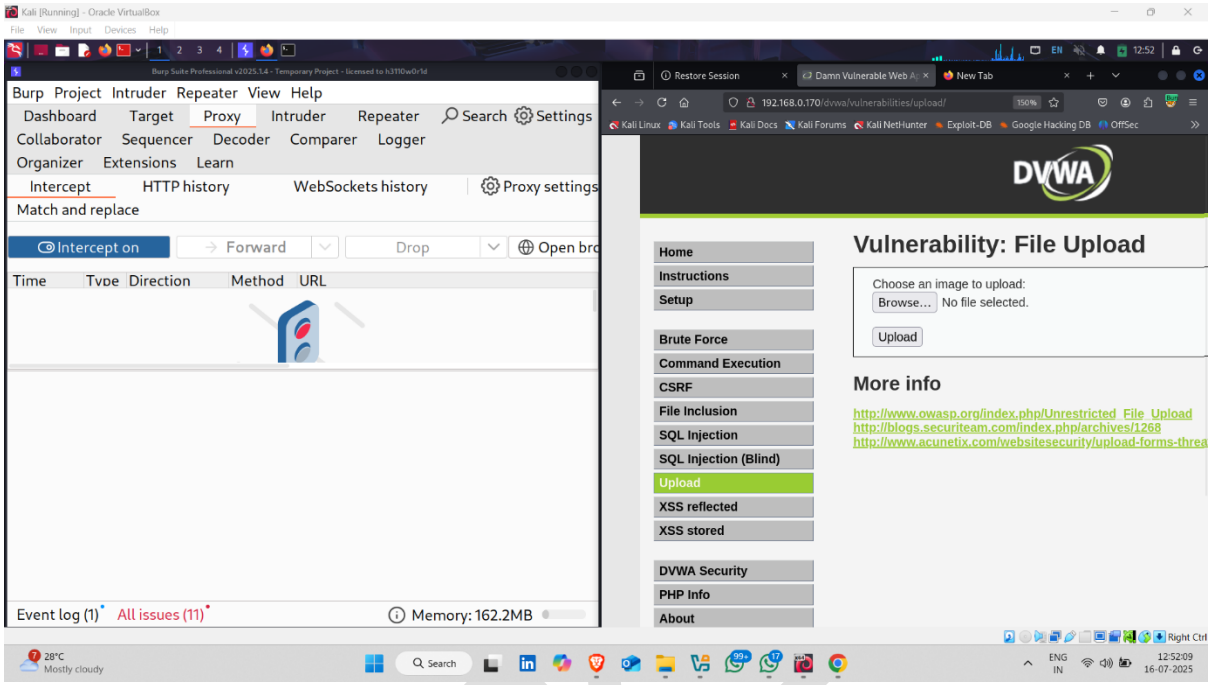

Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Search Settings
Collaborator Sequencer Decoder Comparer Logger
Organizer Extensions Learn
Intercept HTTP history WebSockets history Proxy settings
Match and replace
Intercept on Forward Drop
Time Type Direction Method URL
Event log (1) All issues Memory: 152.5MB
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Submit
ID: 1 OR 1=1--
First name: admin
Surname: admin
ID: 1 OR 1=1--
First name: Gordon
Surname: Brown
ID: 1 OR 1=1--
First name: Hack
Surname: Me
ID: 1 OR 1=1--
First name: Pablo
Surname: Picasso
ID: 1 OR 1=1--
First name: Bob
Surname: Smith
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

- Result 👇✅


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Search Settings
Collaborator Sequencer Decoder Comparer Logger
Organizer Extensions Learn
Intercept HTTP history WebSockets history Proxy settings
Match and replace
Intercept on Forward Drop
Time Type Direction Method URL
Event log (1) All issues (10) Memory: 162.2MB
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Vulnerability: SQL Injection (Bl
User ID:
Submit
ID: 1
First name: admin
Surname: admin
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

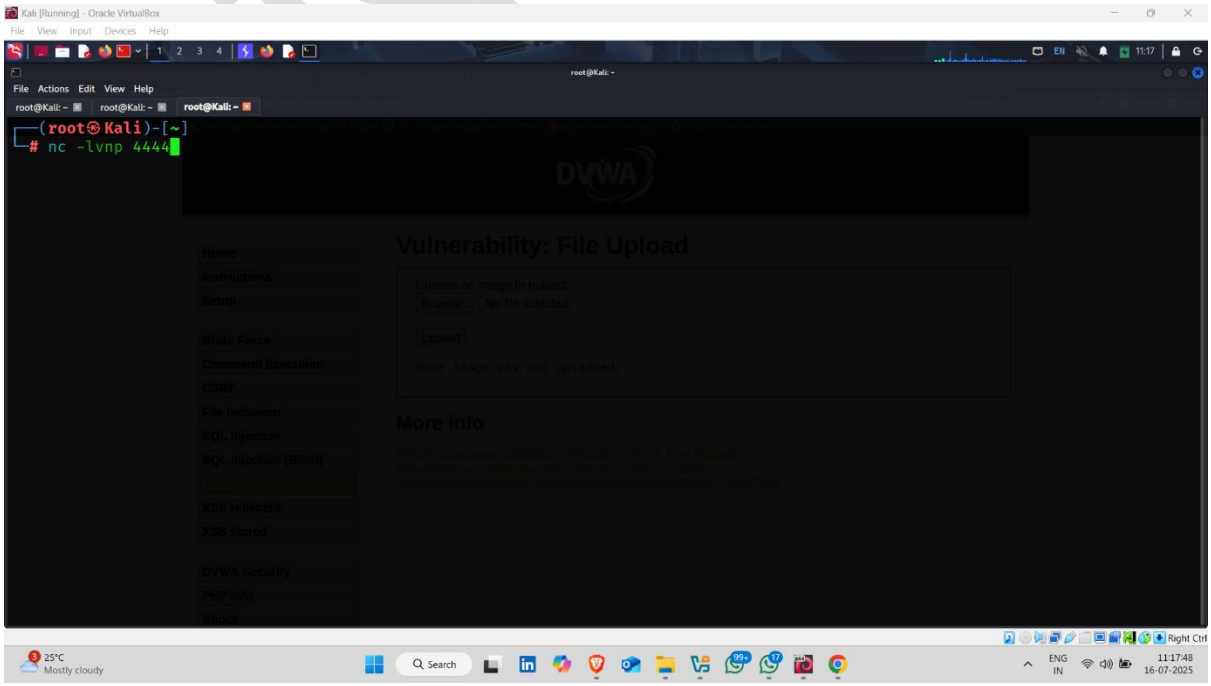
Task-4 – Upload File

- Click on **upload Section**

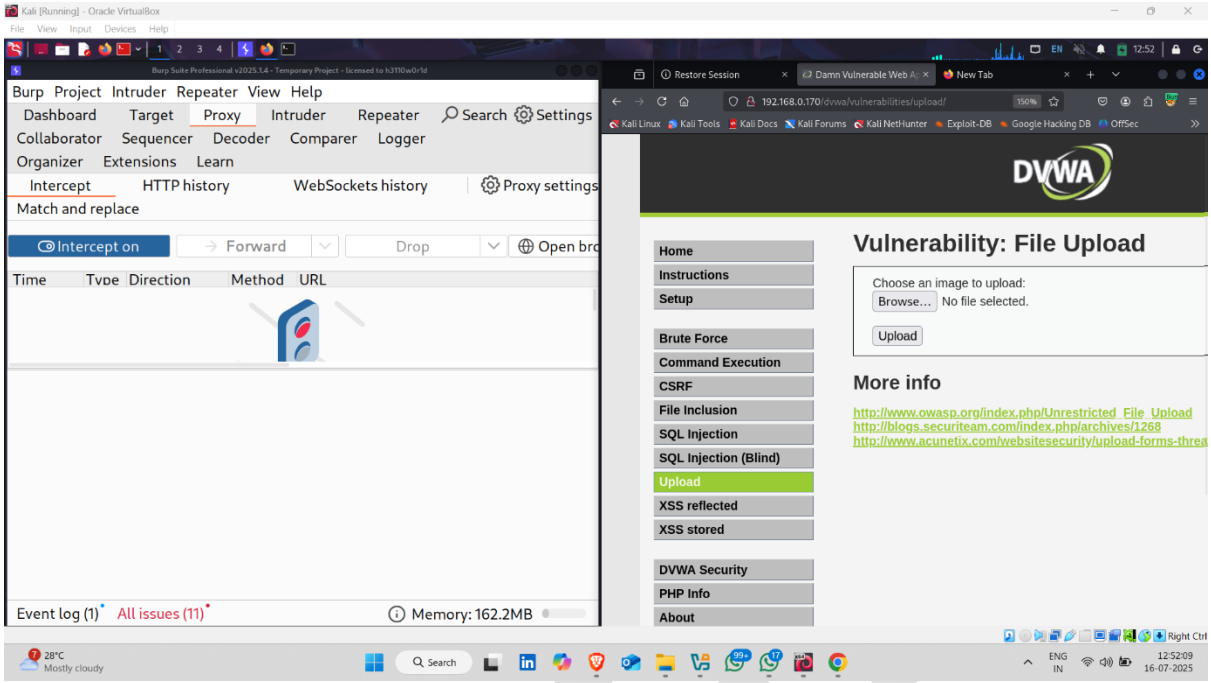


- Before upload a file , open kali linux terminal and type following command

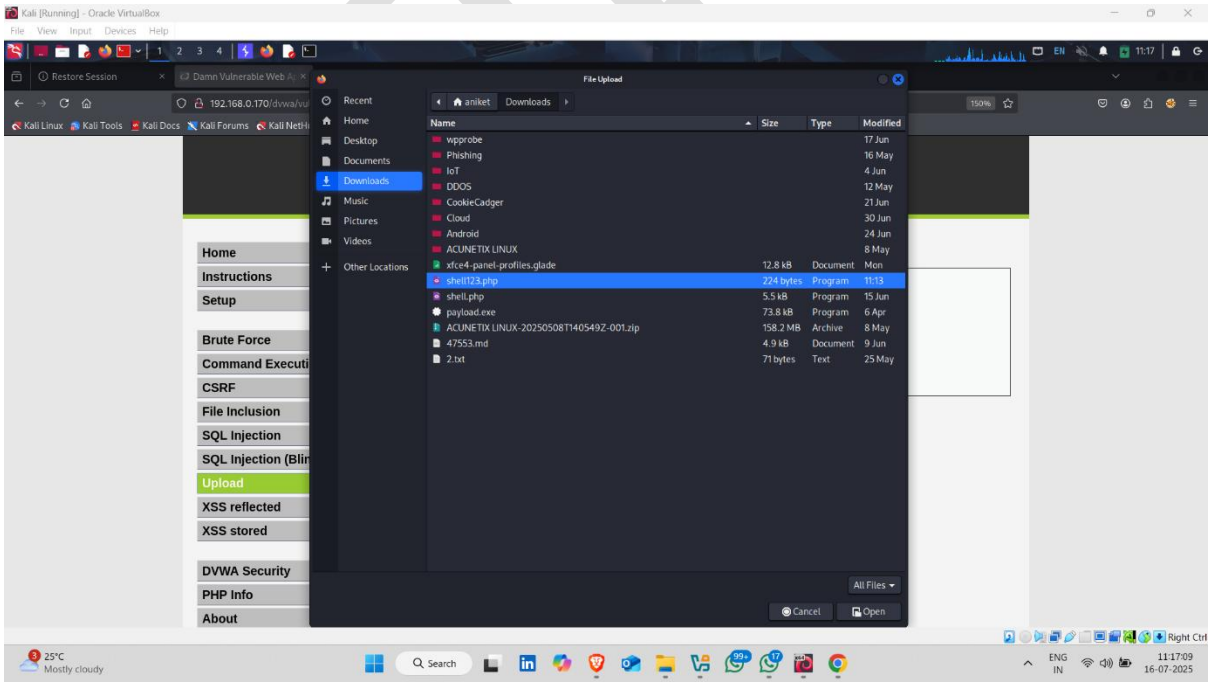
Command -: nc -lvnp 4444



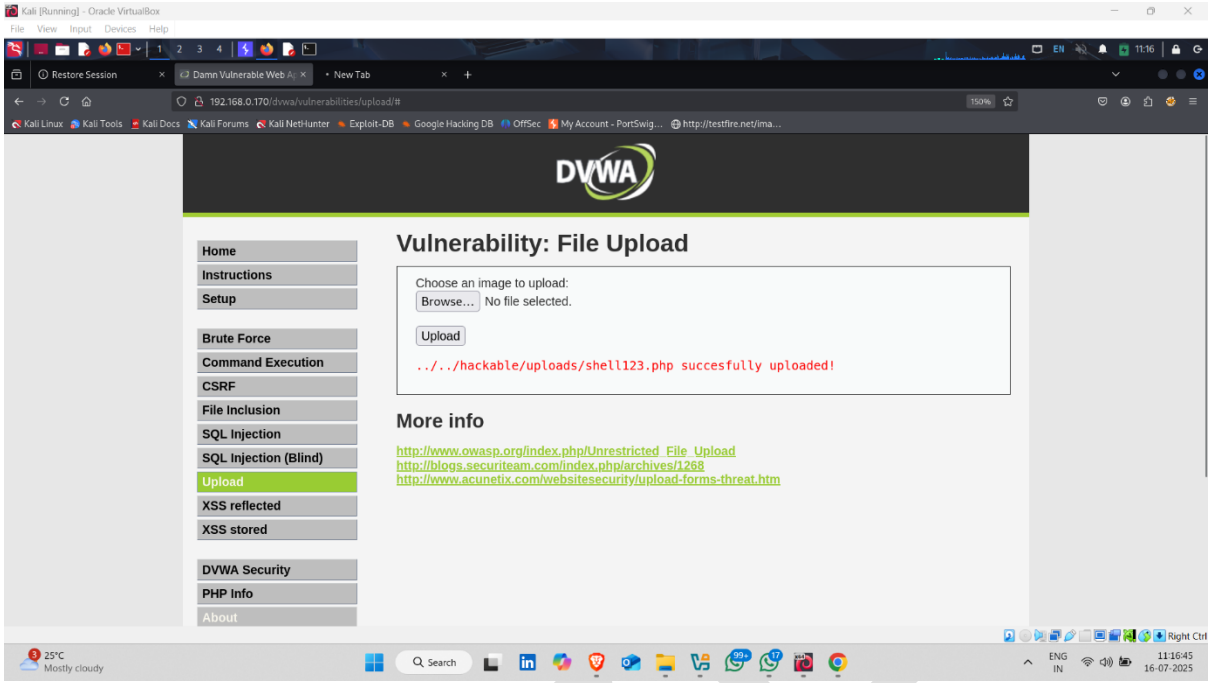
- Click on **Browse**



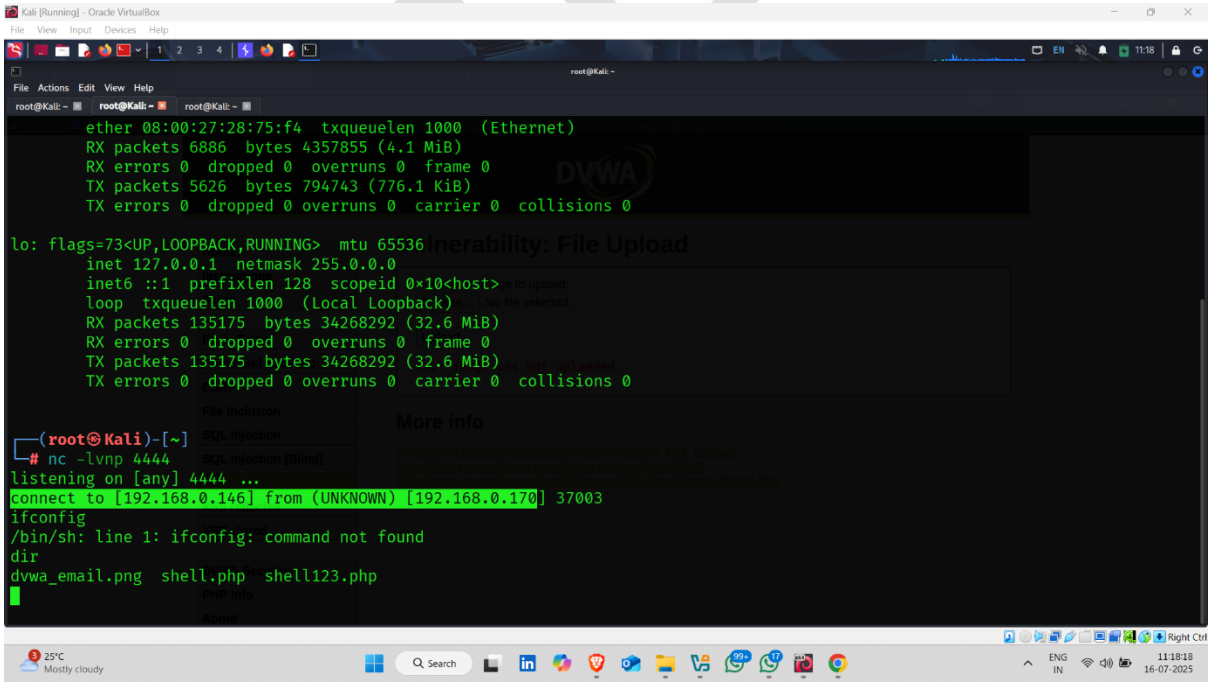
- Select a payload or **malicious file** for uploading and then click on **open**



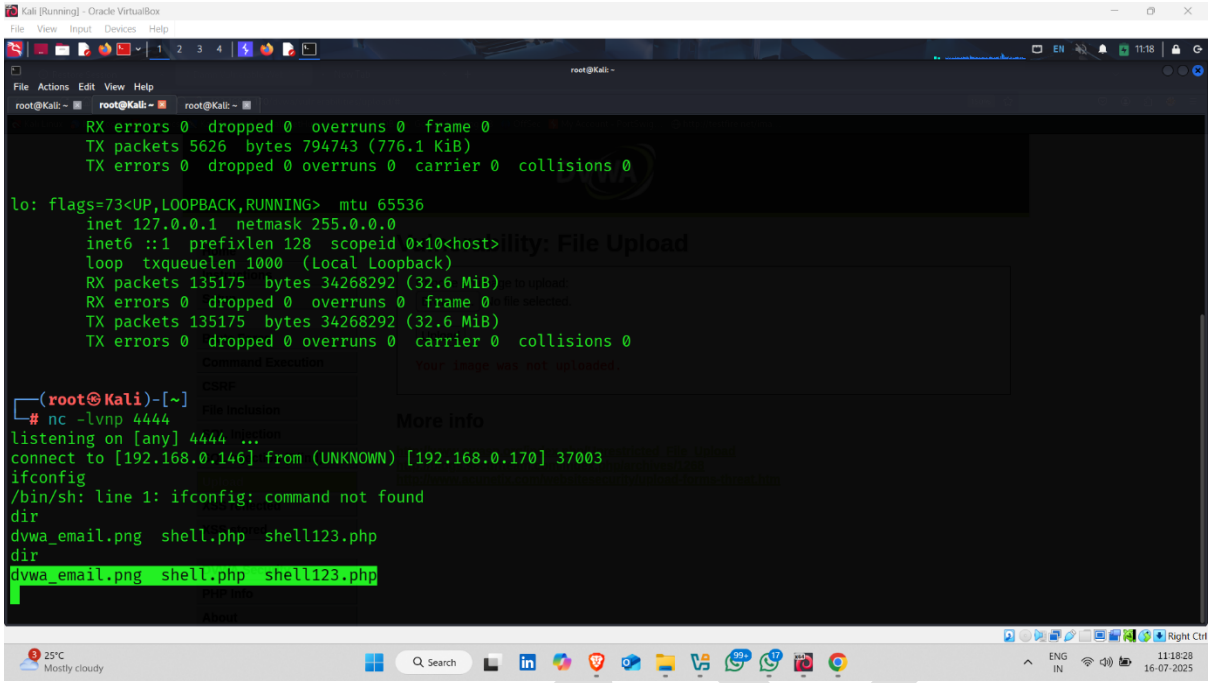
- File uploaded successfully



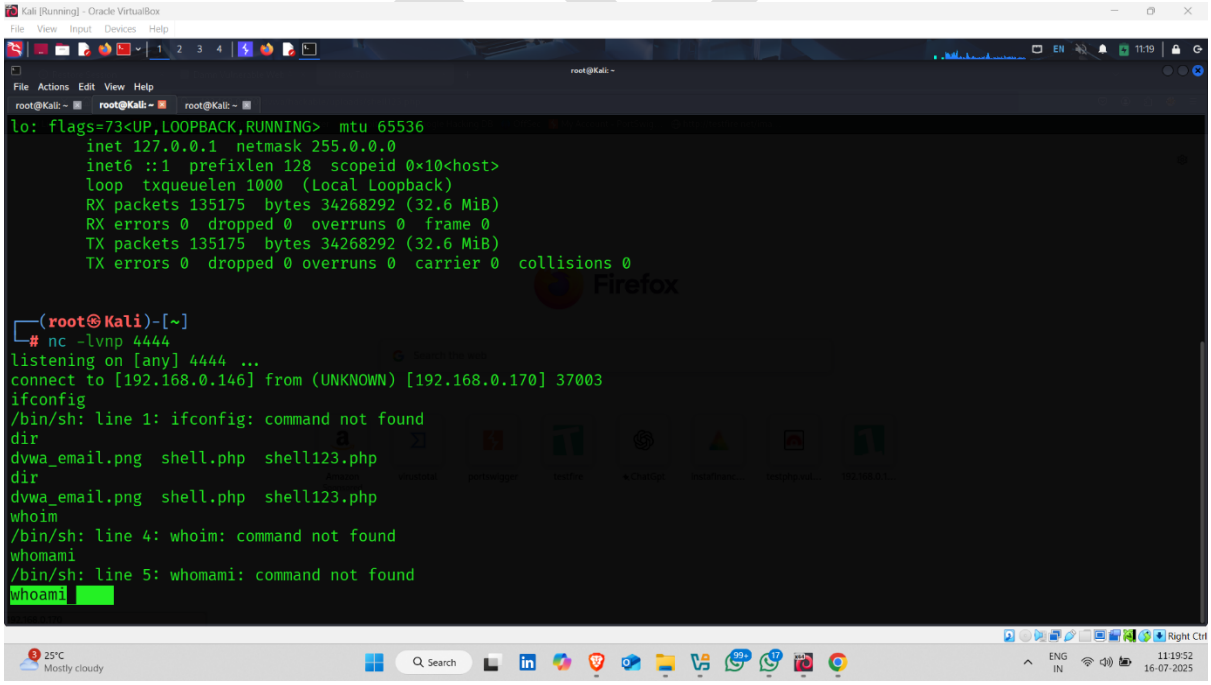
- And here , we received connection from target website



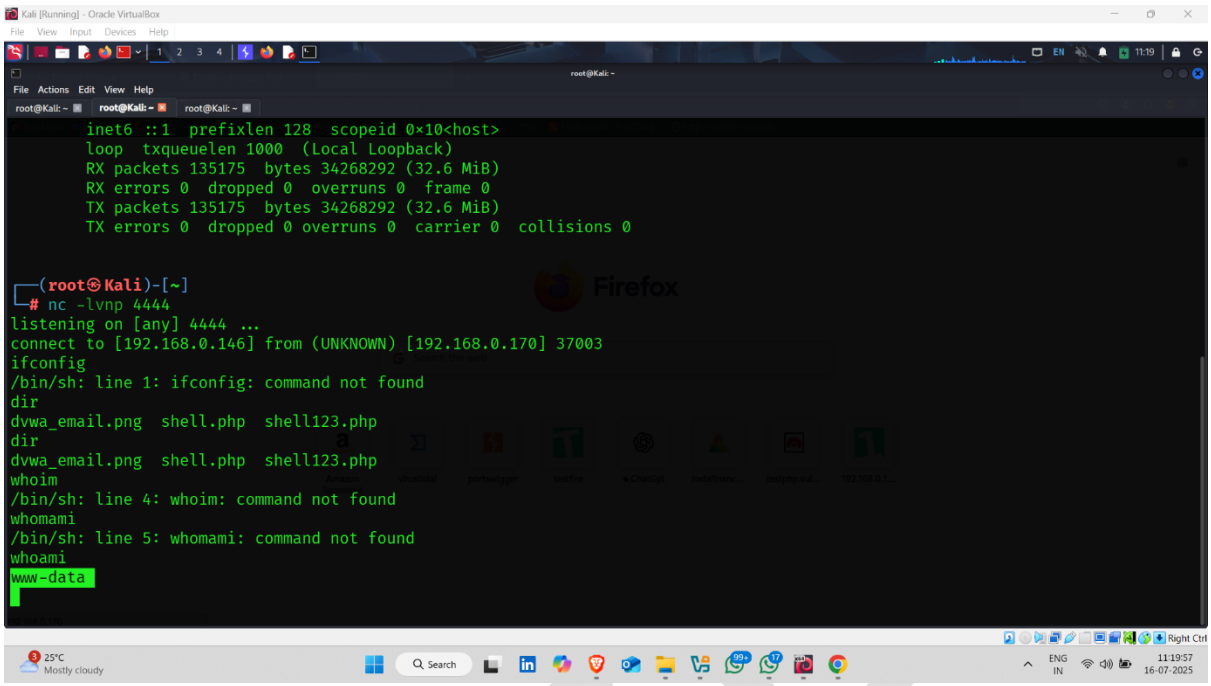
- Type **dir** to see the directories of target website



- Type whoami

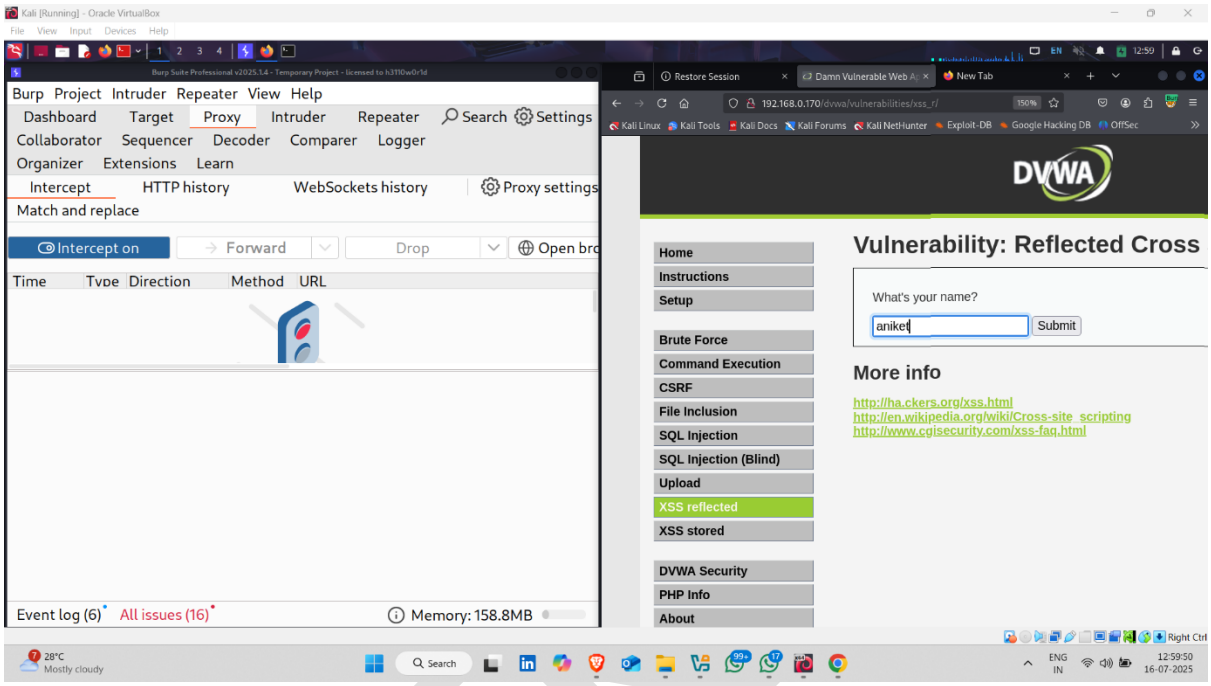


- Result 👇✅

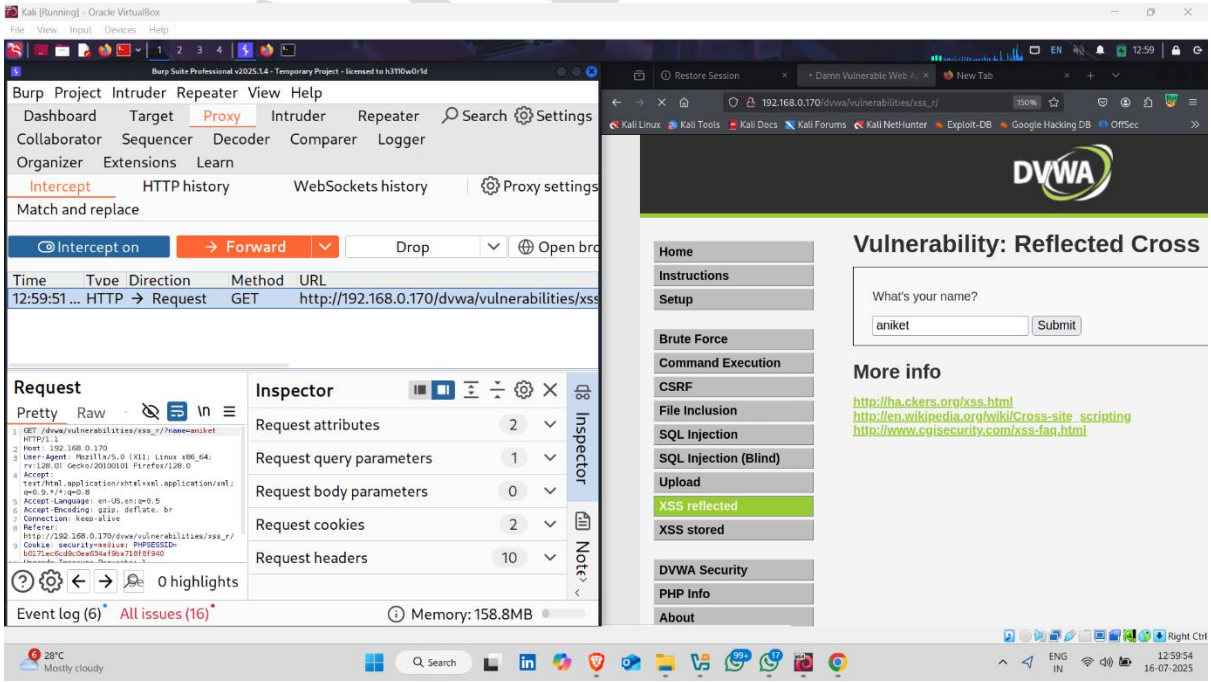


Task -5—XSS Reflected

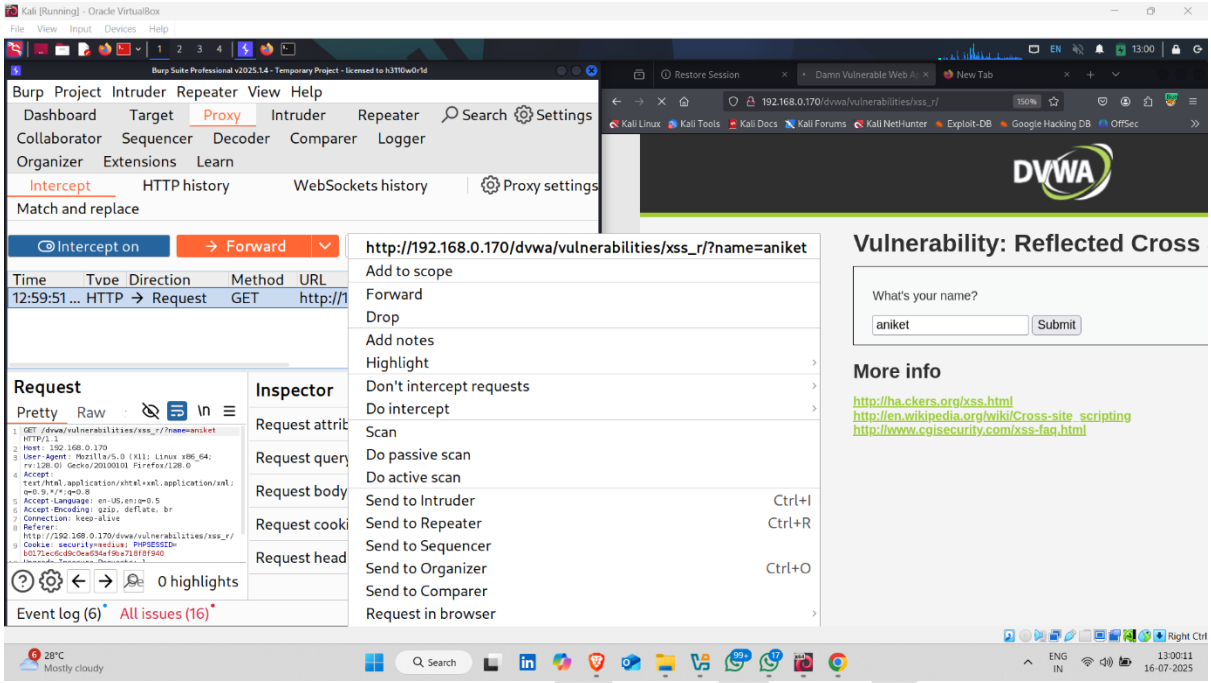
- Click on **Xss Reflected Section**
- Enter **random string** and click on **submit** button



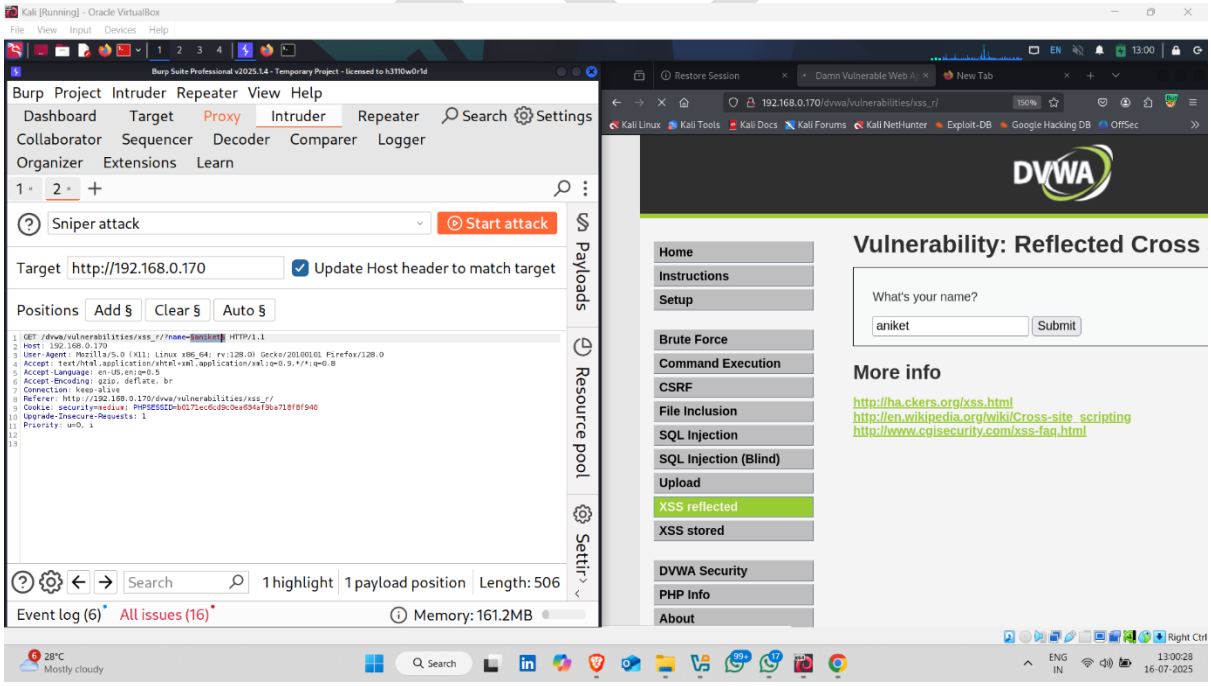
- Request intercepted



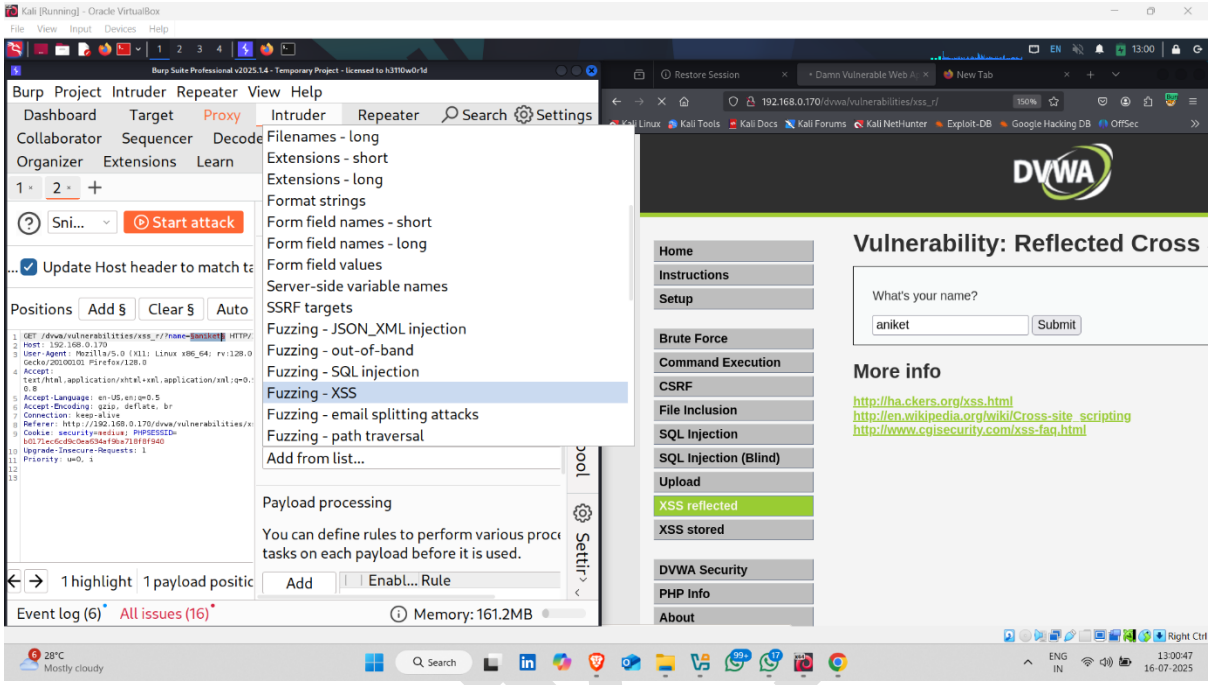
- Right click on request and send it to the **intruder**



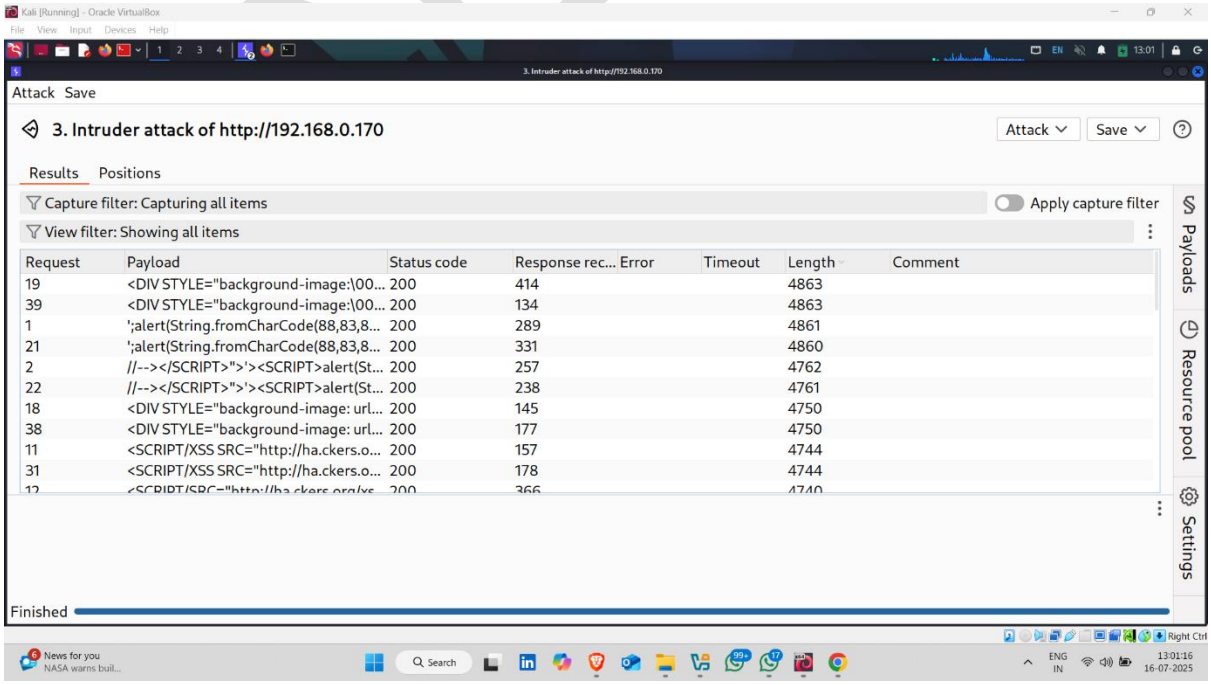
- Set payload position on **string** that you enter



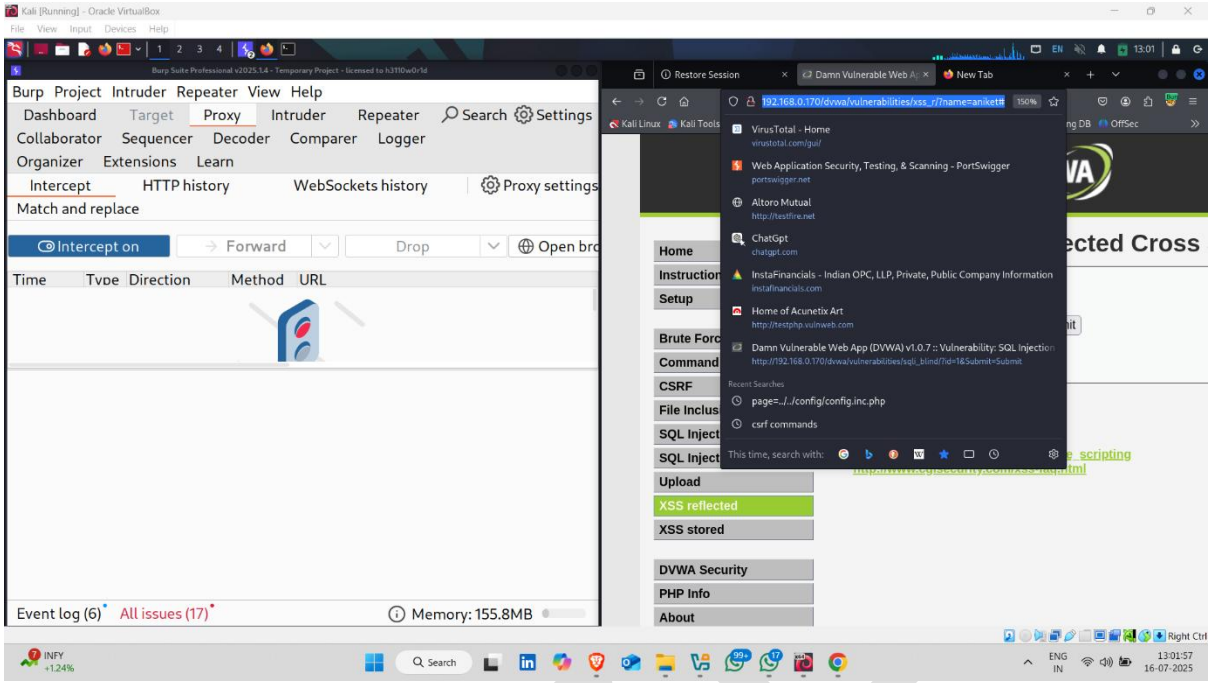
- Click on **payload** option and then click on **add from list** and select **fuzzing-XSS**
- And then click on **start attack**



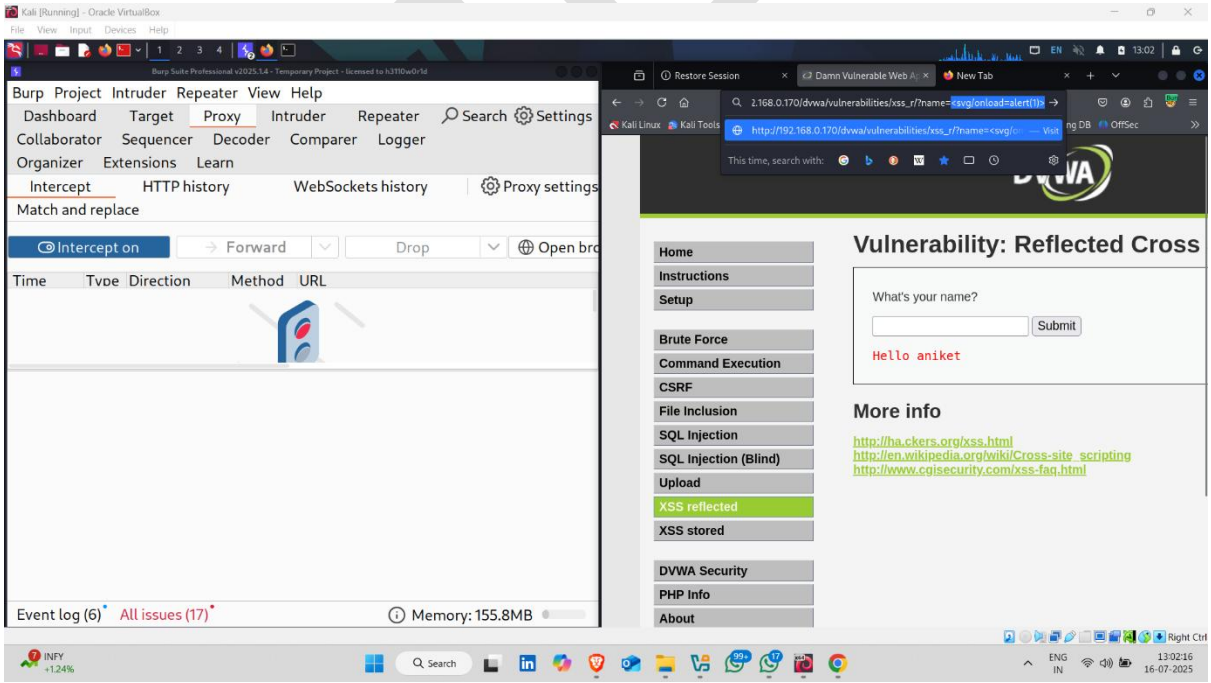
- Attack finished ✅ 👇
- Now copy any script for manual testing



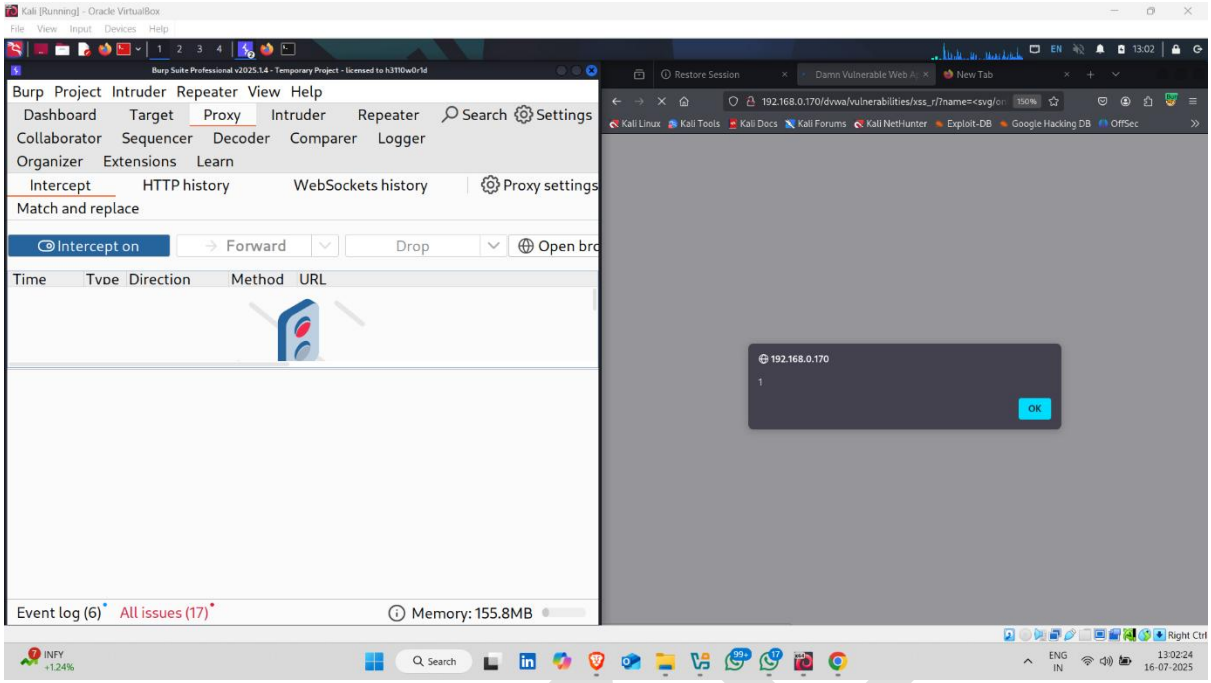
- Click on **url section**



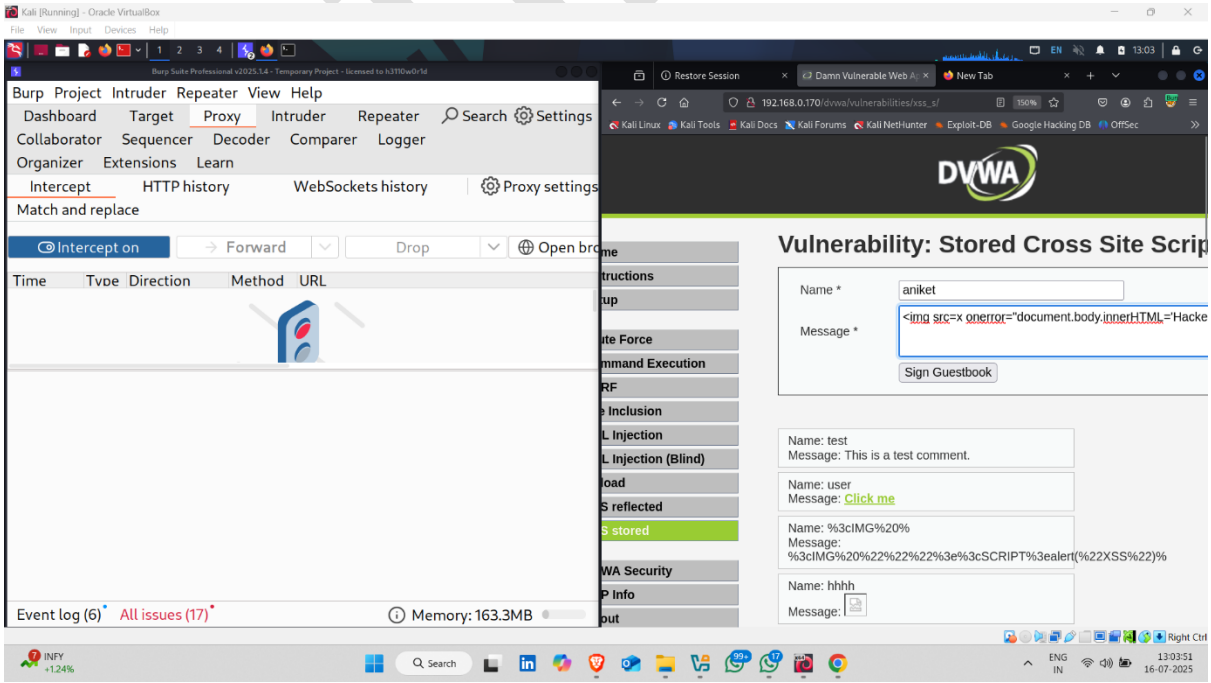
- Use script instead of name string



- Pop Up appears 👇 ✅



- Now click on **XSS Stored**
- Enter a **random name** and **enter script on message box** and click on **Sign Guestbook**



- Pop Up appears ✅ 👇