RUBY
GO
PHP
PYTHON
SWIFT
C++
HTML5
CSS3
CYBER ATTACK
VIRUS

Module 14 web application security

1 What is web application

📄 Examples of Web Applications

🔧 How Web Applications Work

🔧 Types of Web Servers

Types of web server attack

Task1 footprint web infrastructure particular web site and tool

- 1 method footprint web infrastructure using website: whatweb
- 2 method footprint web infrastructure using website: whois lookup
- 3 method footprint web infrastructure using website https://centralops.net/

Task2 Banner Grabbing from SSL Service

- Method 1 Gathering the wordlist from the Target website

Task 3 identifiye the web application port and service discovery with nmap

Task4 Detecting web application firewall using wafw00f

Task 5 perfrom web app application vulnerability using smart scanner

Task 6 perfrom web app application vulnerability using Acunitix

Task 7 perfrom web app application vulnerability using Zaprox

Task 8 Web application attack methodlogy

Types of input validation attack

- Buffer Overflow Input Validation
- Canonical Ideation Input Validation Attack:

- XSS Attack:
- SQL Injection Attack

Types of web application attacks

- SQL Injection (SQLi)
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- Remote Code Execution (RCE)
- Directory Traversal
- File Inclusion (LFI/RFI)
- Command Injection
- Broken Authentication
- Insecure Direct Object References (IDOR)
- Security Misconfiguration
- XML External Entity (XXE) Injection
- Server-Side Request Forgery (SSRF)
- Session Hijacking
- Clickjacking

- Path Disclosure
- Unrestricted File Upload
- HTTP Host Header Attack
- Web Cache Poisoning
- Business Logic Flaws
- Broken Access Contro

Task 9 how to test web application using sniper burp suite using sql injection attack

How to test username

Task 10 how to test web application using burp suite using bomber cluster bomb attack

How to test username and password sql injection vunlabritiy

Task 11 how to test web application using burp suite using

Battering ram attack

How to test web application search box

- **How to defend against injection attack**
- **How to Defend against web application attacks**

Extra activity task 11 who to otp bypass using burp suite there is two method of otp bypass

1 st method is server is response manipulate

2 method is otp bypass in brute force attack

What is web application

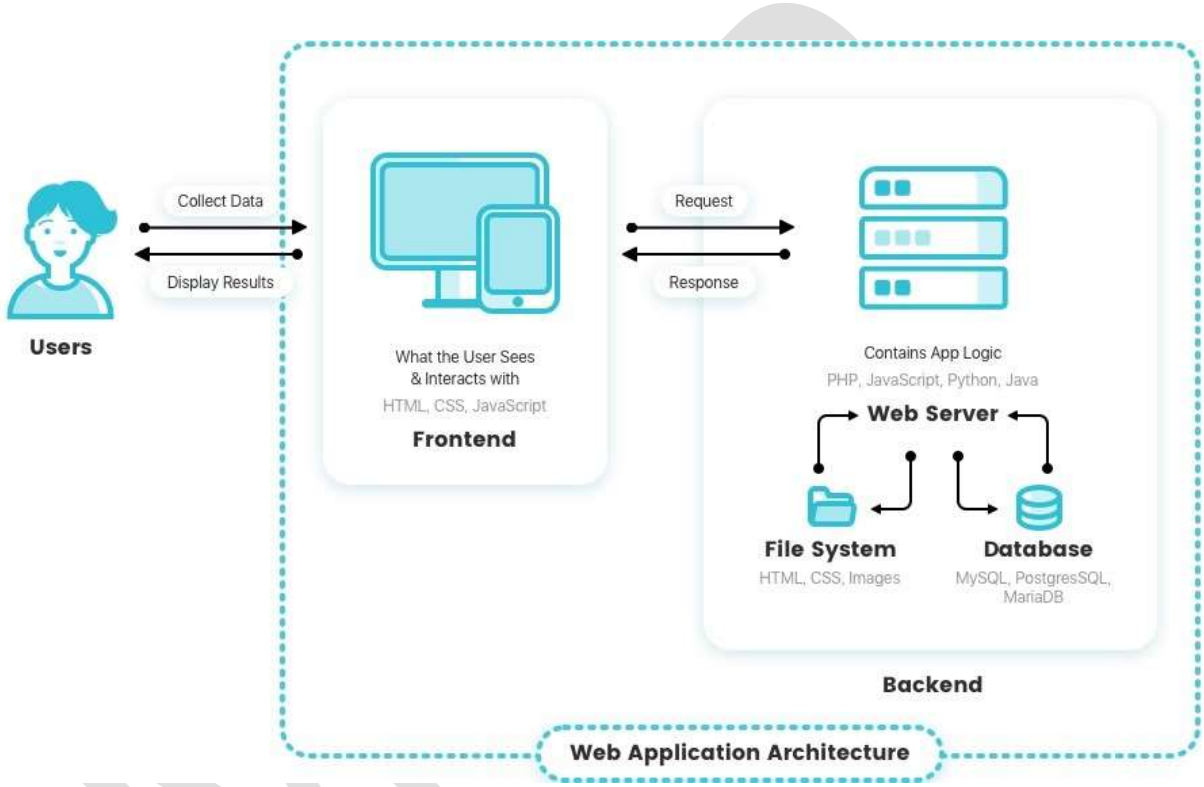
A **web application** (or **web app**) is a software program that runs on a web server and is accessed through a web browser over the internet. Unlike traditional desktop applications, web apps don't require installation on a user's device. They are designed for user interaction, enabling tasks such as form submissions, data processing, and real-time updates.

🔧 How Web Applications Work

Web applications operate on a **client-server model**, comprising:

- **Client-Side (Front End):** The user interface built with HTML, CSS, and JavaScript, which runs in the user's browser.
- **Server-Side (Back End):** The server processes requests, executes business logic, and interacts with databases to serve dynamic content.
- **Database:** Stores and manages data, such as user information and application content.

When a user interacts with a web app, their browser sends a request to the server. The server processes this request, retrieves or updates data in the database if necessary, and sends back a response that the browser renders for the user.



Examples of Web Applications

- **Email Services:** Gmail, Outlook Web
- **Productivity Tools:** Google Docs, Microsoft Office Online
- **E-commerce Platforms:** Amazon, Flipkart
- **Social Media:** Facebook, Twitter
- **Banking Services:** Online banking portals

- **Project Management:** Trello, Asana

🔧 Types of Web Servers

1. Apache HTTP Server

- **Overview**: One of the most widely used open-source web servers, developed by the Apache Software Foundation.
- **Key Features**:
  - Highly customizable with a modular architecture.
  - Supports multiple operating systems, including Linux, Windows, and macOS.
  - Extensive community support and documentation.
- **Use Case**: Ideal for websites requiring flexibility and extensive customization options.

---

2. Nginx

- **Overview**: Pronounced "Engine-X," Nginx is known for its high performance, stability, and low resource consumption.
- **Key Features**:

    - Event-driven architecture capable of handling many concurrent connections.
    - Functions as a web server, reverse proxy, load balancer, and HTTP cache.
    - Widely adopted by high-traffic websites for its scalability.
- **Use Case**: Suitable for high-traffic websites and applications requiring efficient load balancing and reverse proxy capabilities.

---

3. Microsoft Internet Information Services (IIS)

- **Overview**: A flexible, secure, and manageable web server for hosting anything on the web, developed by Microsoft.
- **Key Features**:
    - Tightly integrated with Windows Server and other Microsoft products.
    - Supports ASP.NET applications and other web technologies.
    - Provides a graphical user interface for easy management.
- **Use Case**: Best suited for organizations utilizing the Microsoft ecosystem and developing applications with ASP.NET.

---

4. LiteSpeed Web Server

- **Overview**: A commercial web server known for its high performance and low resource usage.
- **Key Features**:
  - Offers Apache compatibility with enhanced performance.
  - Built-in anti-DDoS features and caching mechanisms.
  - Supports HTTP/3 and QUIC protocols.
- **Use Case**: Ideal for businesses seeking improved performance over Apache without
- significant configuration changes.

---

5. Apache Tomcat

- **Overview**: An open-source implementation of the Java Servlet, JavaServer Pages, and Java Expression Language technologies.
- **Key Features**:
  - Designed specifically to run Java applications.
  - Lightweight and easy to configure.
  - Supports integration with other Apache projects.

- **Use Case**: Suitable for developers building and deploying Java-based web applications

---

Types of web server attack

Task1 footprint web infrastructure particular web site and tools

1 there is tool called whatweb

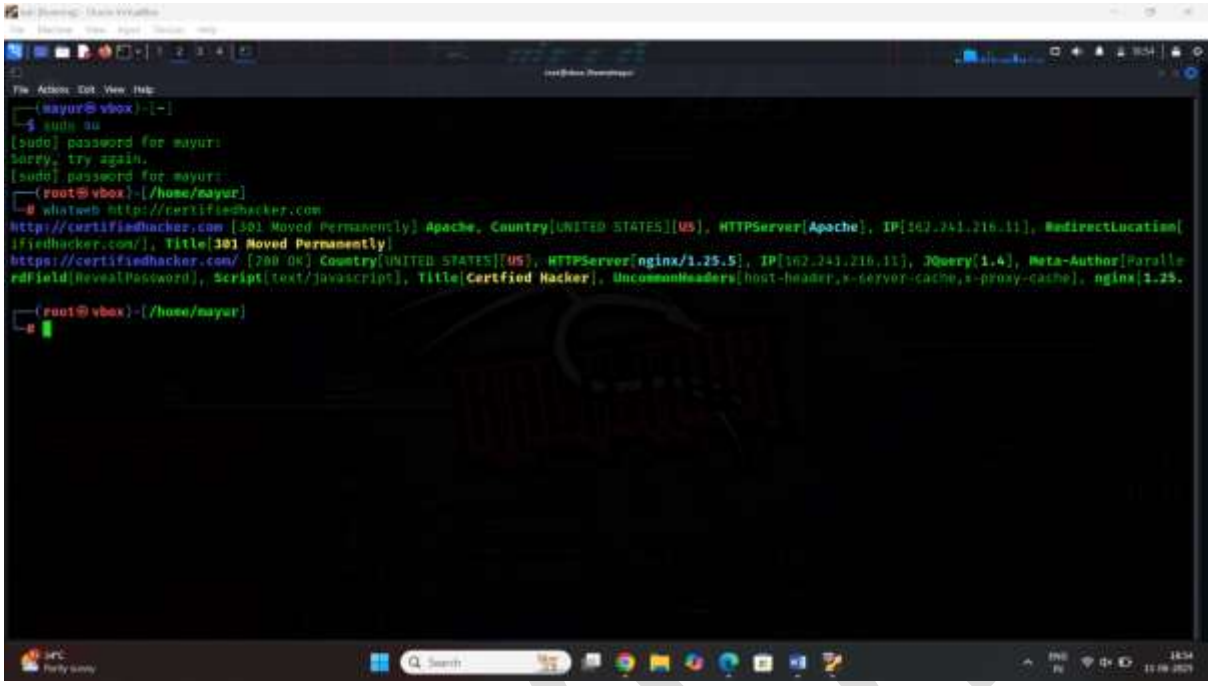
How to install process

1 sudo apt update

2 sudo apt install whatweb

Command: whatweb http://certifiedhacker.com

Result:



Description: this tool is find the information of target web server eg: web server hosted country and web server type ,title etc

2 method footprint web infrastructure using website: whois lookup

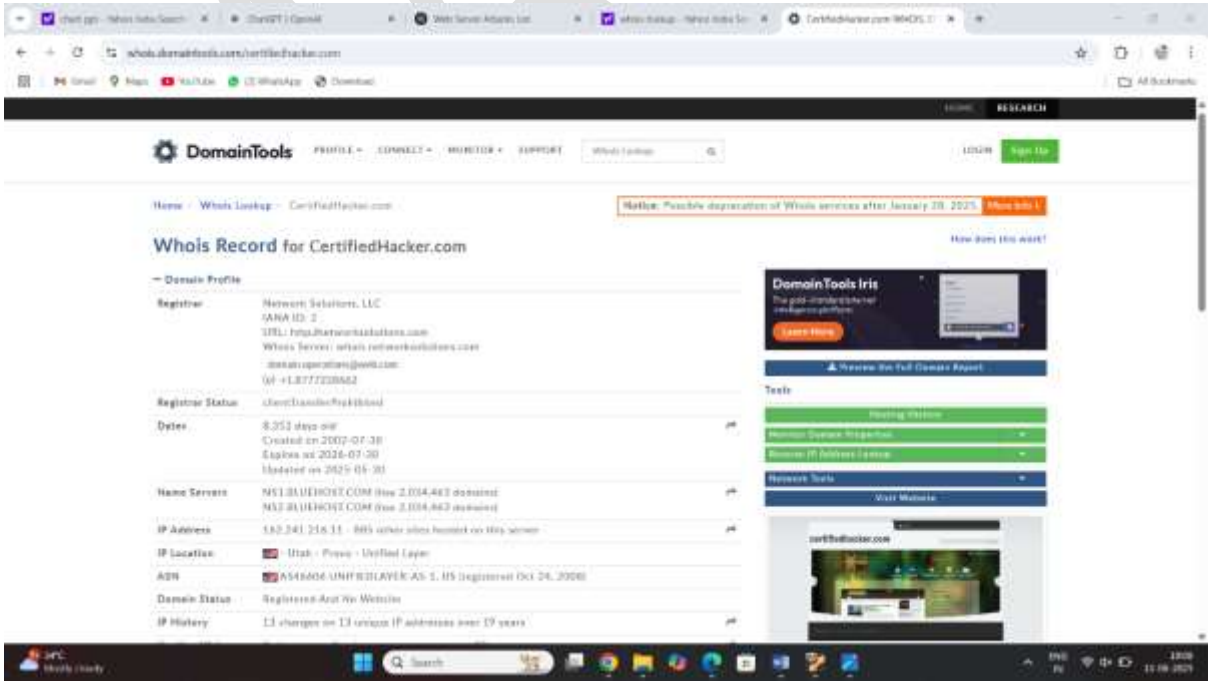
How to use this website

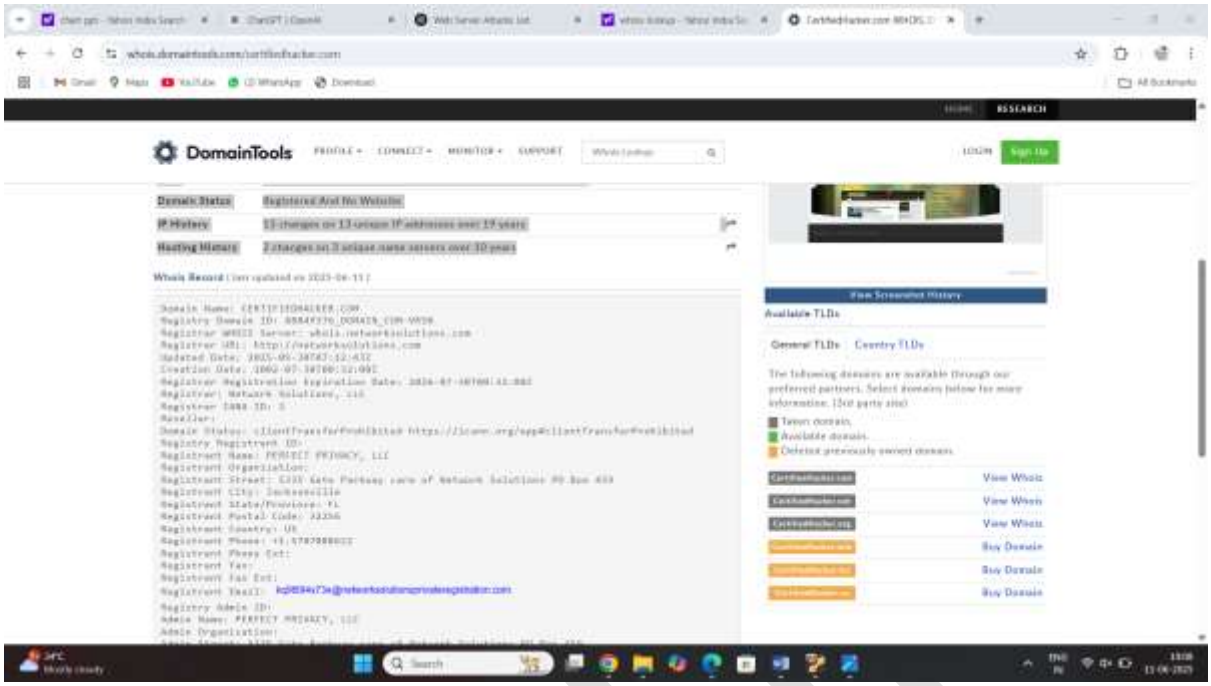
Step1: go to browser type the simple whois lookup

Click on this link and type this target web server domain

Result:


DomainTools
Whois Lookup
certifiedhacker.com
Upgrade Your Membership and Elevate Your Defenses


DomainTools
Whois Record for CertifiedHacker.com
Domain Profile
DomainTools Iris



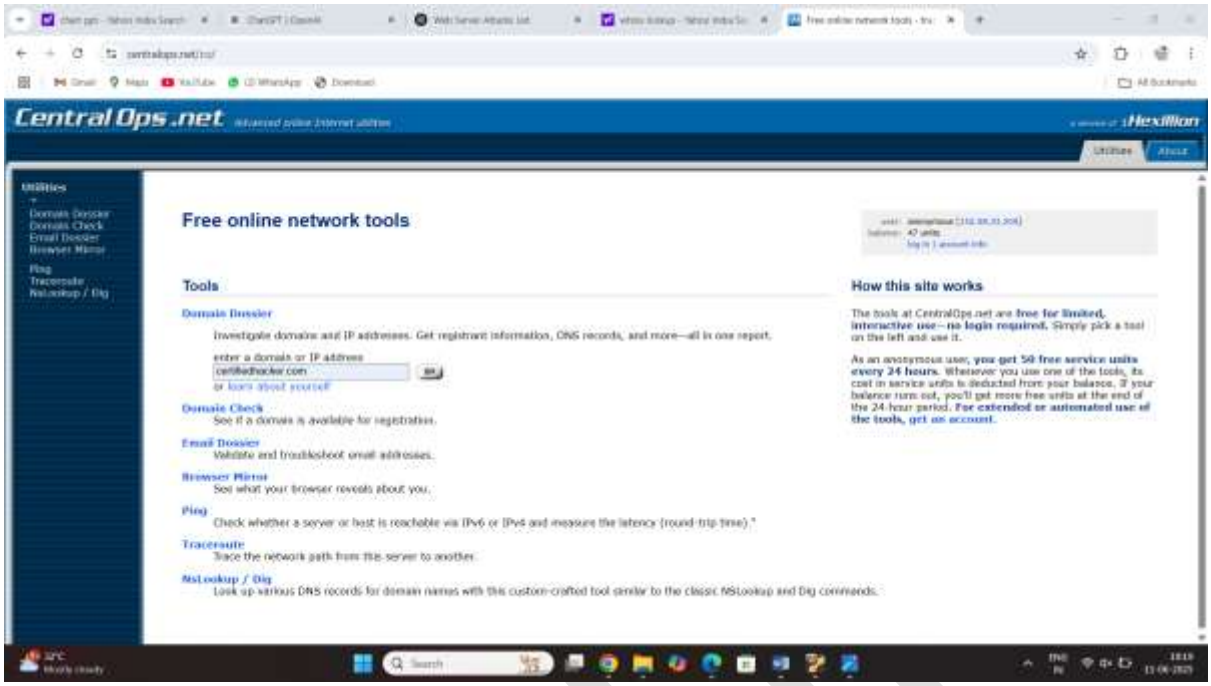
3 method footprint web infrastructure using website

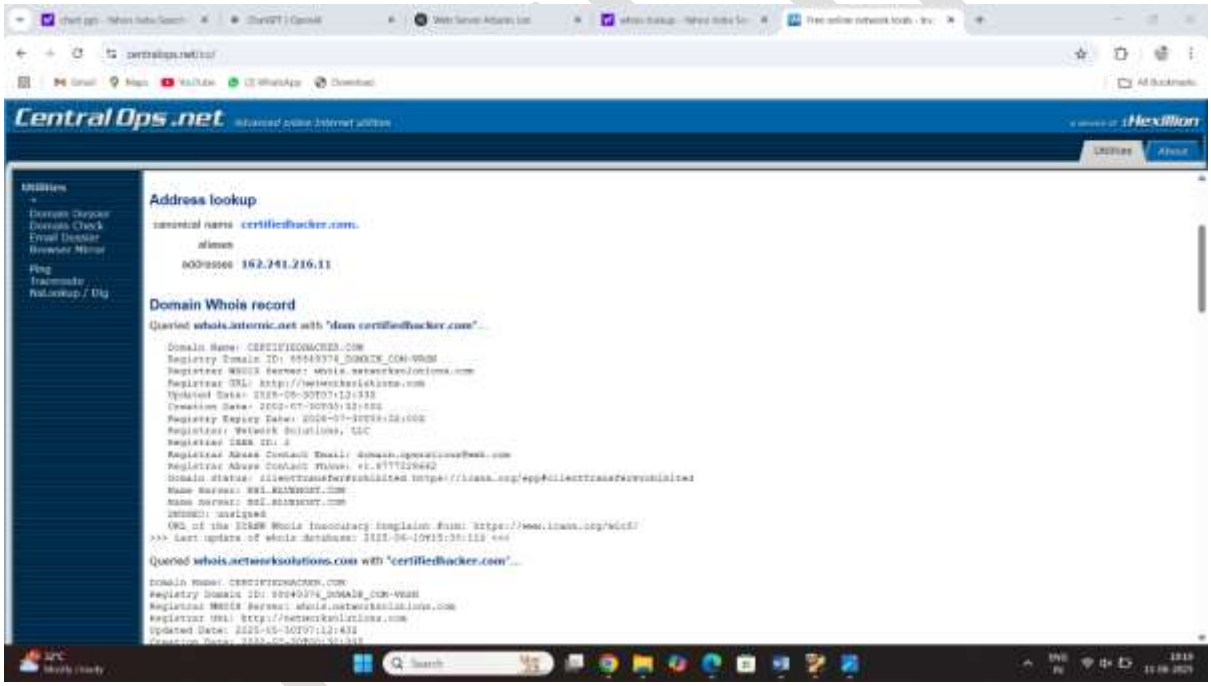
https://centralops.net/

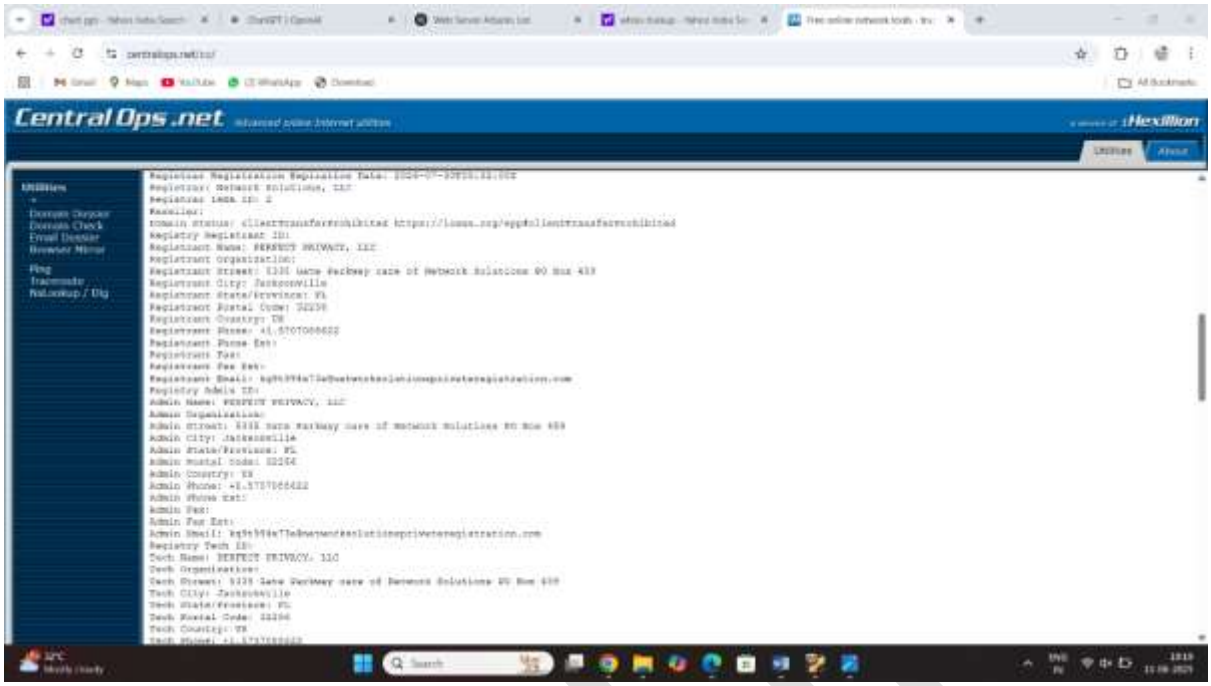
This web site find the domain name and host country and email, contact sub domain etc

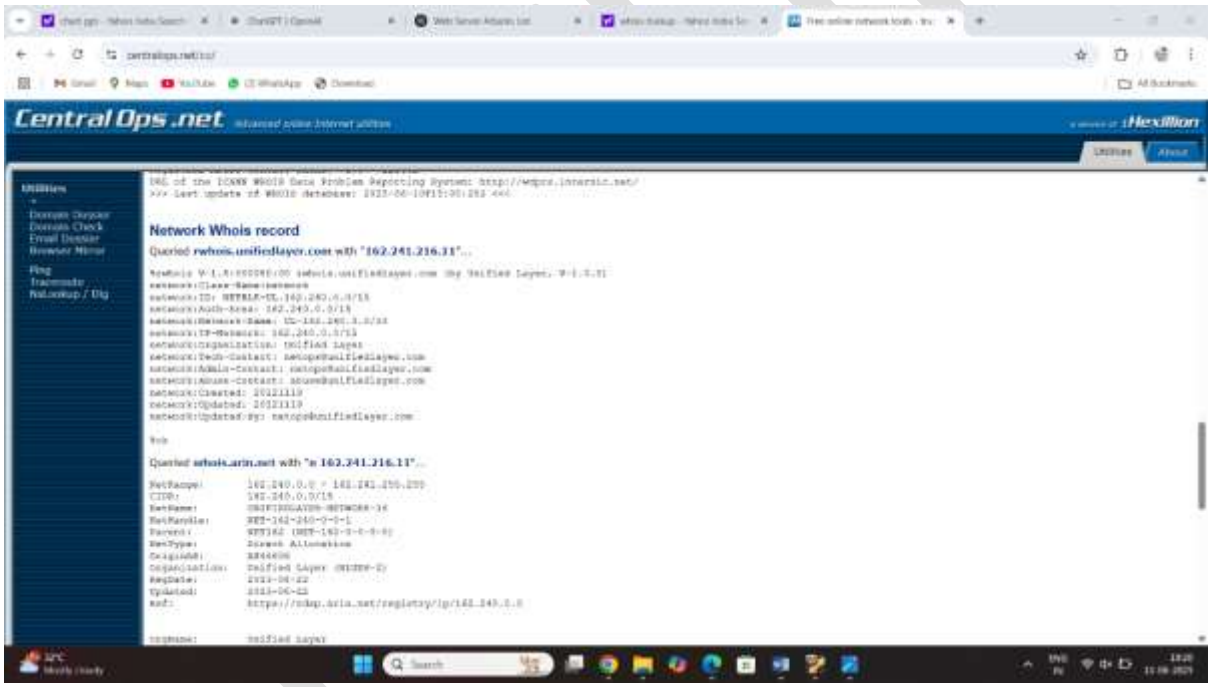
Result:

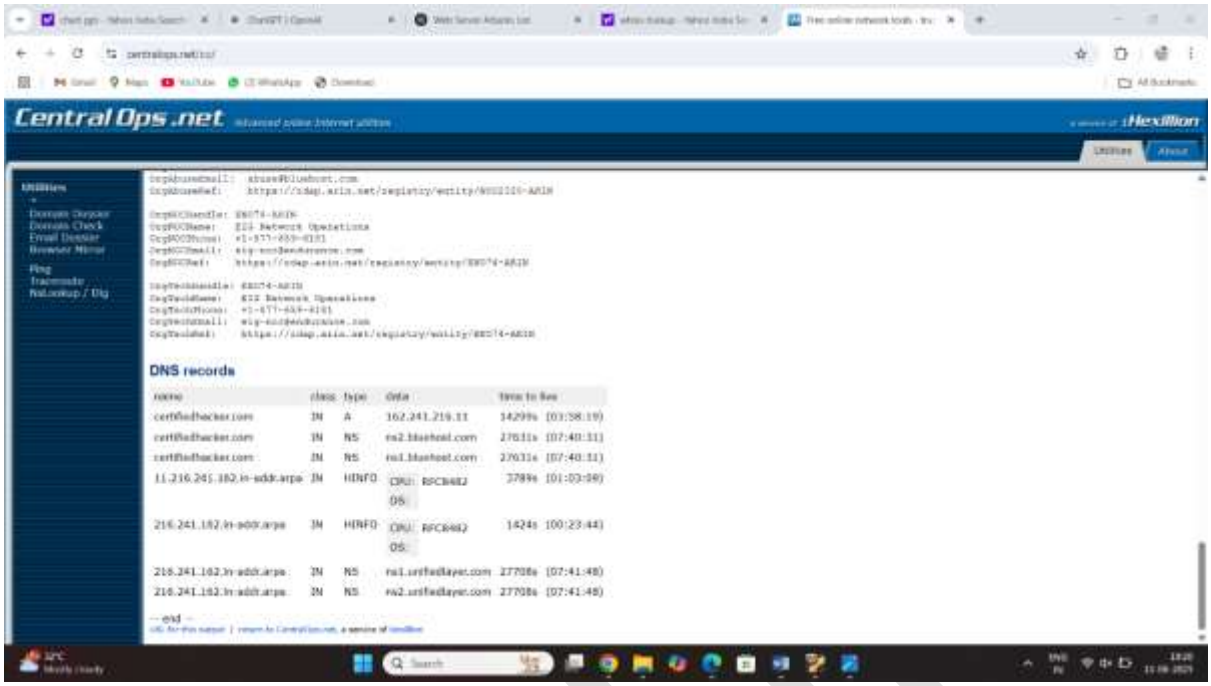
Step1: go to web site and click on the web site type the target web server domain and ip


CentralOps.net
Free online network tools
Tools
Domain Dossier
Investigate domains and IP addresses. Get registrant information, DNS records, and more—all in one report.
enter a domain or IP address
certifiedhacker.com
Domain Check
See if a domain is available for registration.
Email Dossier
Validate and troubleshoot email addresses.
Browser Mirror
See what your browser reveals about you.
Ping
Check whether a server or host is reachable via IPv6 or IPv4 and measure the latency (round-trip time).
Traceroute
Trace the network path from this server to another.
NsLookup / Dig
Look up various DNS records for domain names with this custom-crafted tool similar to the classic NSLookup and Dig commands.
How this site works


CentralOps.net
Address lookup
canonical name certifiedhacker.com.
aliases
addresses 162.241.216.11
Domain Whois record
Queried whois.internic.net with "dom certifiedhacker.com"...
Queried whois.networksolutions.com with "certifiedhacker.com"...


CentralOps.net
Utilities
About
Domain Dossier
Domain Check
Email Dossier
Browser Mirror
Ping
Traceroute
NsLookup / Dig


CentralOps.net
Utilities
About
Network Whois record
Queried rwhois.unifiedlayer.com with "162.241.216.11"...
Queried whois.arin.net with "n 162.241.216.11"...



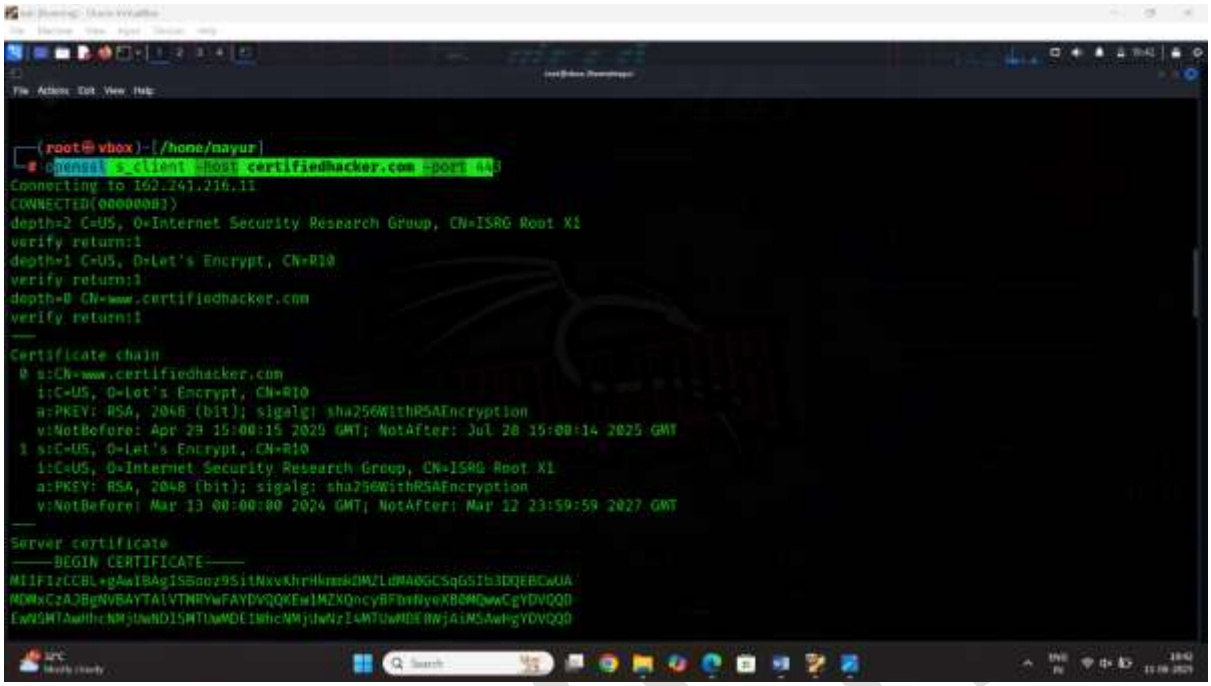
Task2 Banner Grabbing from SSL Service

Step1: open the kali Linux terminal and type the open SSL

Step2: type the command terminal

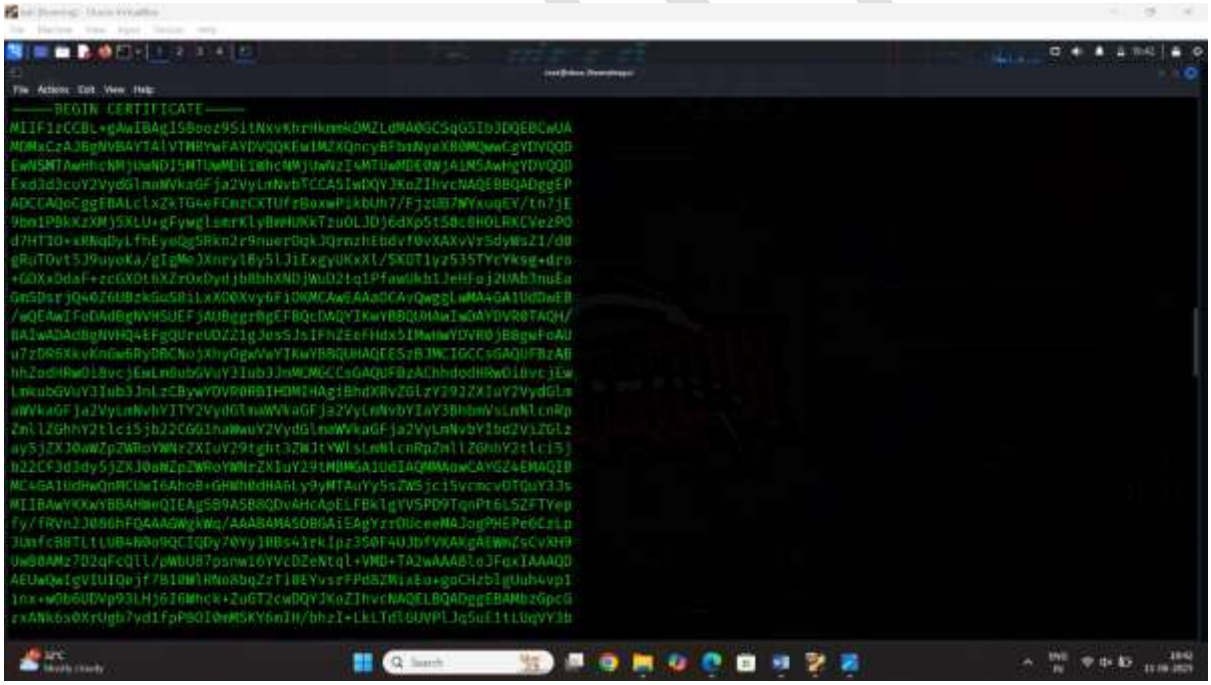
Command: openSSL s_client –host certifiedhacker.com –port 443

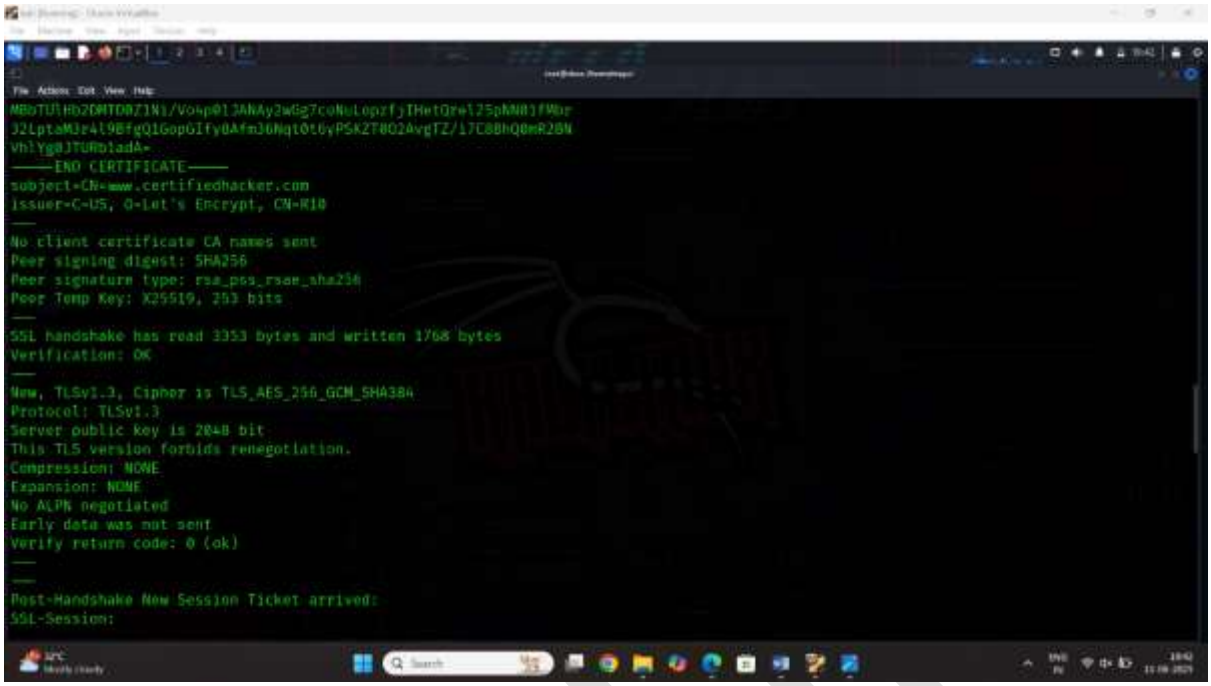
Result:


(root㉿vbox)-[/home/nayur]
openssl s_client -host certifiedhacker.com -port 443
Connecting to 162.241.216.11
CONNECTED(00000003)
depth=2 C=US, O=Internet Security Research Group, CN=ISRG Root X1
verify return:1
depth=1 C=US, O=Let's Encrypt, CN=R10
verify return:1
depth=0 CN=www.certifiedhacker.com
verify return:1

Certificate chain
0 s:CN=www.certifiedhacker.com
i:C=US, O=Let's Encrypt, CN=R10
a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
v:NotBefore: Apr 29 15:00:15 2025 GMT; NotAfter: Jul 28 15:00:14 2025 GMT
1 s:C=US, O=Let's Encrypt, CN=R10
i:C=US, O=Internet Security Research Group, CN=ISRG Root X1
a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
v:NotBefore: Mar 13 00:00:00 2024 GMT; NotAfter: Mar 12 23:59:59 2027 GMT

Server certificate
-----BEGIN CERTIFICATE-----
MIIF1zCCBL+gAwIBAgISBoz9SitNxvkhrHkmmkDMZLdMA0GCSqGSIb3DQEBCwUA
MDMxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCgYDVQQD
EwNSMTAwHhcNMjUwNDI5MTUwMDE1WhcNMjUwNzI4MTUwMDE0WjAiMSAwHgYDVQQD


-----BEGIN CERTIFICATE-----
MIIF1zCCBL+gAwIBAgISBoz9SitNxvkhrHkmmkDMZLdMA0GCSqGSIb3DQEBCwUA
MDMxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCgYDVQQD
EwNSMTAwHhcNMjUwNDI5MTUwMDE1WhcNMjUwNzI4MTUwMDE0WjAiMSAwHgYDVQQD
Exd3d3cuY2VydGlmaWVkaGFja2VyLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEP
ADCCAQoCggEBALclxZkTG6eFCmzCKTUfzBoxwPikbUh7/FjzUB7NYxuqEY/tn7jE
9bn1PBkKzXMj5XLU+gFywglsmrKlyBwHUKkTzuOLJDj6dXp5tS8c8HOLRkCVezPO
d7HTIO+xRNqDyLfhEyoQg5Rkn2r9nuerDqkJQrnzhtEbdvf0vXAXvVrSdyWsZ1/d0
gRuTDvt5J9uyoKa/gIgMeJXnry1By5lJiExgyUKxXl/SKOTlyz535TYcYksg+dro
+GOXx3daF+zcGXDLhXZrOxDydjb8bhXNDjWuO2tq1PfawUkb1JehFoj2UAb3nuEa
GmSDsrjQ4OZ6UBzkGuS8iLxXO0Xvy6FjOKMCAwEAAaOCAvQwggLwMA4GA1UdDwEB
/wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwDAYDVR0TAQH/
BAIwADAdBgNVHQ4EFgQUreUDZZIgJssSJsIFhZEefHdx5IMwHwYDVR0jBBgwFoAU
u7zDR6XkvKnGw6RyDBCNojXhyOgwVwYIKwYBBQUHAQEESzBJMCIGCCsGAQUFBzAB
hhZodHRwOi8vcjEwLmBubGVuY3Iub3JnMCMGCCsGAQUFBzAChhdodHRwOi8vcjEw
LmkubGVuY3Iub3JnLzCBywYDVR0RBIHDMIHAgiBndXRvZGlzY292ZXIuY2VydGlm
aWVkaGFja2VyLmNvbYITY2VydGlmaWVkaGFja2VyLmNvbYIaY3BhbmVsLmNlcnRp
ZmllZGhhY2tlci5jb22CGGlhaWwuY2VydGlmaWVkaGFja2VyLmNvbYIbd2ViZGlz
ay5jZXJOaWZpZWRoYWNrZXIuY29tght3ZWJtYWlsLmNlcnRpZmllZGhhY2tlci5j
b22CF3d3dy5jZXJ0aWZpZWRoYWNrZXIuY29tMBMGA1UdIAQMMAowCAYGZ4EMAQIB
MC4GA1UdHwQnMCUwI6AhoB+GHWh0dHA6Ly9yMTAuYy5sZWSjci5vcmcvDTQuY3Js
MIIBAwYKKwYBBAHWeQIEAgSB9ASB8QDvAHcApELFBklgYVSPD9TqnPt6LSZFTYep
fy/fRVnJJ86hFQAAAGWgkWg/AAAB4MASOBGAiEAgYzrOUceeMAJogPHEPe6CziD
3UnfcB8TLttUB4W0o9QCIQDy70Yy1BBs41rkIpz3S0F4UJbfVKAKgAEWmZsCvXH9
UwB0AMz7D2qFcQll/pWbU87psnwl6YVcDZeNtql+VMD+TA2wAAAB1oJFqxIAAAQD
AEUwQwIgVIUIQpjf7B10WlHNo8bqZzTi0EYvsrFPdBZMiaEo+goCHzblgUuh4vp1
iox+wQb6UDVp93LHj6IGWhck+ZuGT2cwDQYJKoZIhvcNAQELBQADggEBAMbzGpcG
rxANkbs0XrUgb7vd1fpPBOI0mM5KY6mIH/bhzI+LkLTdlGUVPlJq5uE1tLUqVY3b

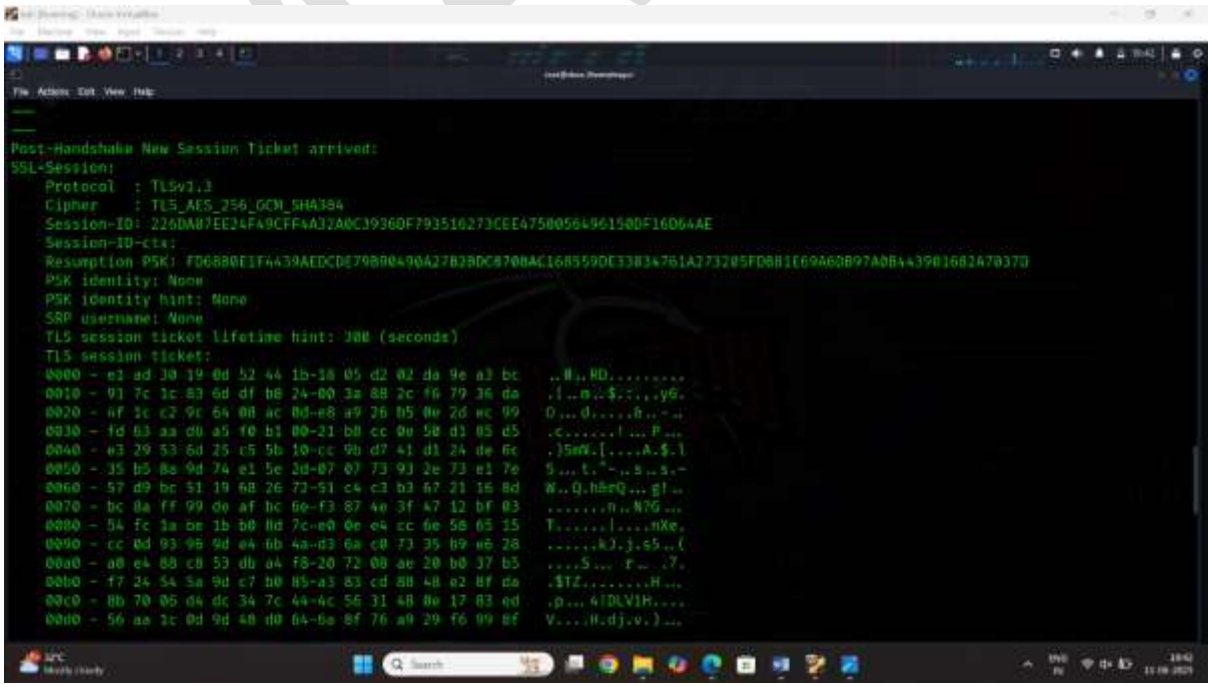

MBbTU\HD2DMTDBZ1N1/Vo4p01JANAy2wGg7coNuLopzfjIHetQre125pNN01fMDr
32LptaMJr4l9BfgQ1GopGIfy0AfmJ6Nqt0t6yPSkZT8O2AvgTZ/l7C8BhQ0mR2BN
vhlYg8JTURbladA=
-----END CERTIFICATE-----
subject=CN=www.certifiedhacker.com
issuer=C=US, O=Let's Encrypt, CN=R10

No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: rsa_pss_rsae_sha256
Peer Temp Key: X25519, 253 bits

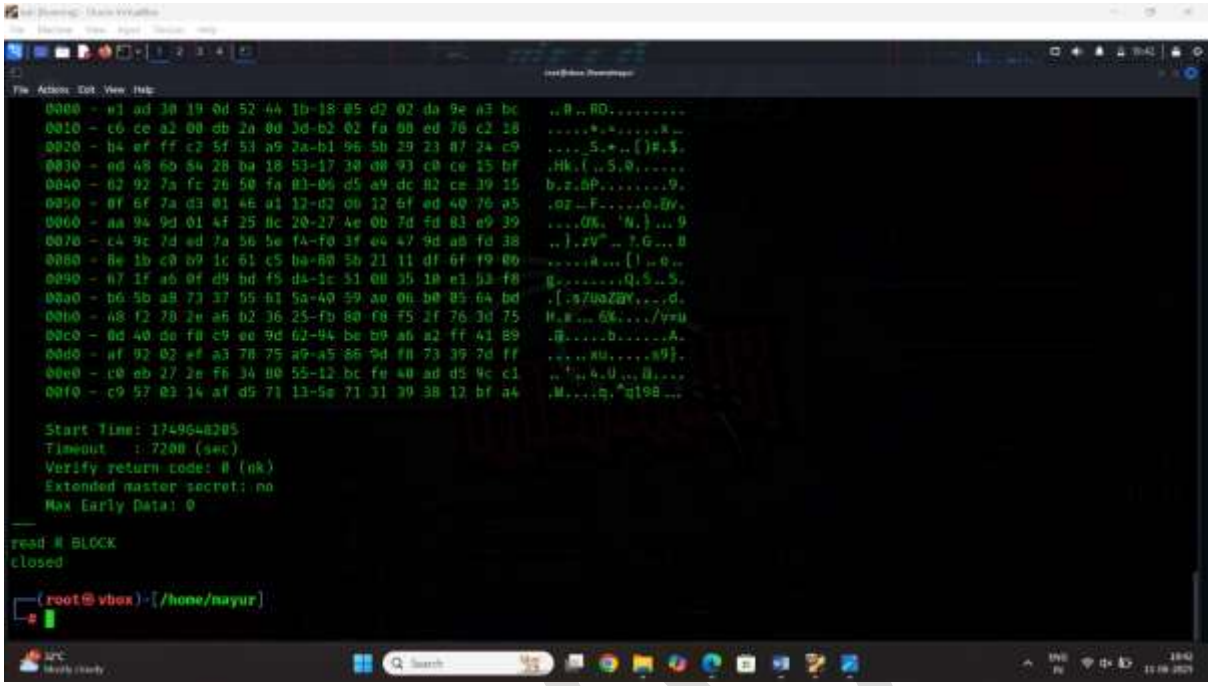
SSL handshake has read 3353 bytes and written 1768 bytes
Verification: OK

New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Protocol: TLSv1.3
Server public key is 2048 bit
This TLS version forbids renegotiation.
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)

Post-Handshake New Session Ticket arrived:
SSL-Session:



Post-Handshake New Session Ticket arrived:
SSL-Session:
Protocol : TLSv1.3
Cipher : TLS_AES_256_GCM_SHA384
Session-ID: 226DA87EE24FA9CFF4A32A0C3936DF79351627 3CEE4758056496150DF16D64AE
Session-ID-ctx:
Resumption PSK: FD6880E1F4439AEDCDE79B0490A27B28DC870BAC168559DE33834761A273205FDB81E69A60B97A0B4439016824703 7D
PSK identity: None
PSK identity hint: None
SRP username: None
TLS session ticket lifetime hint: 300 (seconds)
TLS session ticket:
0000 - e1 ad 30 19 0d 52 44 1b-10 05 d2 02 da 9e a3 bc ..0..RD.........
0010 - 91 7c 1c 83 6d df b8 24-00 3a 88 2c f6 79 36 da .|..m..$.:.,.y6.
0020 - af 1c c2 9c 64 08 ac 8d-e8 a9 26 b5 0e 2d ec 99 ....d.....&..-..
0030 - fd 63 aa d0 a5 f0 b1 00-21 b8 cc 9e 50 d1 05 d5 .c......!...P...
0040 - e3 29 53 6d 25 c5 5b 10-cc 9b d7 41 d1 24 de 6c .)Sm%.[....A.$.l
0050 - 35 b5 8e 9d 74 e1 5e 2d-07 07 73 93 2e 73 e1 7e 5...t.^-..s..s.~
0060 - 57 d9 bc 51 19 68 26 72-51 c4 c3 b3 67 21 16 8d W..Q.h&rQ...g!..
0070 - bc 0a ff 99 de af bc 60-f3 87 4e 3f a7 12 bf 03 ....`..N?G...
0080 - 54 fc 1a be 1b b0 8d 7c-e0 0e e4 cc 6e 58 65 15 T......|....nXe.
0090 - cc 0d 93 95 9d e4 6b 4a-d3 6a c0 73 35 b9 e6 28 ......kJ.j.s5..(
00a0 - a0 e4 88 c8 53 db a4 f8-20 72 08 ae 20 b0 37 b5 ....S... r.. .7.
00b0 - f7 24 54 5a 9d c7 b0 85-a3 83 cd 88 48 e2 8f da .$TZ........H...
00c0 - 8b 70 05 d4 dc 34 7c 44-4c 56 31 48 8e 17 83 ed .p...4|DLV1H....
00d0 - 56 aa 1c 0d 9d 48 d0 64-6e 8f 76 a9 29 f6 09 8f V....H.dj.v.)...

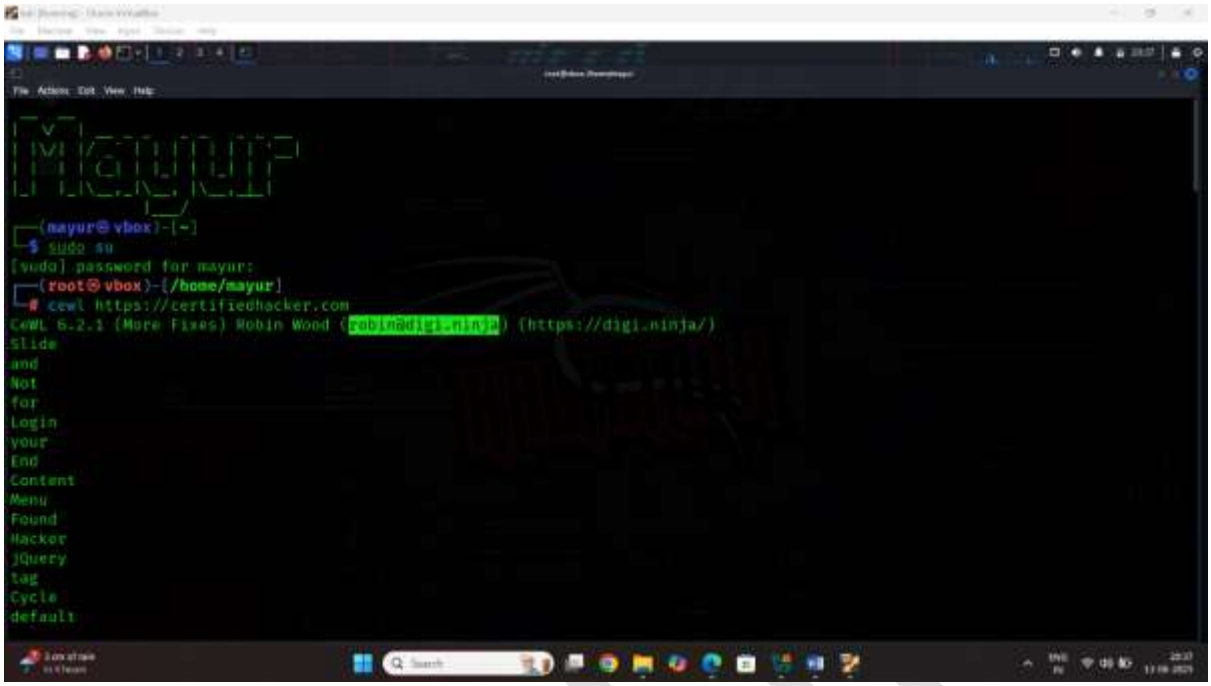


Method 1 Gathering the wordlist from the Target website

Step1: start the kali linux open the terminal type the command

Command: cewl https://www.certifiedhacker.com

Result:



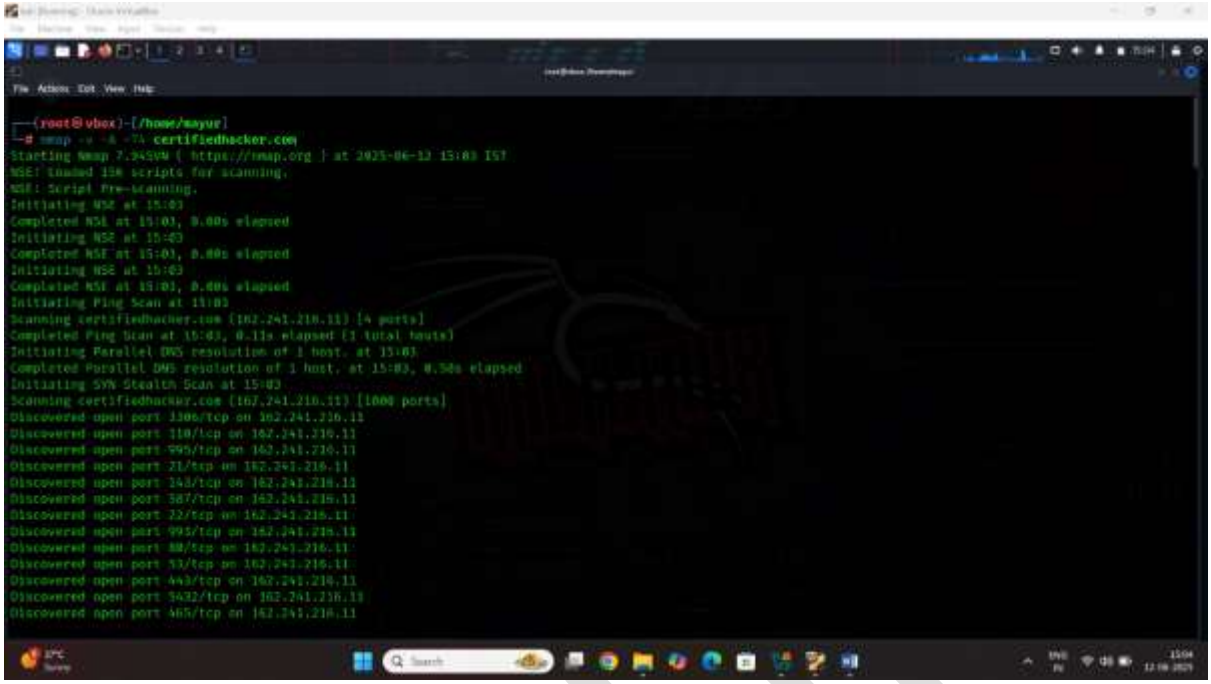
Task 3 identifiye the web application port and servic discovery with nmap

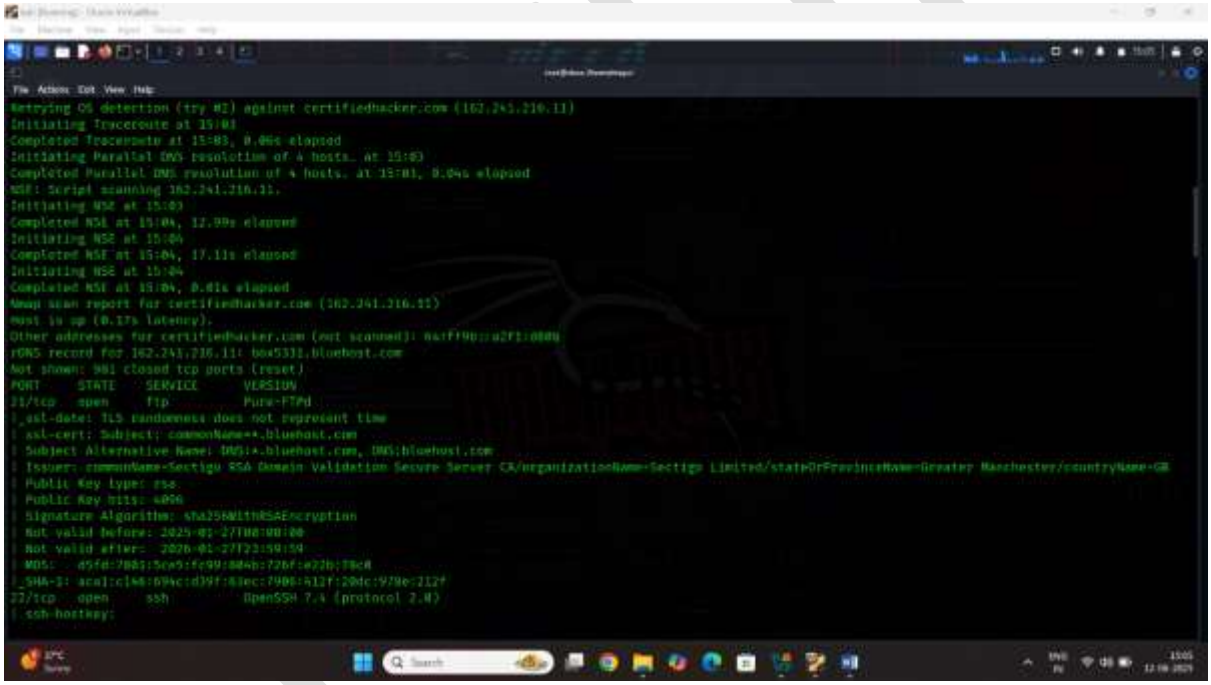
Step1 start the kali Linux open the terminal and type the command:

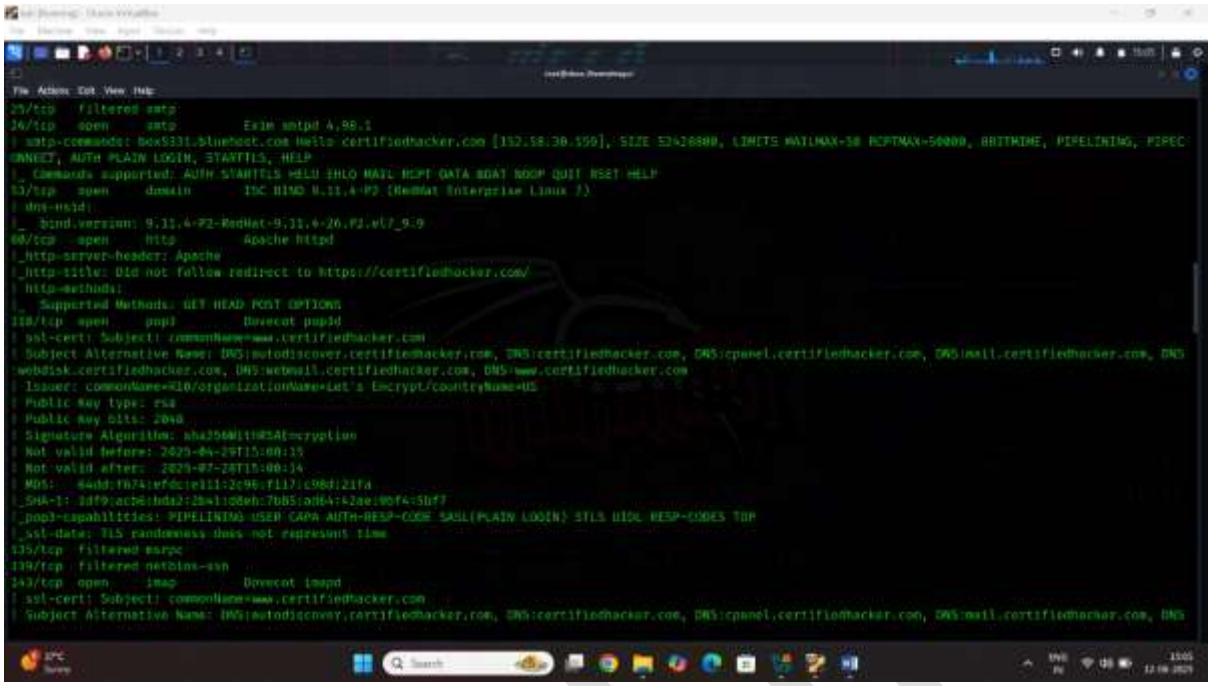
Nmap –v –A –T4 certifiedhacker.com

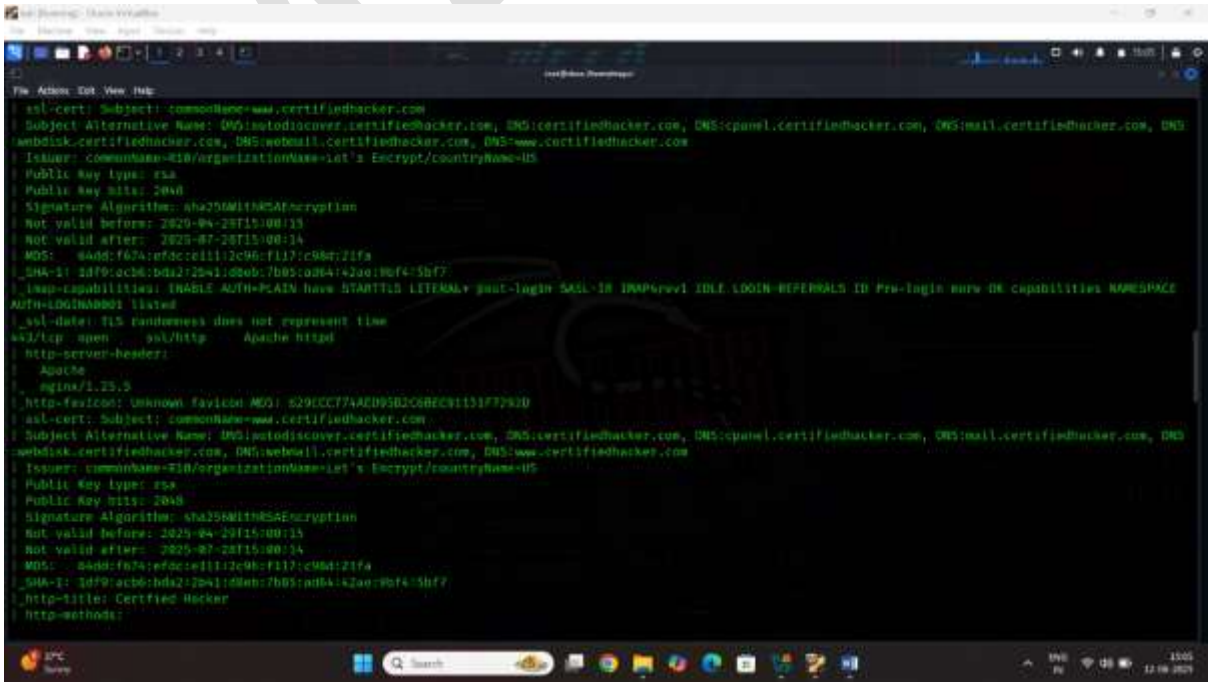
This command all information show of the target

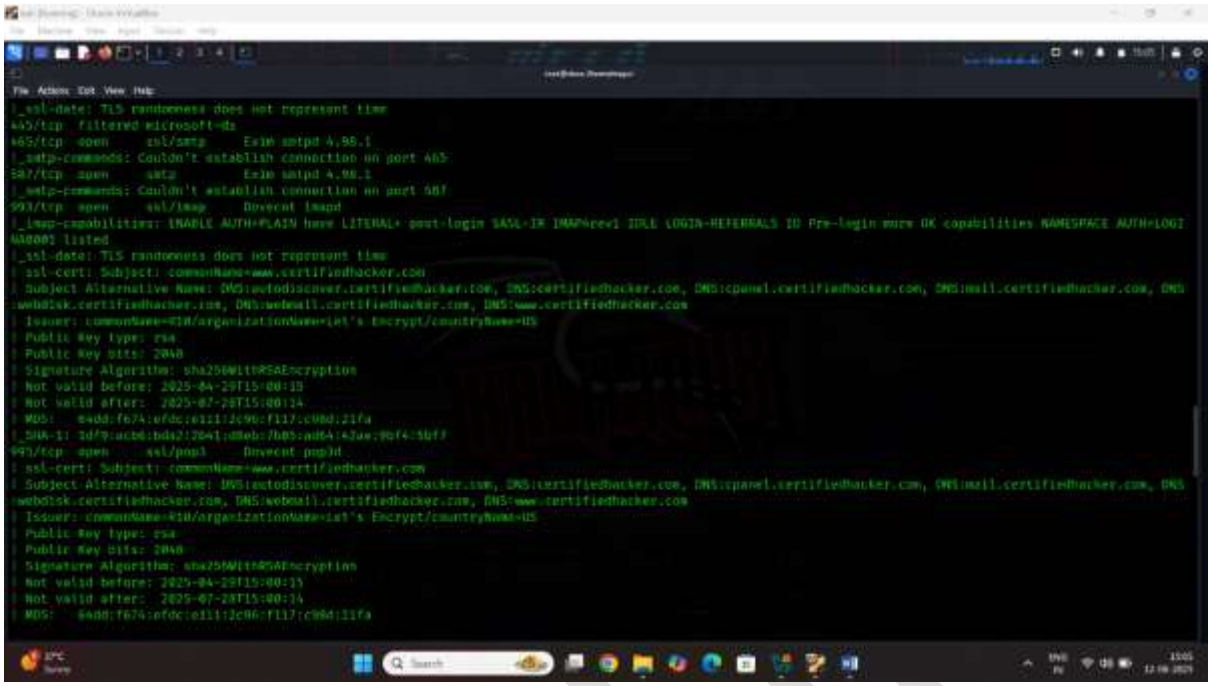
Result:

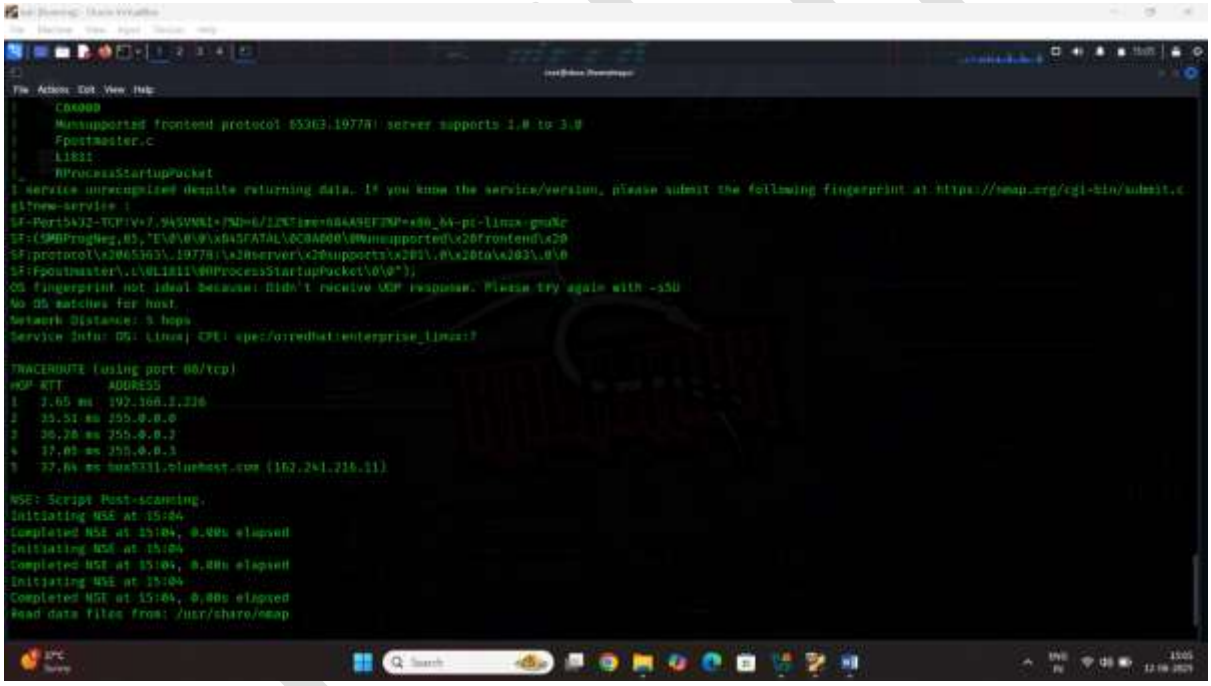

File Actions Edit View Help
┌──(root㉿vbox)-[/home/mayur]
└─# nmap -v -A -T4 certifiedhacker.com
Starting Nmap 7.945VN (https://nmap.org) at 2025-06-12 15:03 IST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 15:03
Completed NSE at 15:03, 0.00s elapsed
Initiating NSE at 15:03
Completed NSE at 15:03, 0.00s elapsed
Initiating NSE at 15:03
Completed NSE at 15:03, 0.00s elapsed
Initiating Ping Scan at 15:03
Scanning certifiedhacker.com (162.241.216.11) [4 ports]
Completed Ping Scan at 15:03, 0.11s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:03
Completed Parallel DNS resolution of 1 host. at 15:03, 0.50s elapsed
Initiating SYN Stealth Scan at 15:03
Scanning certifiedhacker.com (162.241.216.11) [1000 ports]
Discovered open port 3306/tcp on 162.241.216.11
Discovered open port 110/tcp on 162.241.216.11
Discovered open port 995/tcp on 162.241.216.11
Discovered open port 21/tcp on 162.241.216.11
Discovered open port 143/tcp on 162.241.216.11
Discovered open port 587/tcp on 162.241.216.11
Discovered open port 22/tcp on 162.241.216.11
Discovered open port 993/tcp on 162.241.216.11
Discovered open port 80/tcp on 162.241.216.11
Discovered open port 53/tcp on 162.241.216.11
Discovered open port 443/tcp on 162.241.216.11
Discovered open port 5432/tcp on 162.241.216.11
Discovered open port 465/tcp on 162.241.216.11


File Actions Edit View Help
Retrying OS detection (try #2) against certifiedhacker.com (162.241.216.11)
Initiating Traceroute at 15:03
Completed Traceroute at 15:03, 0.06s elapsed
Initiating Parallel DNS resolution of 4 hosts. at 15:03
Completed Parallel DNS resolution of 4 hosts. at 15:03, 0.04s elapsed
NSE: Script scanning 162.241.216.11.
Initiating NSE at 15:03
Completed NSE at 15:04, 12.99s elapsed
Initiating NSE at 15:04
Completed NSE at 15:04, 17.11s elapsed
Initiating NSE at 15:04
Completed NSE at 15:04, 0.01s elapsed
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.17s latency).
Other addresses for certifiedhacker.com (not scanned): 0x1f10:::a2f1:d00b
rDNS record for 162.241.216.11: box5331.bluehost.com
Not shown: 981 closed tcp ports (reset)
PORT STATE SERVICE VERSION
21/tcp open ftp Pure-FTPd
| ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=*.bluehost.com
| Subject Alternative Name: DNS:*.bluehost.com, DNS:bluehost.com
| Issuer: commonName=Sectigo RSA Domain Validation Secure Server CA/organizationName=Sectigo Limited/stateOrProvinceName=Greater Manchester/countryName=GB
| Public Key type: rsa
| Public Key bits: 4096
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2025-01-27T00:00:00
| Not valid after: 2026-01-27T23:59:59
| MD5: d5fd:70d1:5ce5:fc99:804b:726f:e22b:78c8
|_SHA-1: aca1:c1e6:69cc:d39f:61ec:7906:412f:20dc:9798:212f
22/tcp open ssh OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:


File Actions Edit View Help
25/tcp filtered smtp
26/tcp open smtp Exim smtpd 4.98.1
| smtp-commands: box5331.bluehost.com Hello certifiedhacker.com [152.58.30.159], SIZE 52428800, LIMITS MAILMAX=50 RCPTMAX=50000, 8BITMIME, PIPELINING, PIPEC
ONNECT, AUTH PLAIN LOGIN, STARTTLS, HELP
|_ Commands supported: AUTH STARTTLS HELO EHLO MAIL RCPT DATA BDAT NOOP QUIT RSET HELP
53/tcp open domain ISC BIND 9.11.4-P2 (RedHat Enterprise Linux 7)
| dns-nsid:
|_ bind.version: 9.11.4-P2-RedHat-9.11.4-26.P2.el7_9.9
80/tcp open http Apache httpd
|_http-server-header: Apache
|_http-title: Did not follow redirect to https://certifiedhacker.com/
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
110/tcp open pop3 Dovecot pop3d
| ssl-cert: Subject: commonName=www.certifiedhacker.com
| Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:mail.certifiedhacker.com, DNS
:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com
| Issuer: commonName=R10/organizationName=Let's Encrypt/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2025-04-29T15:00:15
| Not valid after: 2025-07-28T15:00:14
| MD5: 44dd:f674:efdc:e111:2c96:f117:c98d:21fa
|_SHA-1: 1df9:acb6:bda2:2b41:d8eb:7b85:ad64:42ae:90f4:5bf7
|_pop3-capabilities: PIPELINING USER CAPA AUTH-RESP-CODE SASL(PLAIN LOGIN) STLS UIDL RESP-CODES TOP
|_ssl-date: TLS randomness does not represent time
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
143/tcp open imap Dovecot imapd
| ssl-cert: Subject: commonName=www.certifiedhacker.com
| Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:mail.certifiedhacker.com, DNS


File Actions Edit View Help
| ssl-cert: Subject: commonName=www.certifiedhacker.com
| Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:mail.certifiedhacker.com, DNS
:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com
| Issuer: commonName=R10/organizationName=Let's Encrypt/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2025-04-29T15:00:15
| Not valid after: 2025-07-28T15:00:14
| MD5: 44dd:f674:efdc:e111:2c96:f117:c98d:21fa
| SHA-1: 1df9:acb6:bda2:2b41:d8eb:7b85:ad64:42ae:90f4:5bf7
|_imap-capabilities: ENABLE AUTH=PLAIN have STARTTLS LITERAL+ post-login SASL-IR IMAPrev1 IDLE LOGIN-REFERRALS ID Pre-login more OK capabilities NAMESPACE
AUTH=LOGINA0001 listed
|_ssl-date: TLS randomness does not represent time
443/tcp open ssl/http Apache httpd
| http-server-header:
| Apache
|_ nginx/1.25.5
| http-favicon: Unknown favicon MD5: 629CCC774AED95B2C68EC911317F7930
| ssl-cert: Subject: commonName=www.certifiedhacker.com
| Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:mail.certifiedhacker.com, DNS
:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com
| Issuer: commonName=R10/organizationName=Let's Encrypt/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2025-04-29T15:00:15
| Not valid after: 2025-07-28T15:00:14
| MD5: 44dd:f674:efdc:e111:2c96:f117:c98d:21fa
| SHA-1: 1df9:acb6:bda2:2b41:d8eb:7b85:ad64:42ae:90f4:5bf7
|_http-title: Certified Hacker
| http-methods:



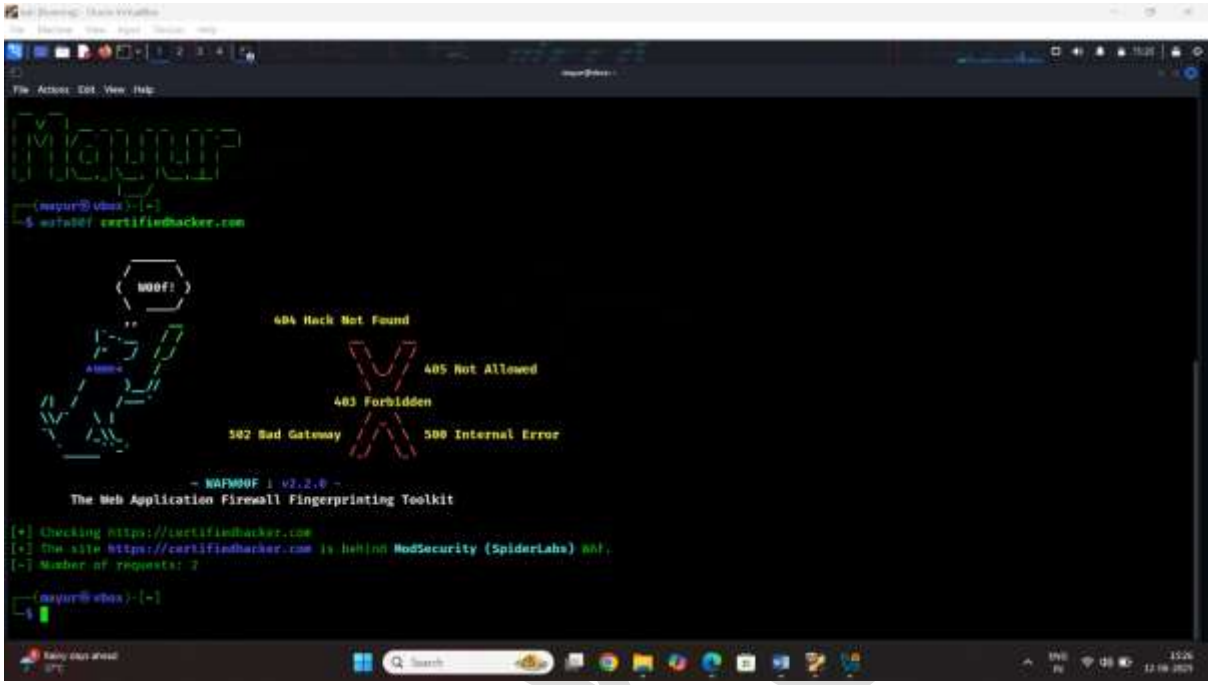


Task4 Detecting web application firewall using wafw00f

Step1 start the kali linux open the terminal and type the wafw00f

Command: wafw00f certifiedhacker.com

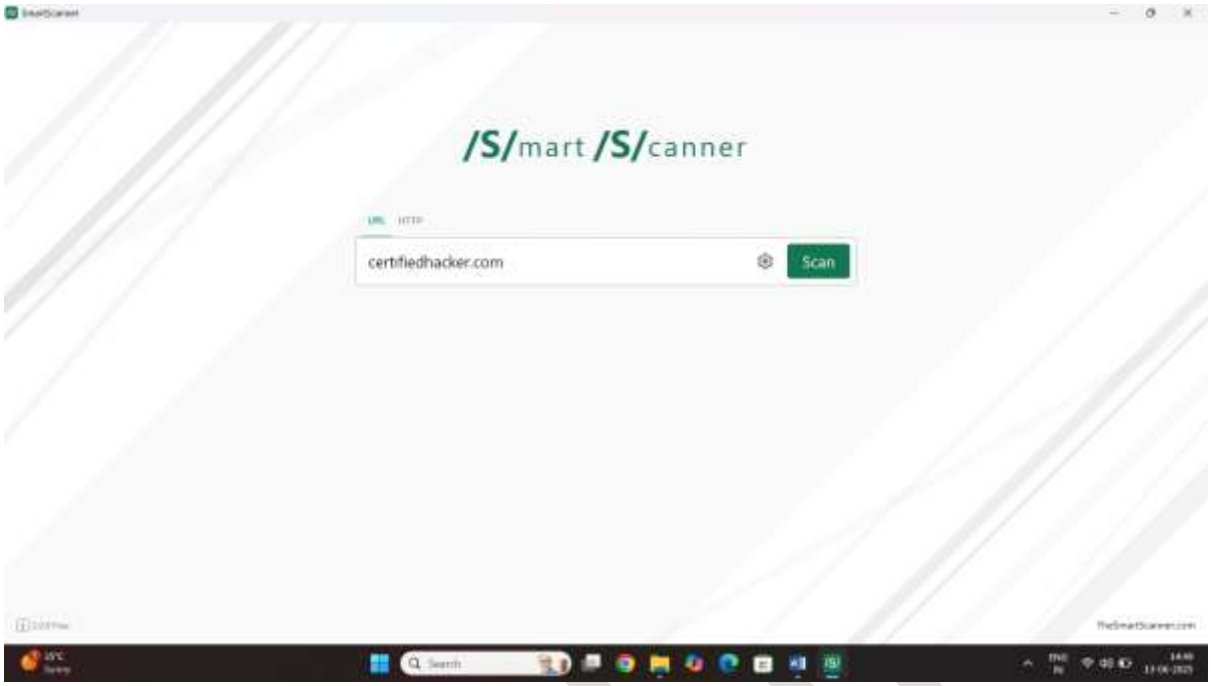
Result:



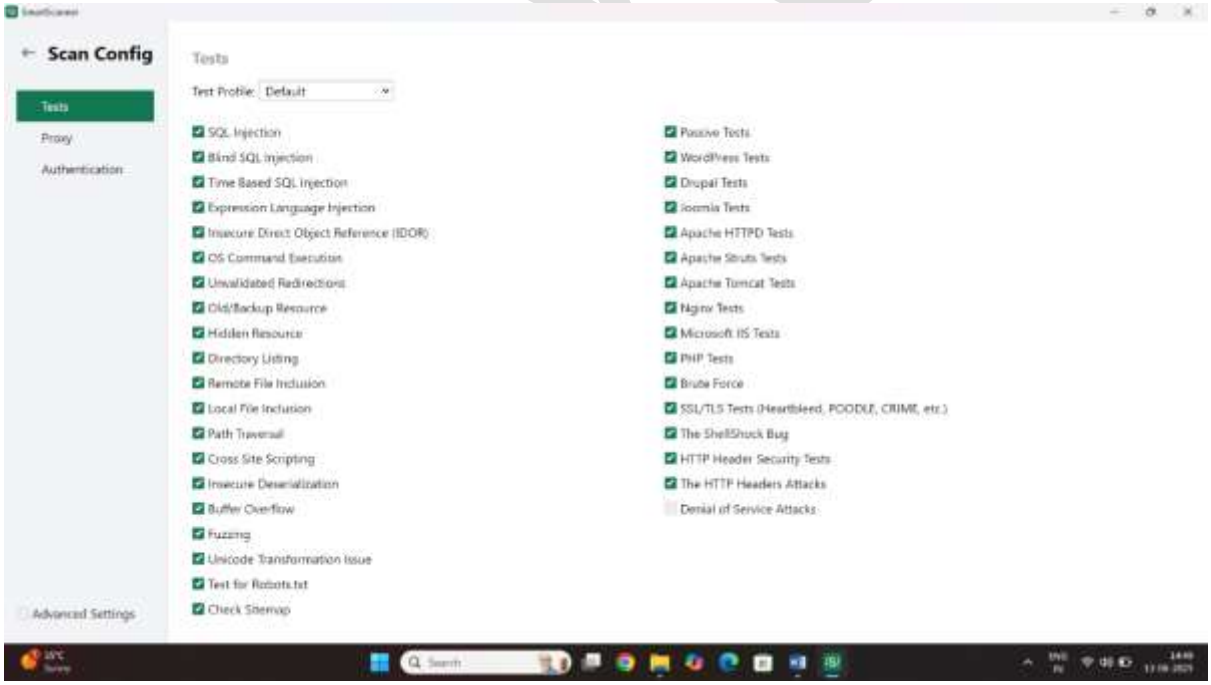
Task 5 perfrom web app application vulnerability using smart scanner

Step1: open the smart scanner type the URL target web site

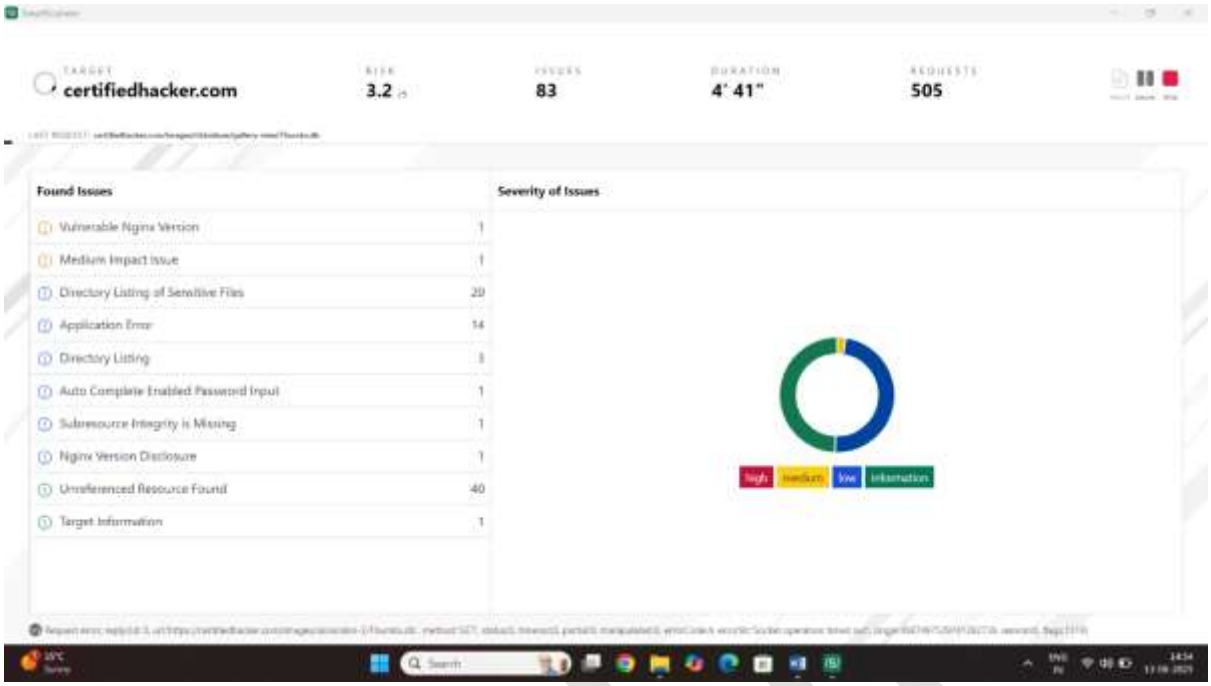
Eg certifiedhacker.com



Step2: click on the setting option you choice type of scan



Step3 I am select the defult choice but you can choice the client riquerment choice



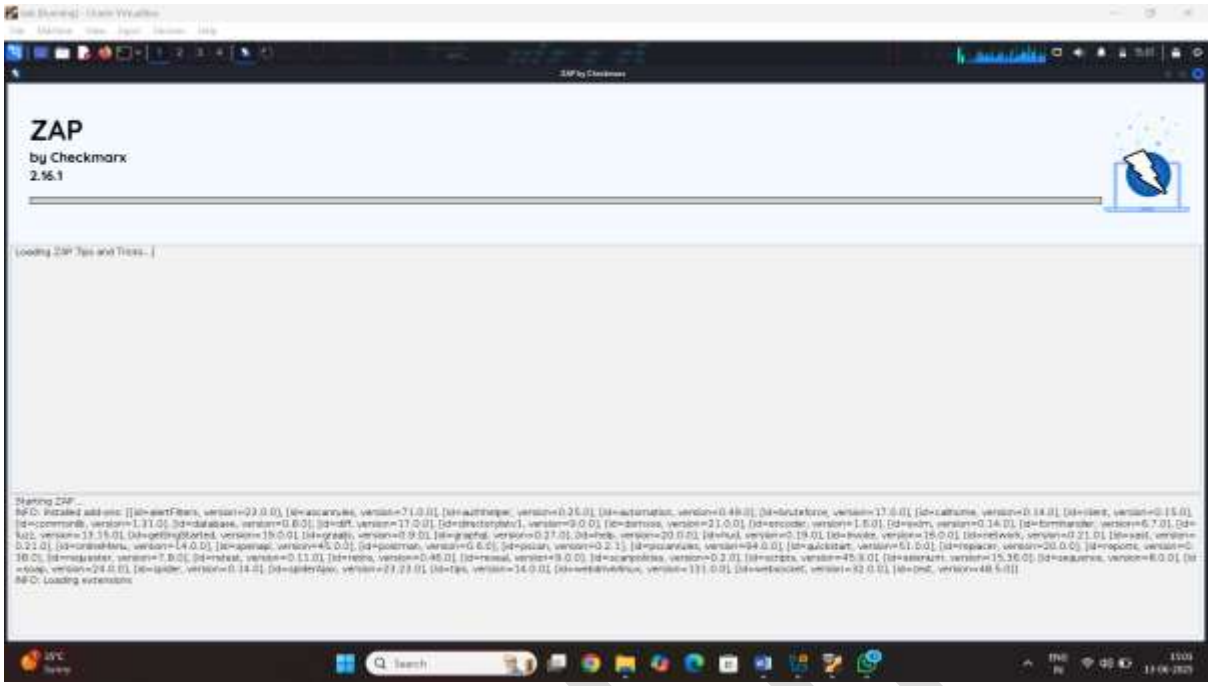
Task 6 perfrom web app application vulnerability using Acunitix

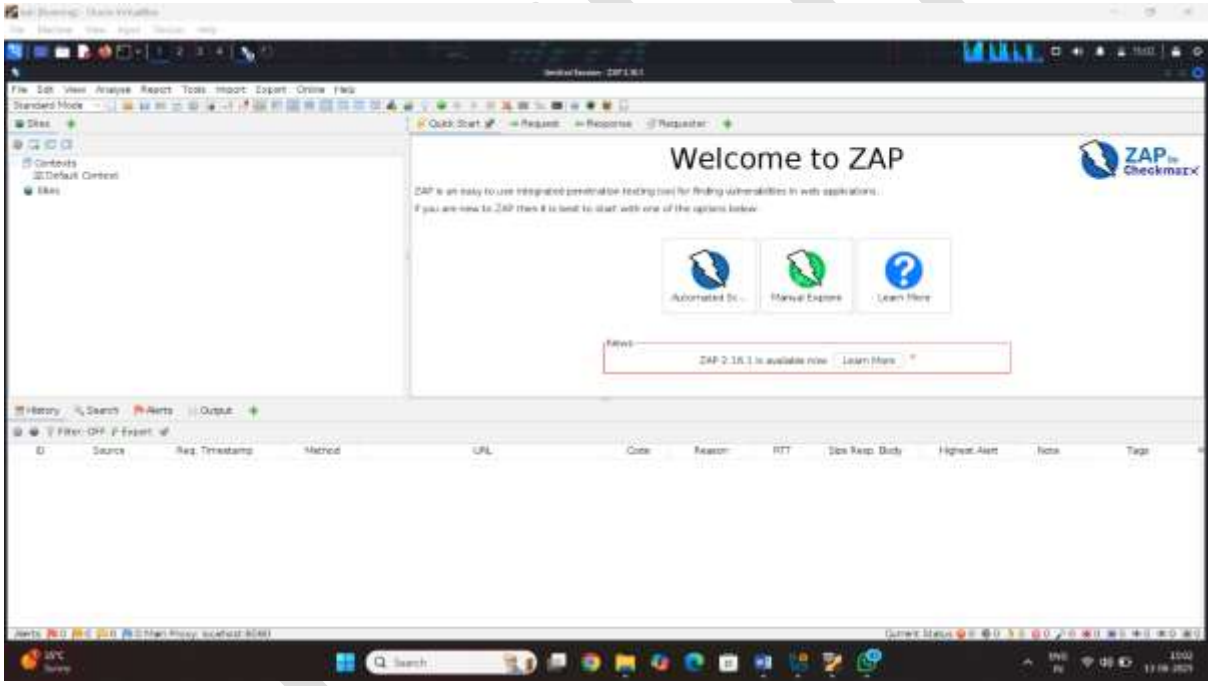
Step1: start the kali Linux open the Acunitix scanner

Task 7 perfrom web app application vulnerability using Zaproxy

Step1: start the kali Linux open the terminal

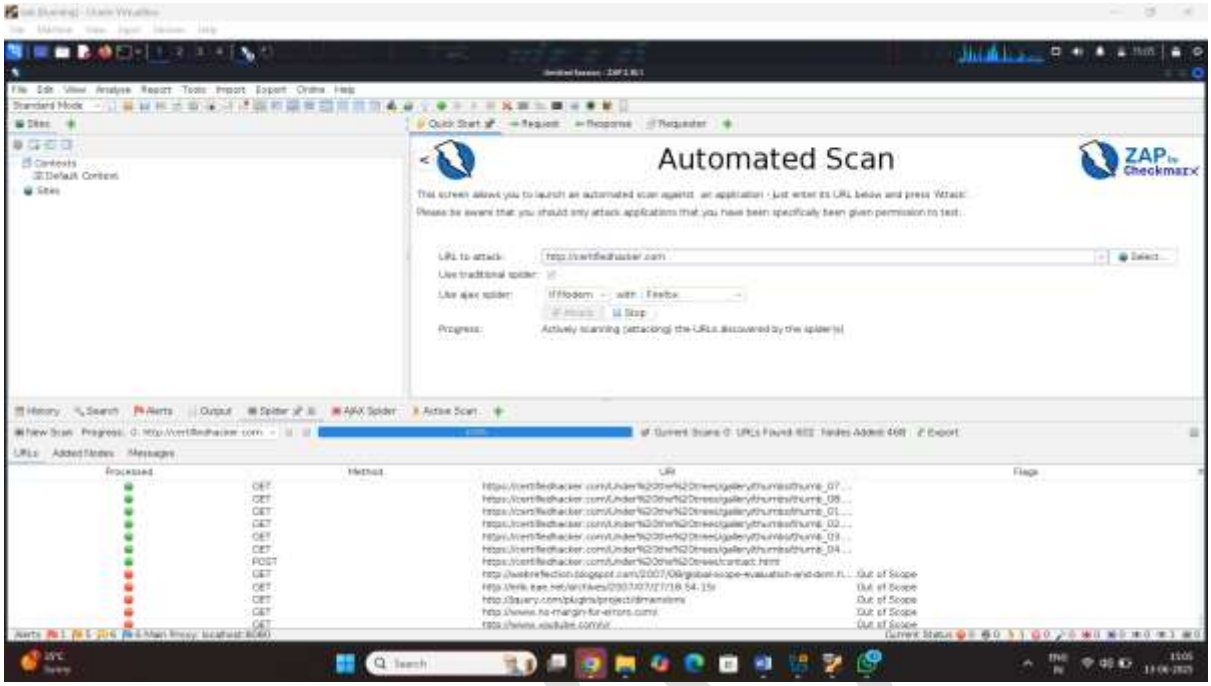
Start the zaproxy



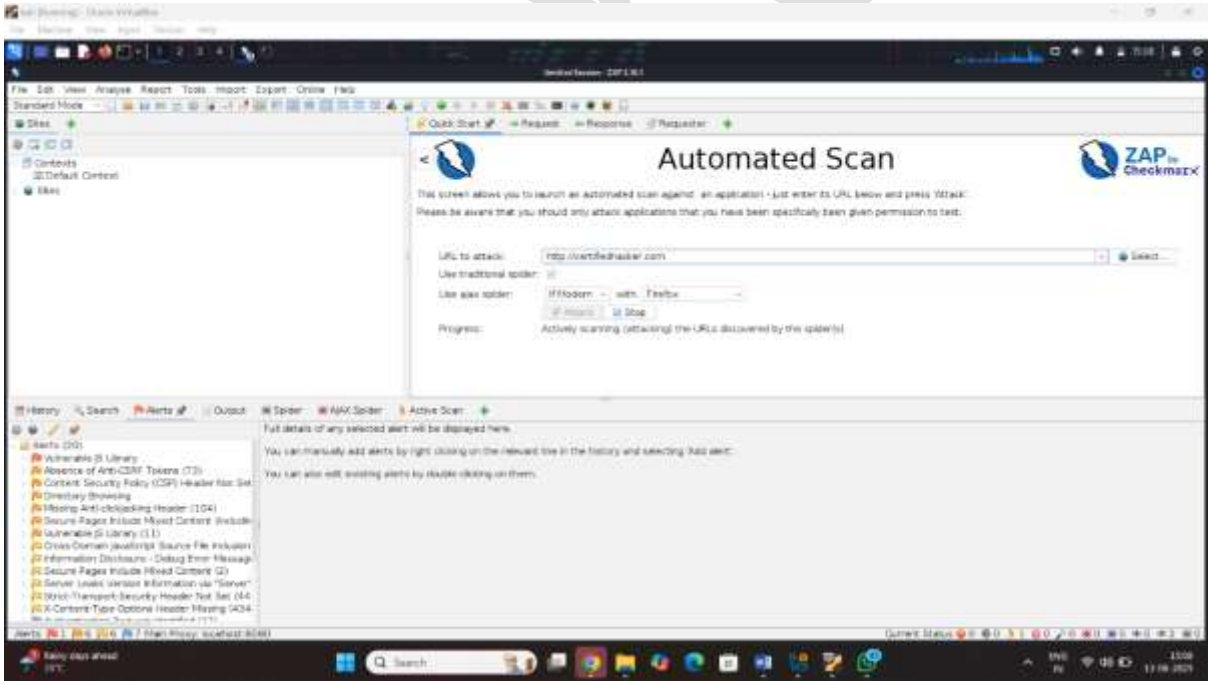


Step2: click on the autometed scanner

Step3: click on the attack and start the attack



Specific alert and critical issu



Task 8 Web application attack methodlogy

Types of input validation attack

1. Buffer Overflow Input Validation:

Buffer Overflow is a type of Input Validation Attack that makes the computer system unresponsive by overloading it with a huge chunk of information. The huge chunks result in successive memory consumption and occupy a great part of computer memory.

2. Canonical Ideation Input Validation Attack:

Canonical Ideation is a type of Input Validation Attack is caused as a result of changing the file path that had secure access to secure information. Thereby making the secure and sensitive information accessible to unauthorized users to view, make changes, and even steal private sensitive information as and when required.

3. XSS Attack:

XSS Attacks are cross-site scripting attacks where a suspicious link is placed alongside the valid legitimate URLs. The user is unable to detect or distinguish between the

legitimate and malicious user link and unknowingly becomes a victim of the XSS Input Validation attack.

4. SQL Injection Attack: SQL Injection is another type of Input Validation Attack, involving the phenomenon where the public URL is tampered with by the injection of SQL code in the Public URL. The hacker injects the code with the purpose to allow actions such as copying of confidential user data, manipulating sensitive information, and purposely deleting significant important information.

Types of web application attack

1. **SQL Injection (SQLi)**
   Attackers insert malicious SQL queries into input fields to access, modify, or delete database content.
   It can expose sensitive data like usernames, passwords, or entire databases.
2. **Cross-Site Scripting (XSS)**
   Malicious scripts are injected into trusted websites to run in other users' browsers.
   This can lead to session hijacking, defacement, or redirection to malicious sites.

3. **Cross-Site Request Forgery (CSRF)**
   Tricks a user's browser into sending unauthorized requests to a web app they are authenticated in.
   It can lead to unwanted actions like changing passwords or making purchases.
4. **Remote Code Execution (RCE)**
   Attackers exploit vulnerabilities to execute arbitrary code on the server.
   It can give full control over the web server, allowing data theft or takeover.
5. **Directory Traversal**
   By manipulating URL paths, attackers access restricted files and directories.
   This exposes sensitive configuration files or passwords stored on the server.
6. **Local File Inclusion (LFI)**
   Loads local files through vulnerable scripts by modifying input parameters.
   Can expose critical files like `/etc/passwd` or application logs.
7. **Remote File Inclusion (RFI)**
   Includes external files via URL input, often executing them on the server.
   It allows attackers to run remote malicious scripts.

8. **Command Injection**
   Attackers inject OS commands through web input fields that get executed by the server.
   This may lead to full system compromise if input is not properly sanitized.
9. **Broken Authentication**
   Weak or flawed login systems allow attackers to bypass authentication or hijack sessions.
   It can lead to unauthorized access to user accounts and data.
10. **Insecure Direct Object References (IDOR)**
    Users can access or modify data by altering input like user IDs in the URL.
    If access control is missing, this can lead to data leakage or manipulation.
11. **Security Misconfiguration**
    Improper server, app, or database settings expose vulnerabilities.
    Default credentials, open ports, or verbose error messages are common issues.
12. **XML External Entity (XXE) Injection**
    Malicious XML input exploits parsers to access internal files or services.
    It may lead to data disclosure or server-side request forgery.

13. **Server-Side Request Forgery (SSRF)**
    Forces the server to make requests to internal or external systems.
    Can be used to scan internal networks or access restricted services.
14. **Session Hijacking**
    Attackers steal or predict session tokens to impersonate legitimate users.
    They gain unauthorized access to user accounts or sensitive actions.
15. **Clickjacking**
    A malicious overlay tricks users into clicking hidden buttons or links.
    Users may unknowingly approve actions or reveal sensitive data.
16. **Path Disclosure**
    Error messages reveal server file paths or internal structure.
    Attackers use this to plan further targeted attacks.
17. **Unrestricted File Upload**
    Uploading malicious files like web shells due to poor file validation.
    Once uploaded, attackers can execute arbitrary commands on the server.
18. **HTTP Host Header Attack**
    Modifying the Host header to poison cache or

bypass access controls.
Can be used in phishing or cache poisoning attacks.

19. **Web Cache Poisoning**
Injects malicious content into a cache to be served to users.
Spreads malicious scripts or fake content to legitimate users.

20. **Business Logic Flaws**
Exploits weaknesses in how an application functions or enforces rules.
Can be used to bypass pricing rules, gain free items, or escalate privileges

What is burp suite

Burp Suite is a widely used web security testing tool by ethical hackers and testers.
It helps detect and exploit vulnerabilities in web applications.
The tool functions as an intercepting proxy between the browser and web server.
Users can inspect, intercept, and modify HTTP/S requests and responses.
It includes tools like Proxy, Scanner, Intruder, Repeater, Decoder, and Comparer.

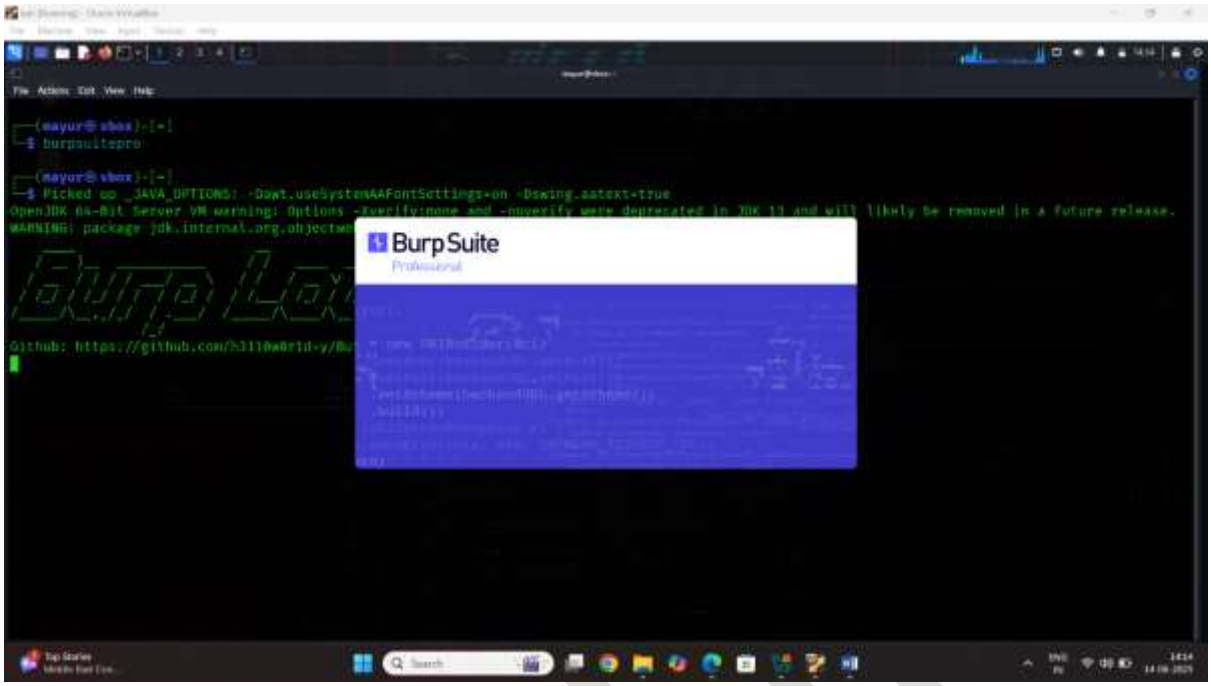
The Community Edition is free but limited; the Pro version offers scanning.
Intruder allows brute force and fuzz testing, while Repeater is for manual testing.
Decoder helps analyze encoded data like Base64 or URL encoding.
Burp Suite is essential for web app penetration testing and vulnerability assessment.

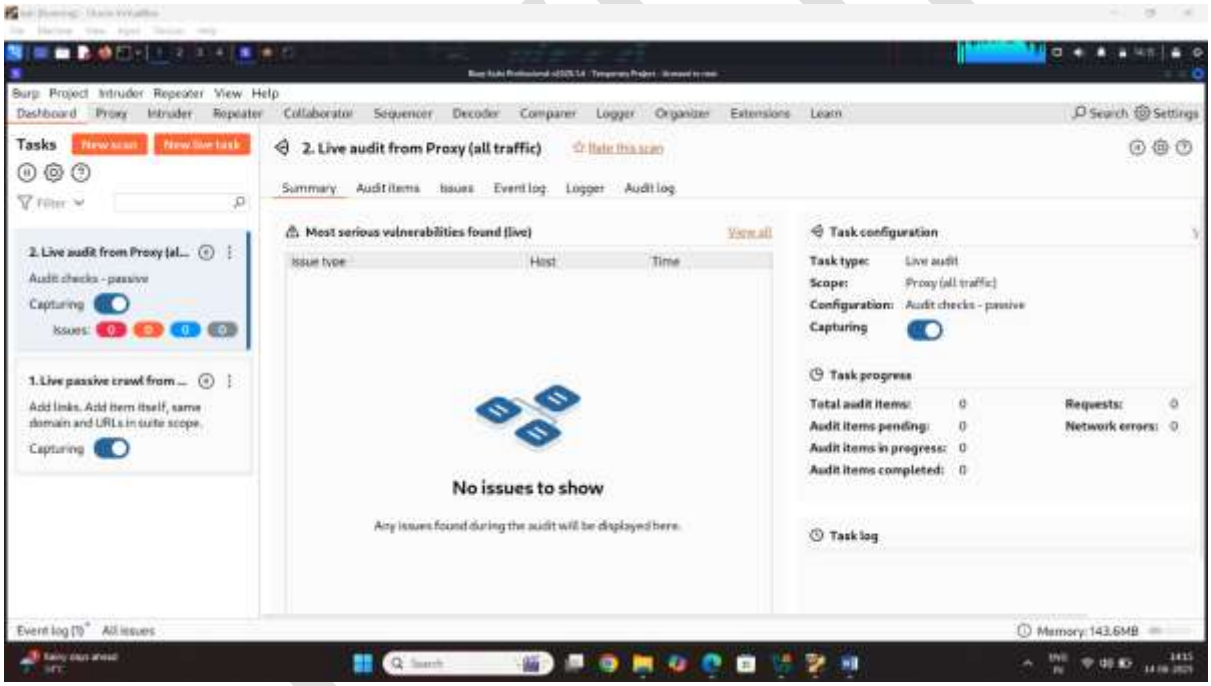
Task 9 how to test web application using sniper burp suite using sql injection attack

I am select the testing web site testfire.net

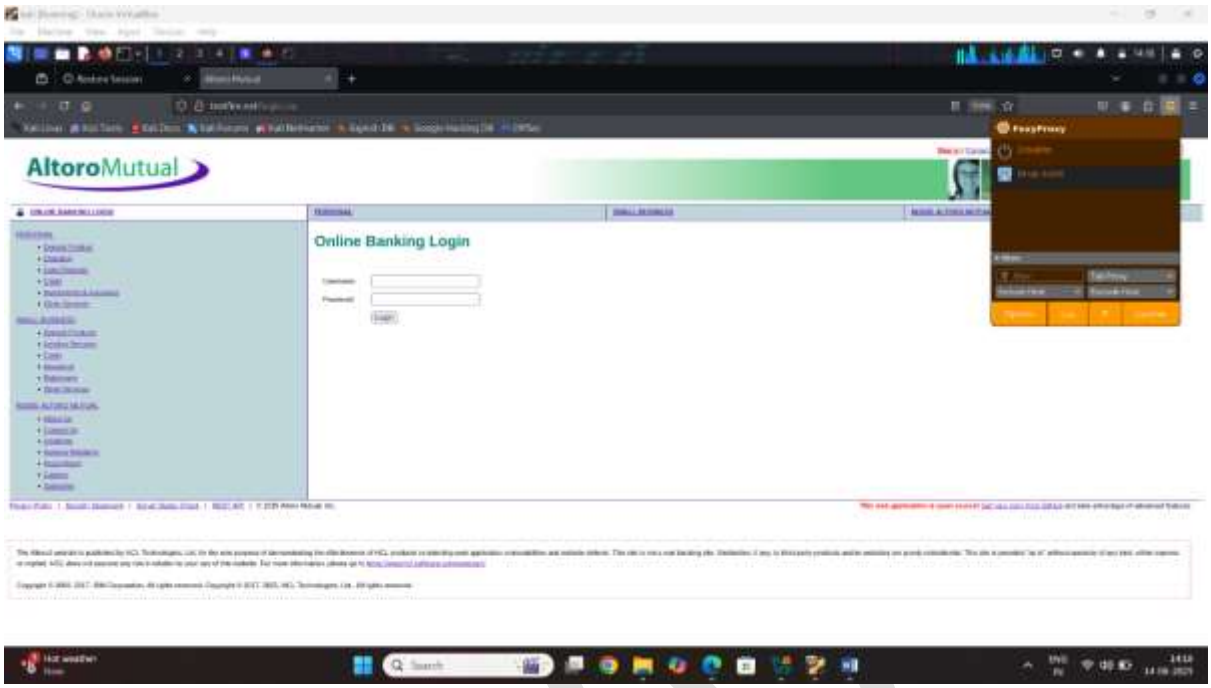
How to test username

Step1: start the kali Linux machine open the terminal search then burpsuitepro



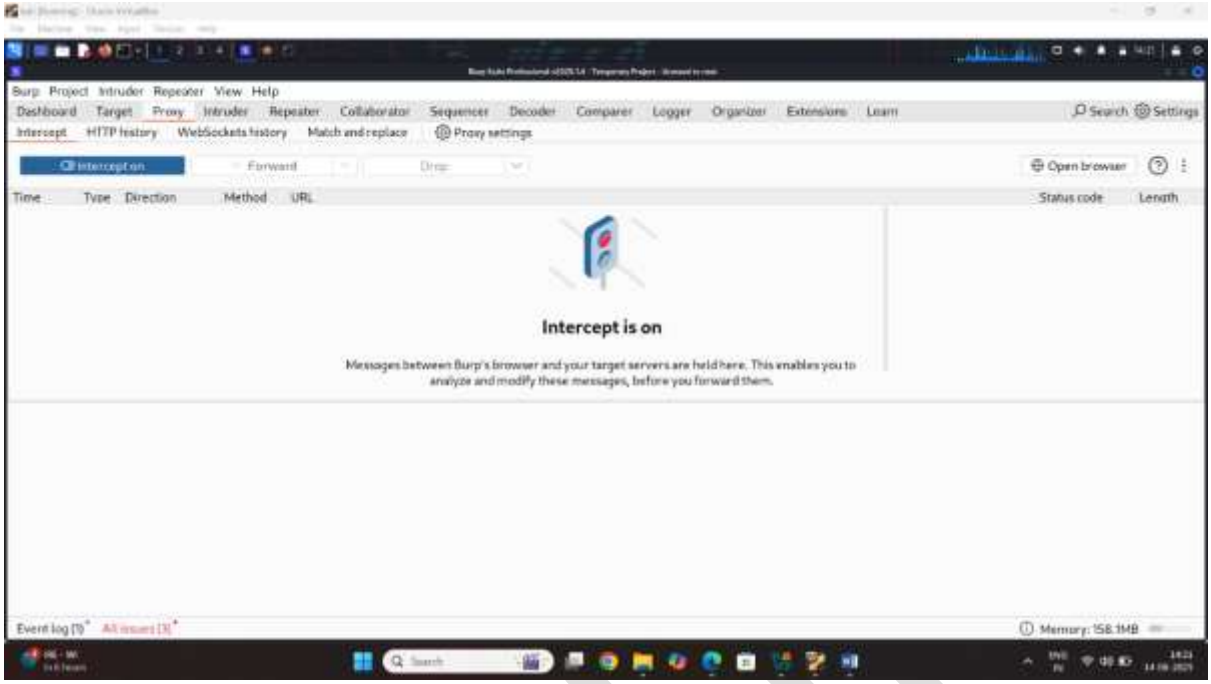


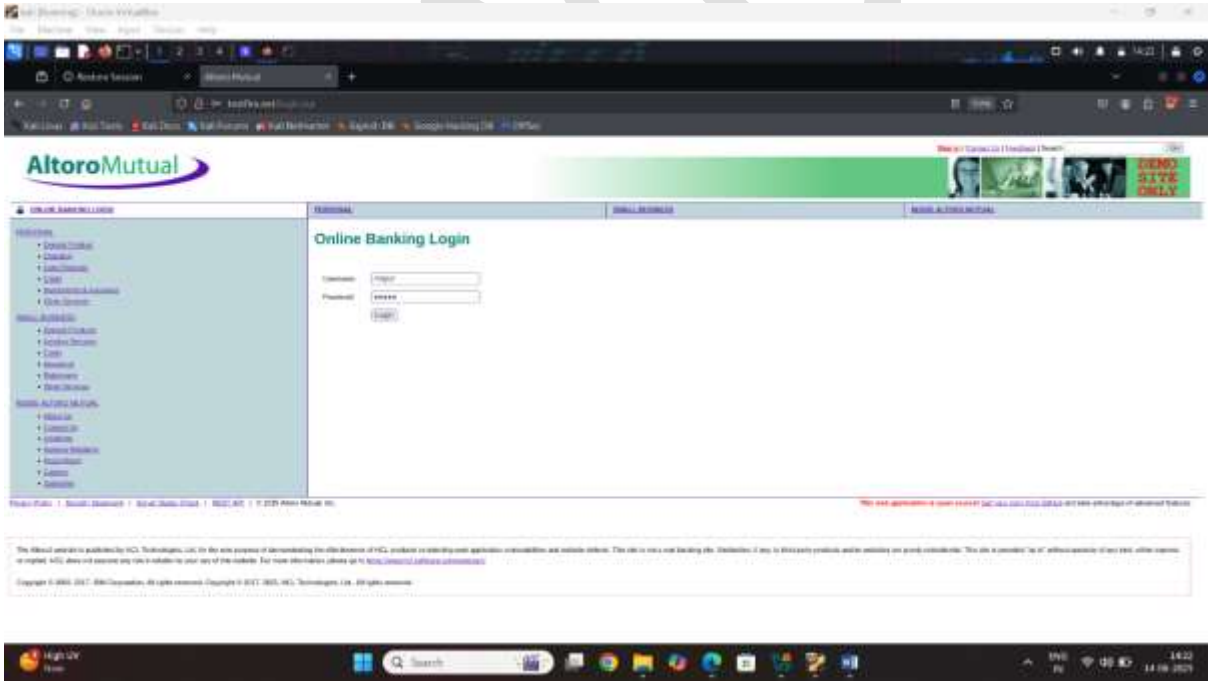
Step2: select the target web site and set up burp suite proxy



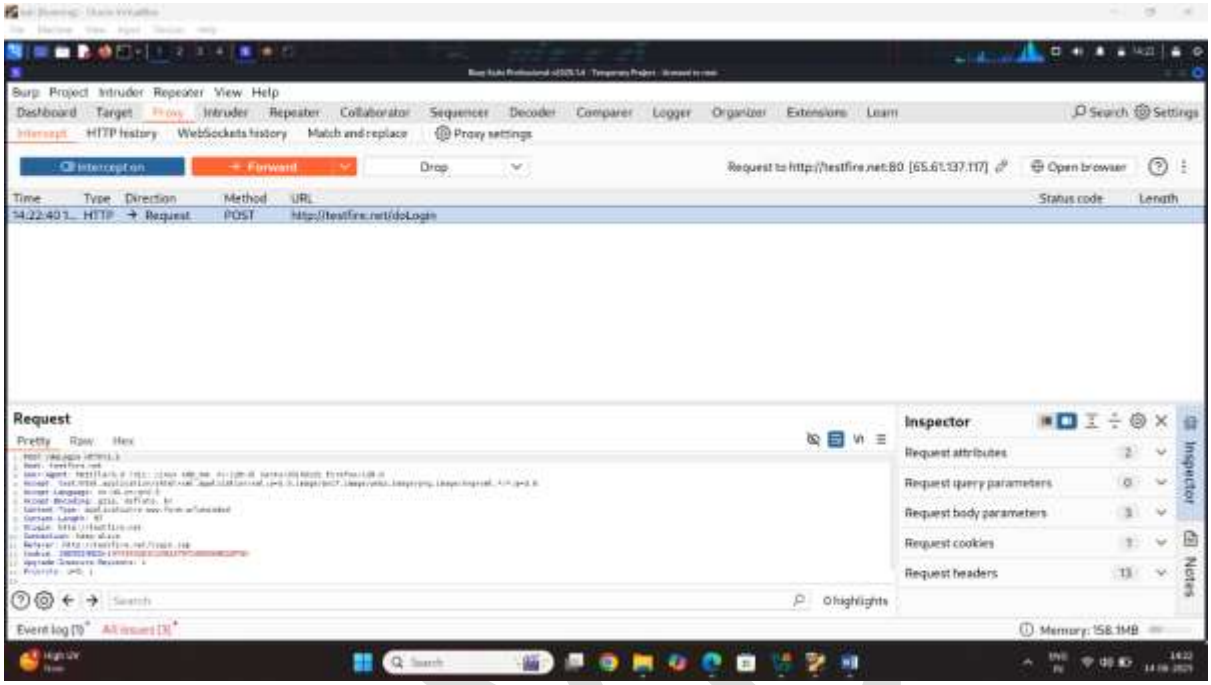
Step3: on the burp suite intercept button

The main purpose of inercept is captering the request of web site


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Search Settings
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Open browser
Time Type Direction Method URL
Status code Length
Intercept is on
Messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.
Event log (1)
All issues (3)
Memory: 158.1MB

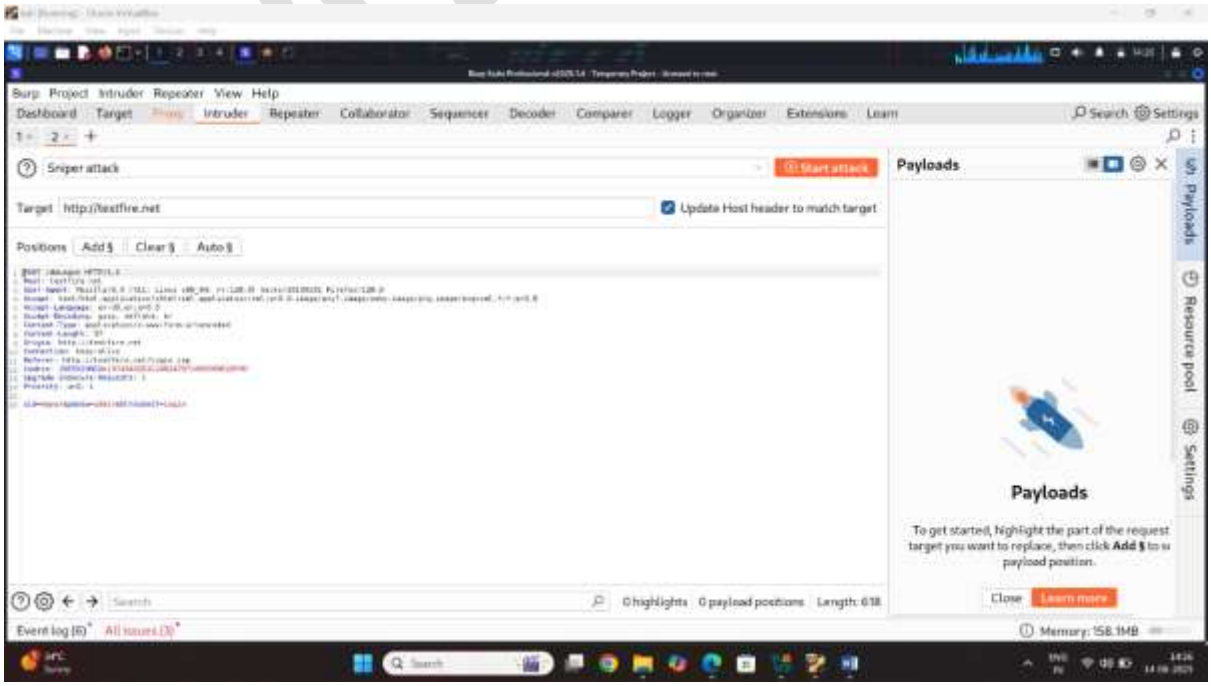

AltoroMutual
DEMO SITE ONLY
Online Banking Login

Step4: intercept the request



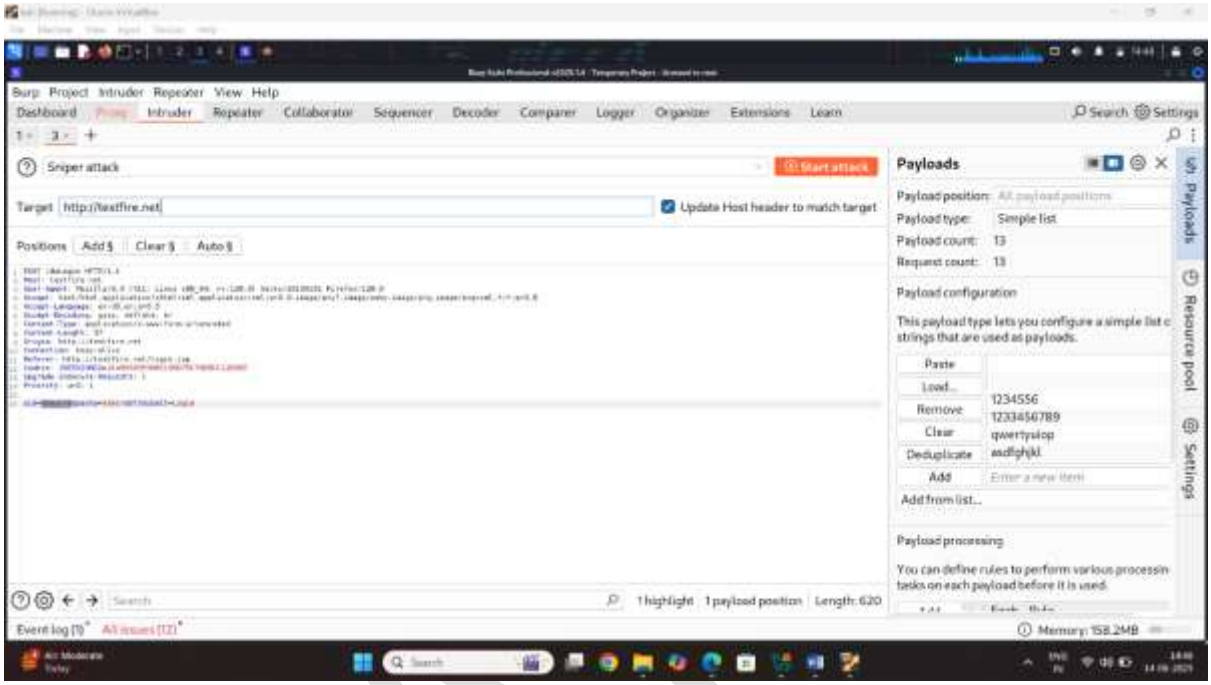
Step5: capter the request to send the intruder

What is intruder in burp suite

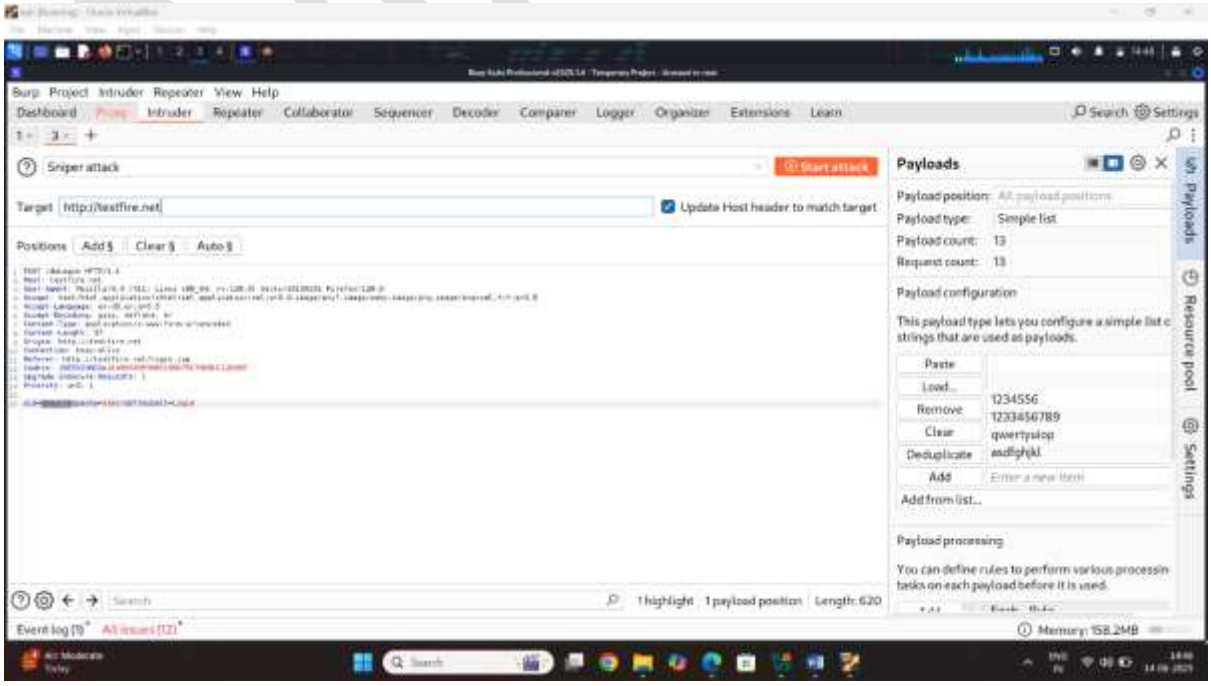


Step6: select the use name add click on the option

Because is test username



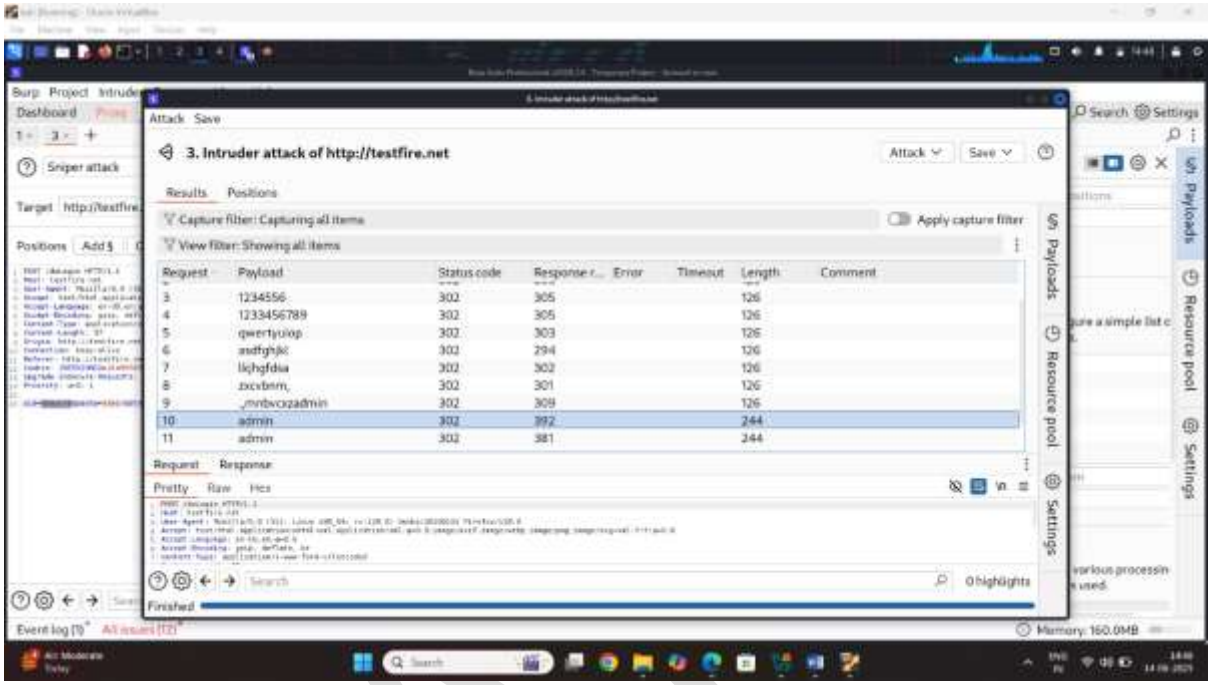
Step7: select the sniper attack click on start the attack



Step8: add wordlist and sql injection fuzzing attack

Click on start attack

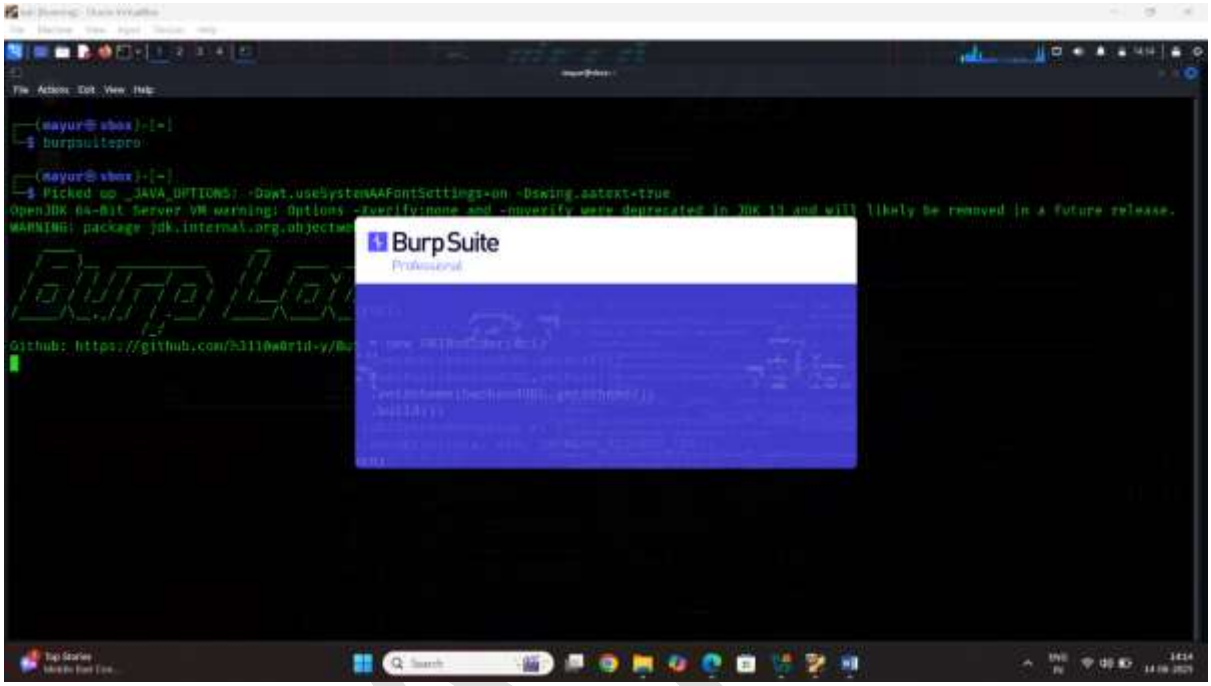
Result:

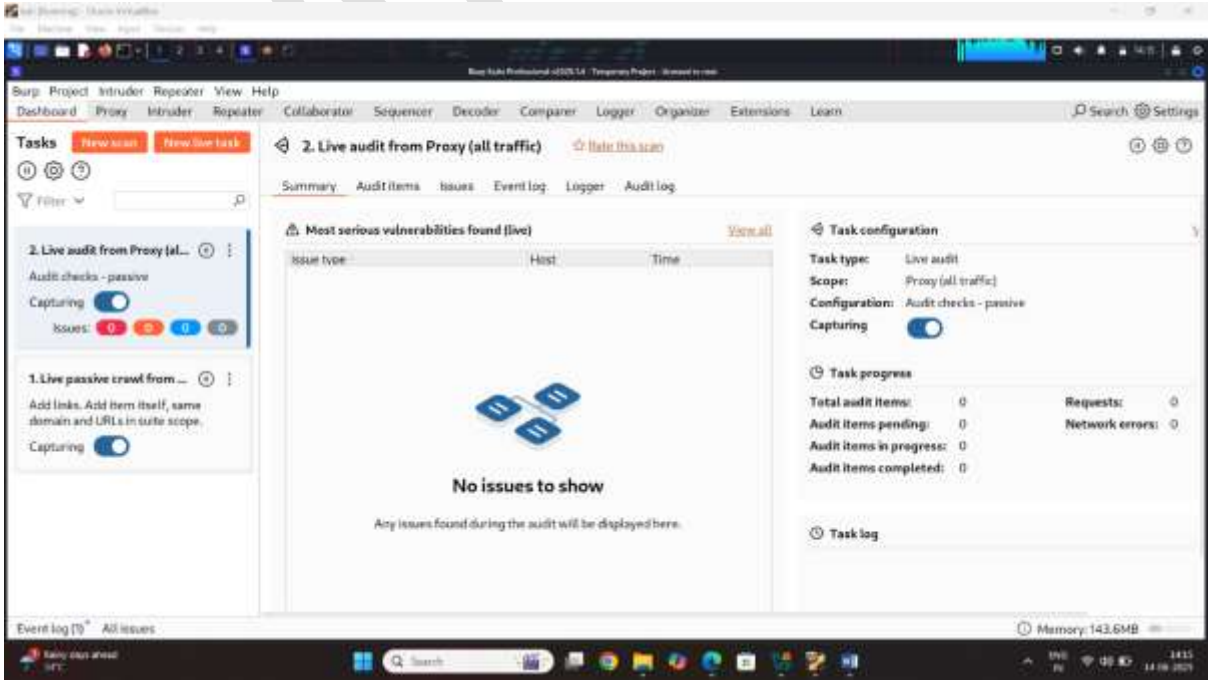


Task 10 how to test web application using burp suite using bomber cluster bomb attack

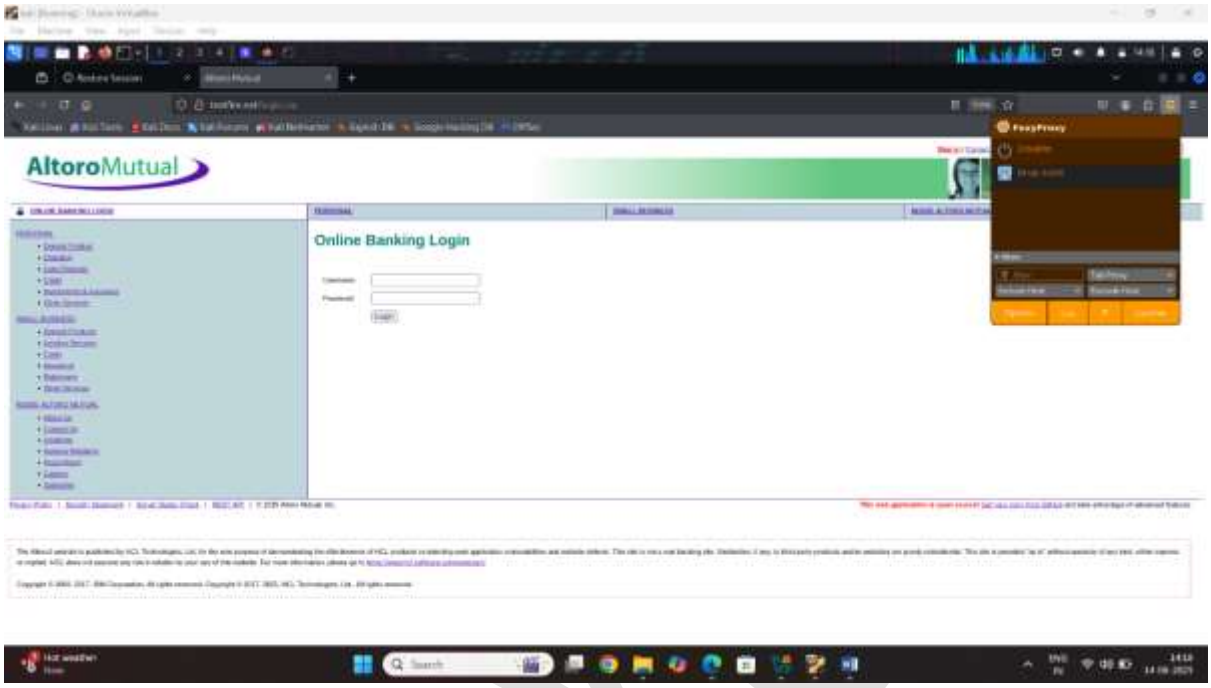
How to test username and password sql injection vunlabritiy

Step1: start the kali Linux machine open the terminal search then burpsuitepro



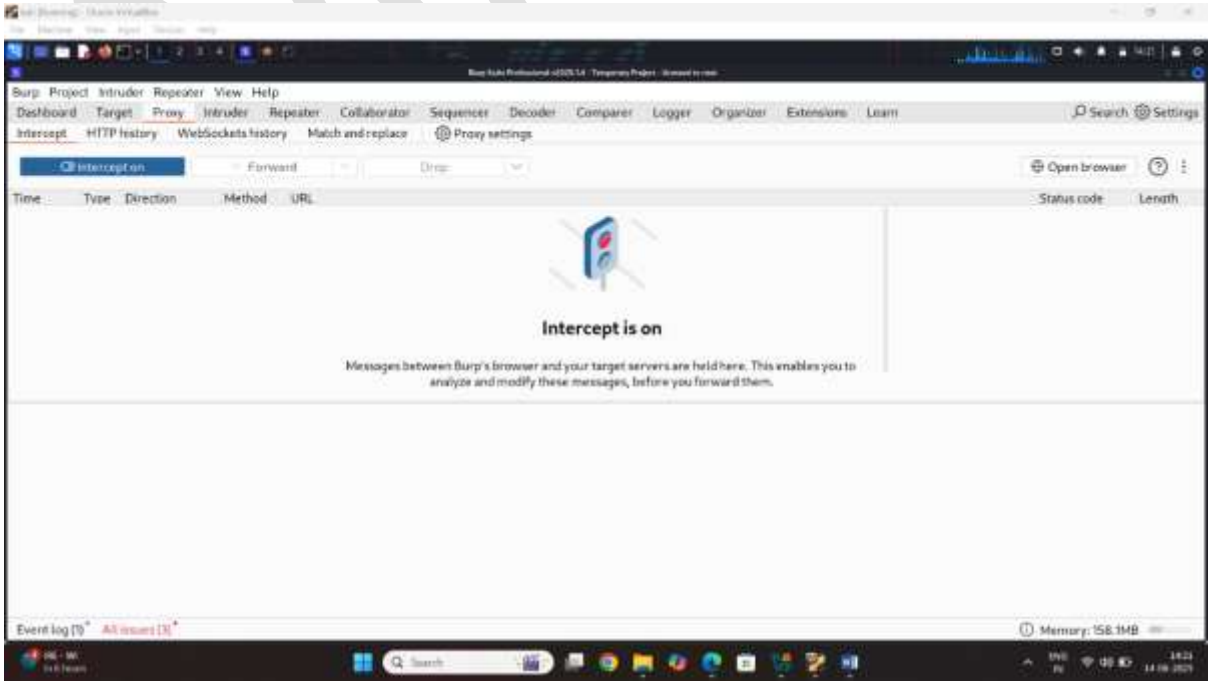


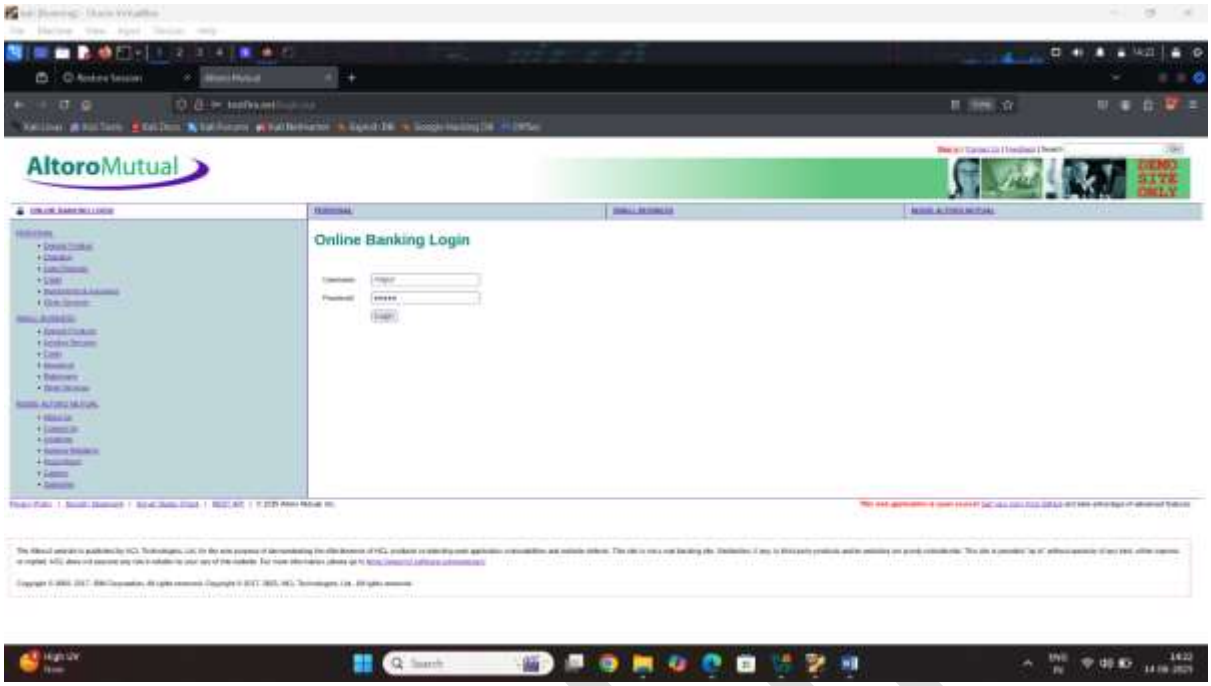
Step2: select the target web site and set up burp suite proxy



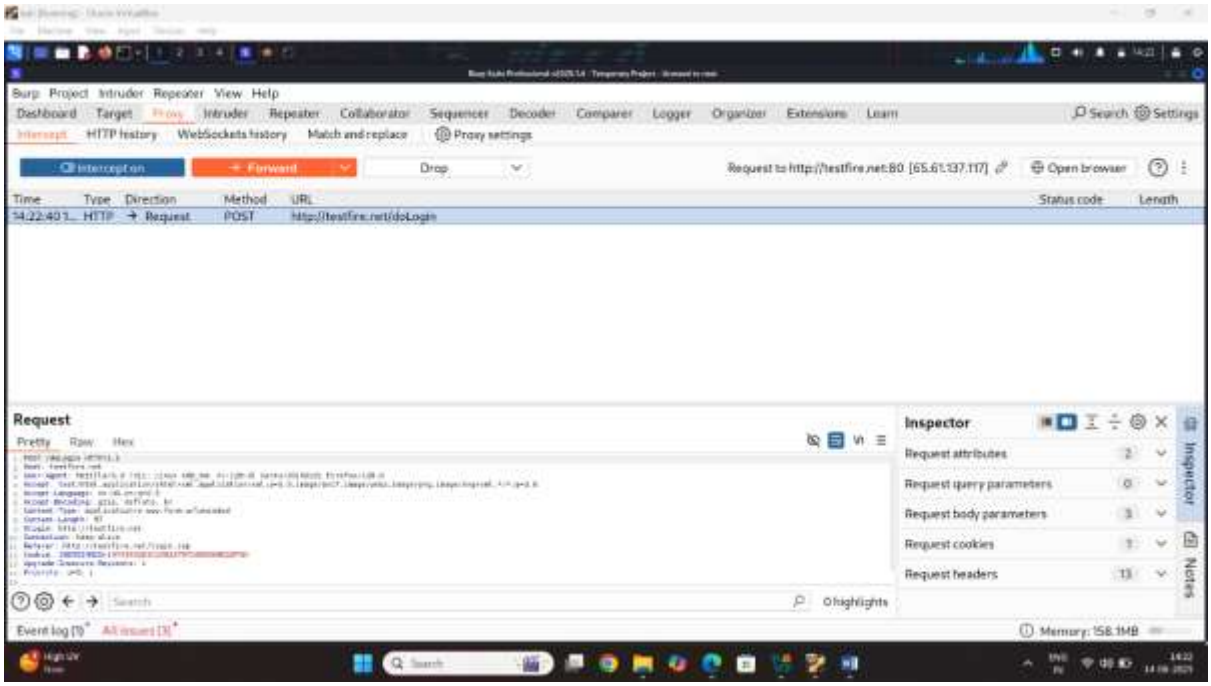
Step3: on the burp suite intercept button

The main purpose of inercept is captering the request of web site

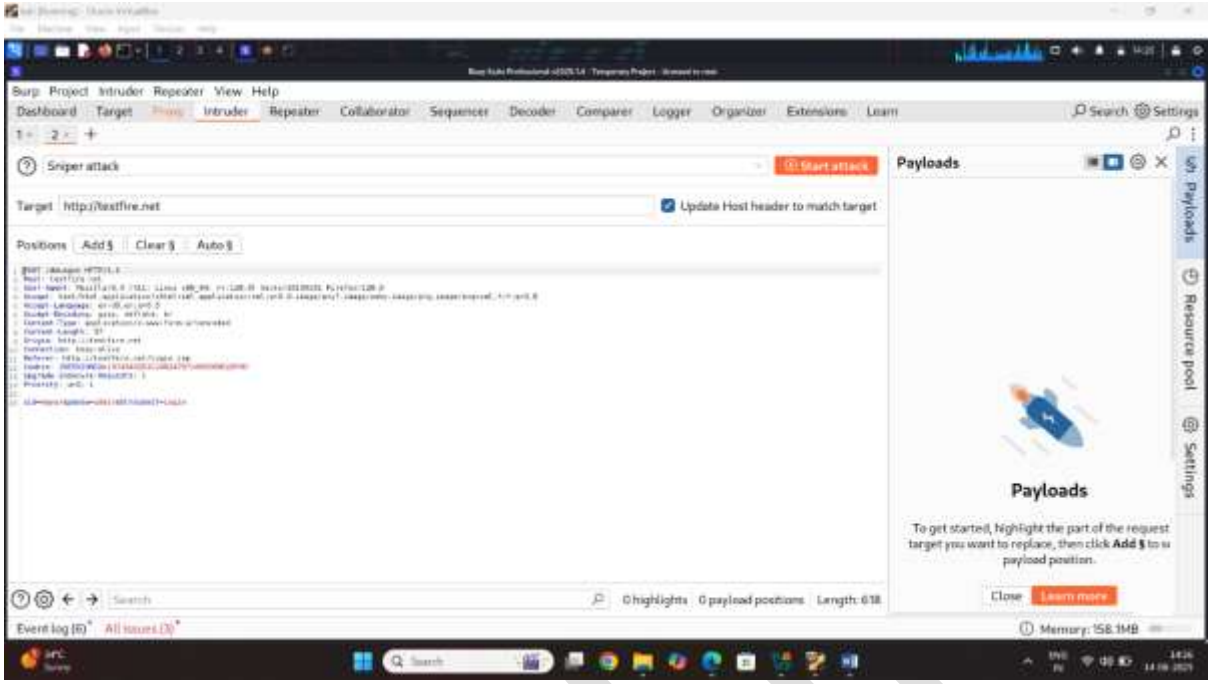




Step4: intercept the request

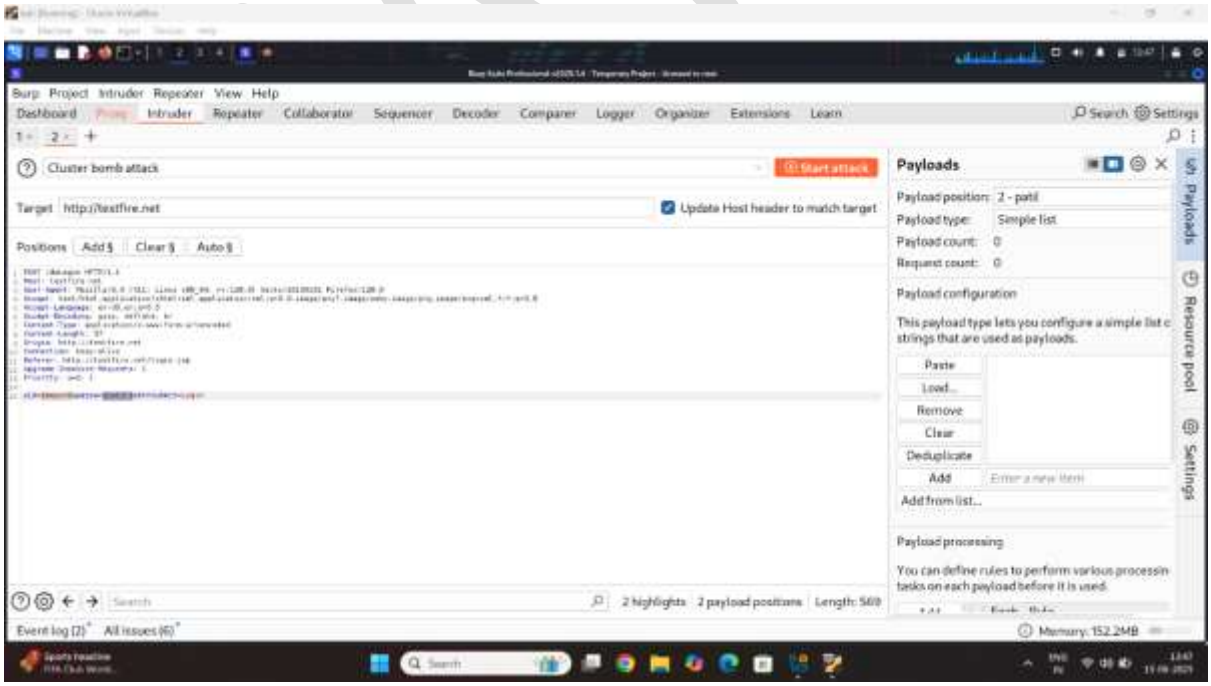


Step5: intercept the request to send the intruder



Step6: select the use name add click on the option

Because is test username and password

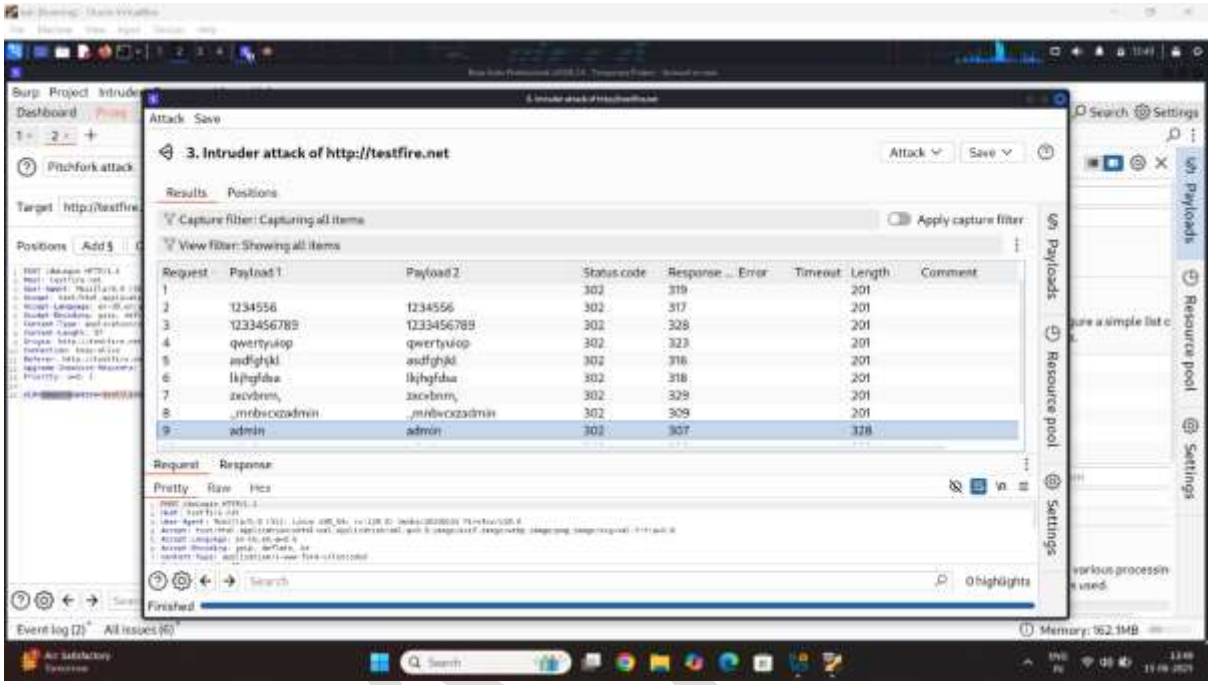


Step7: select the cluster bomb attack click on start the attack

Step8: add wordlist and sql injection fuzzing attack

Click on start attack

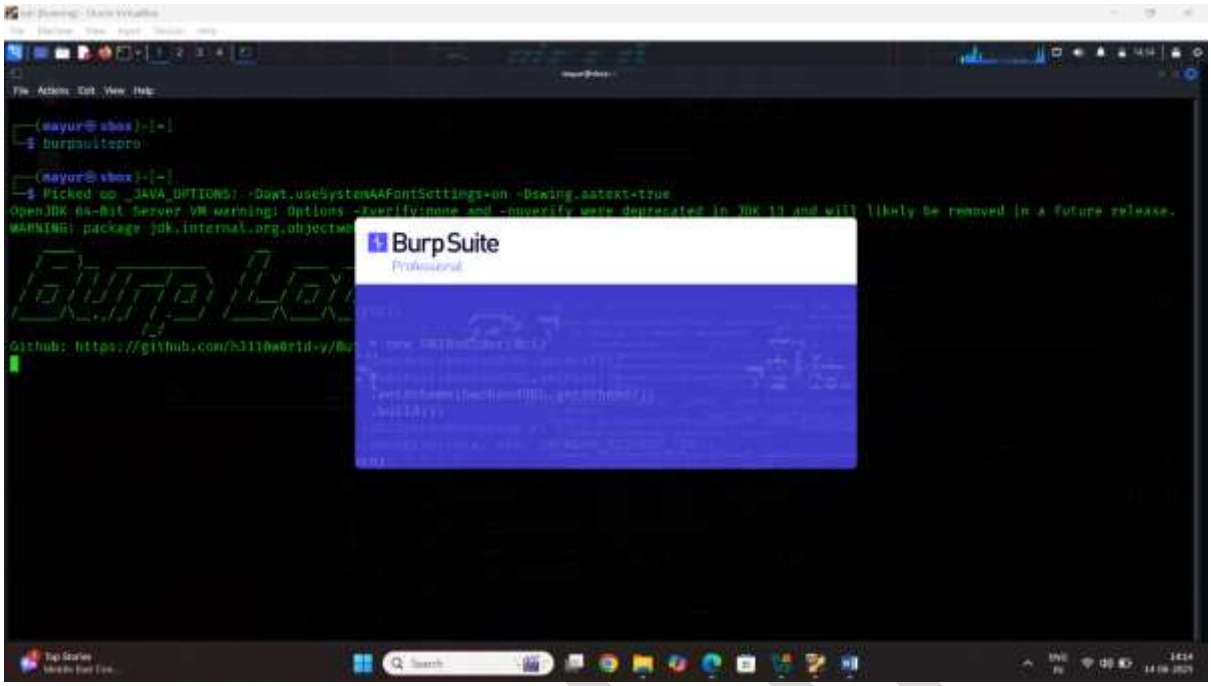
Result:

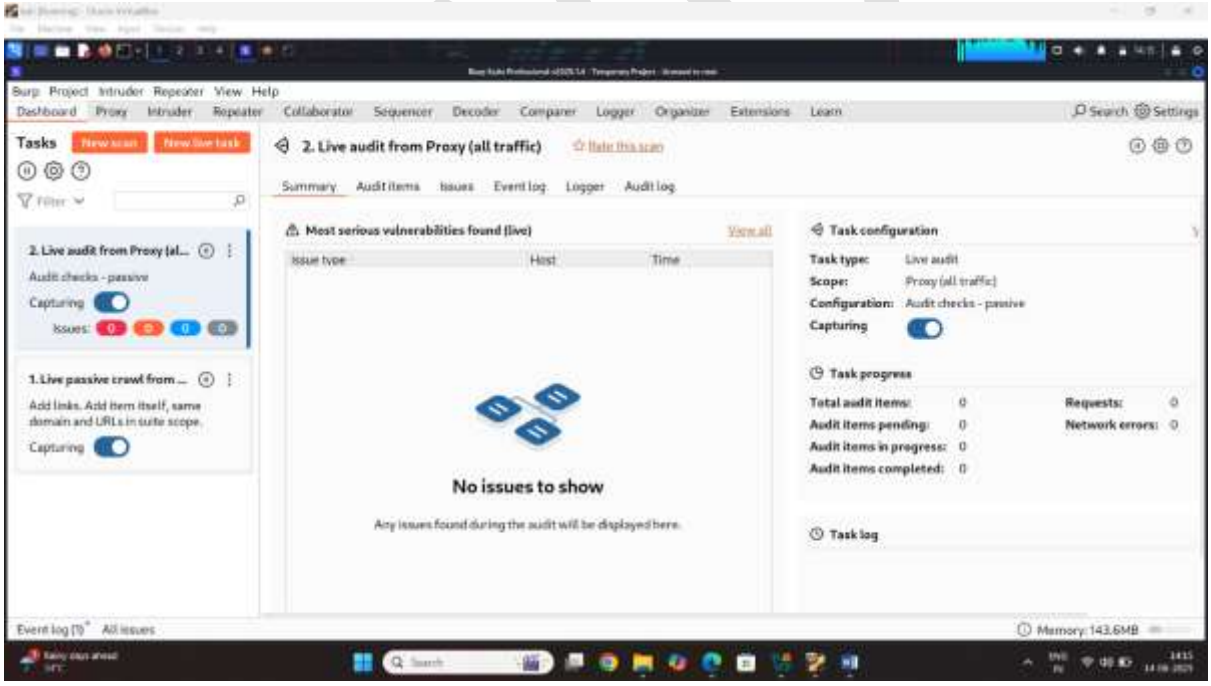


Task 11 how to test web application using burp suite using Battering ram attack

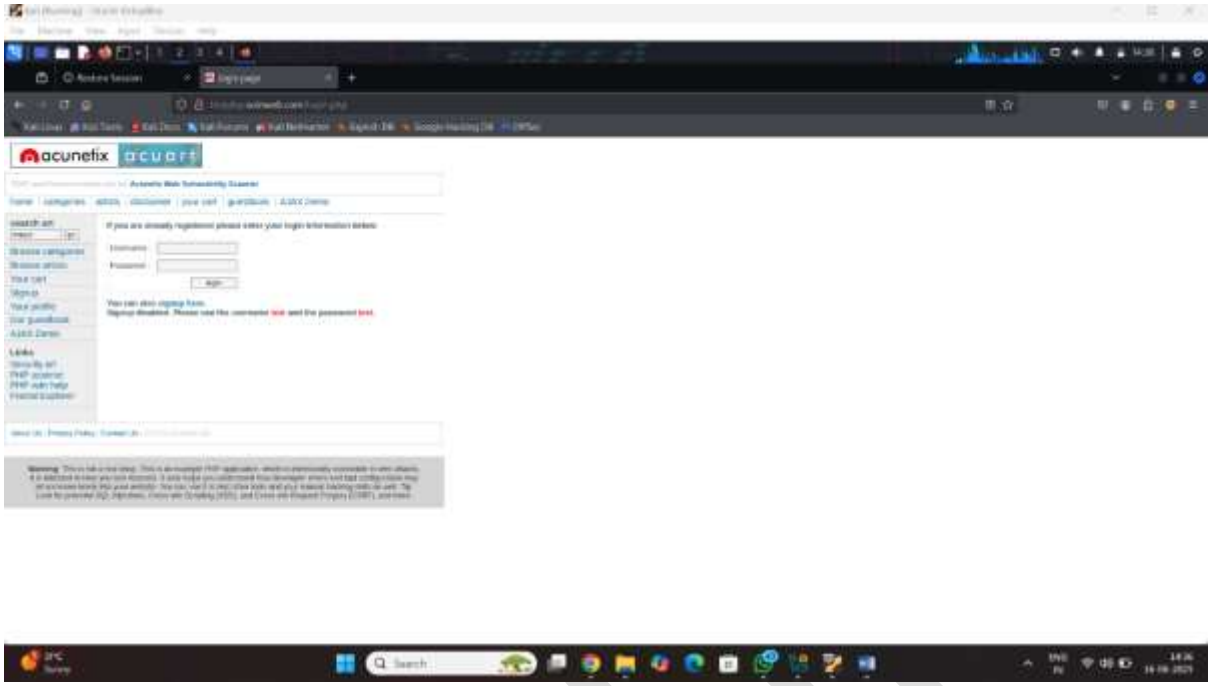
How to test web application search box

Step1: start the kali Linux machine open the terminal search then burpsuitepro



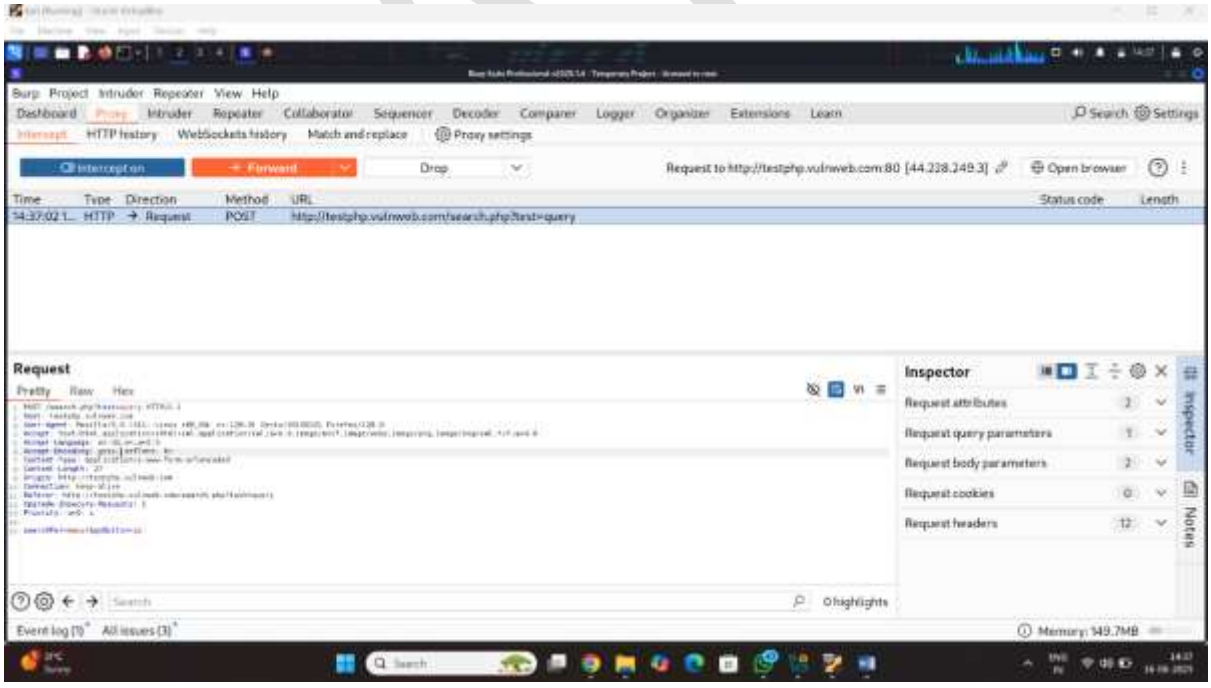


Step2: select the target web site and set up burp suite proxy



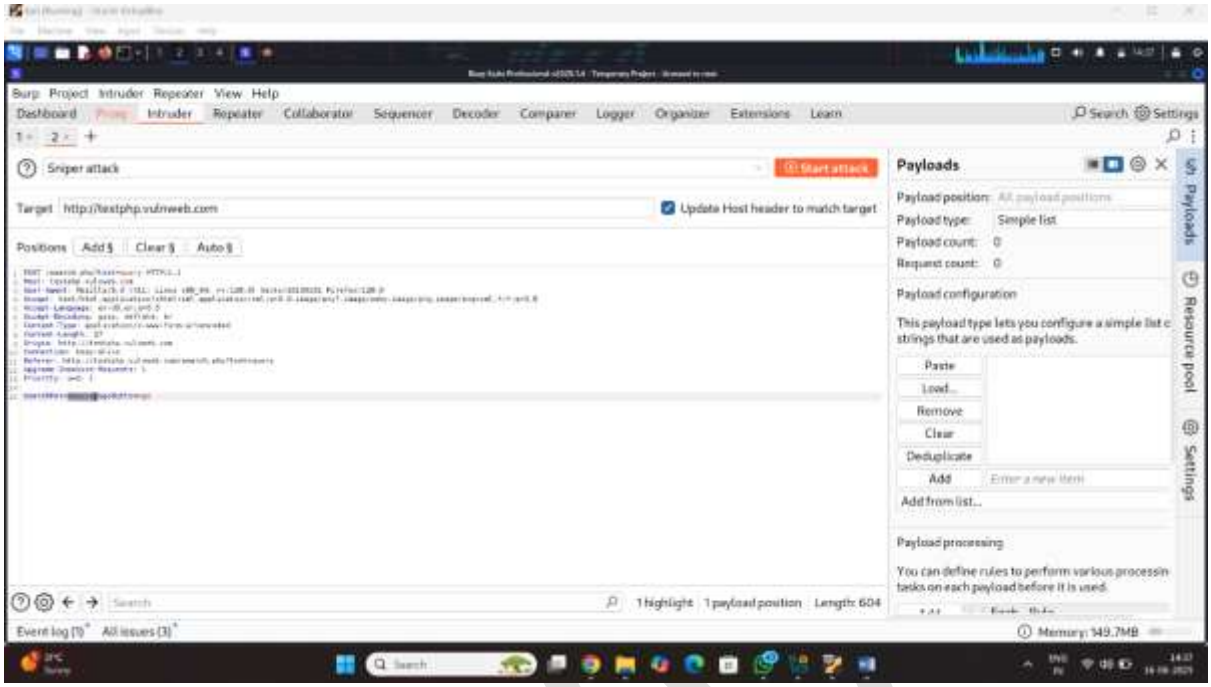
Step3: on the burp suite intercept button

The main purpose of inercept is captering the request of web site

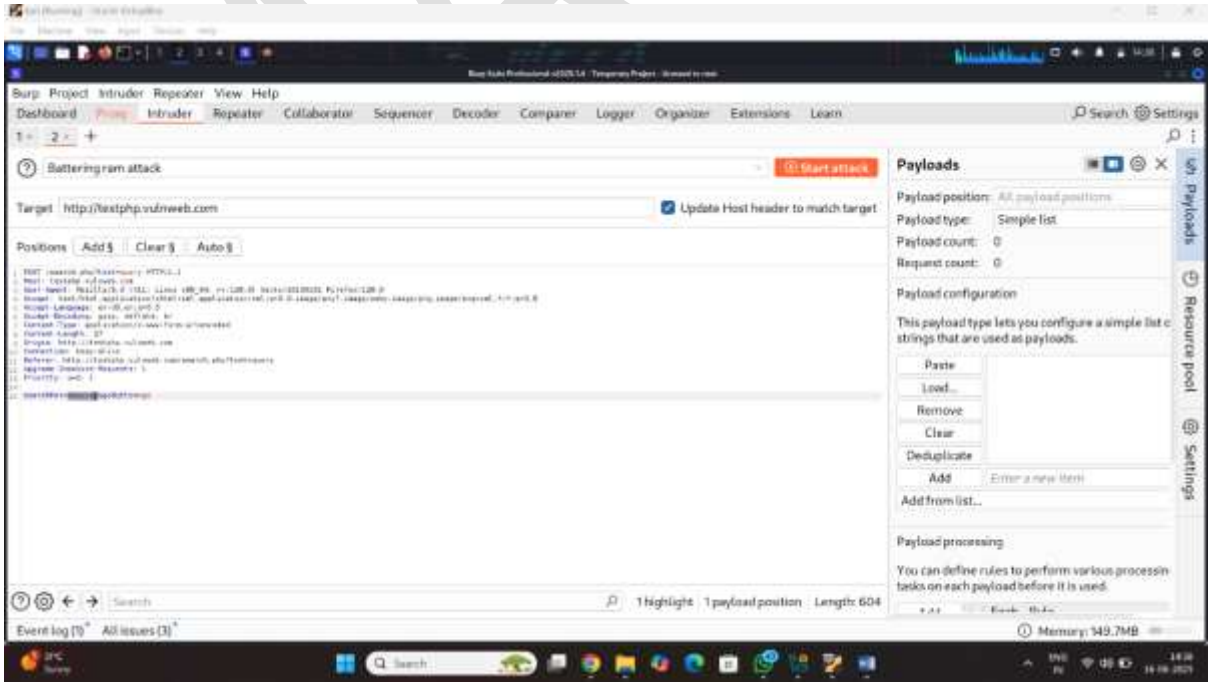


Step4: intercept the request

Step5: intercept the request to send the intruder

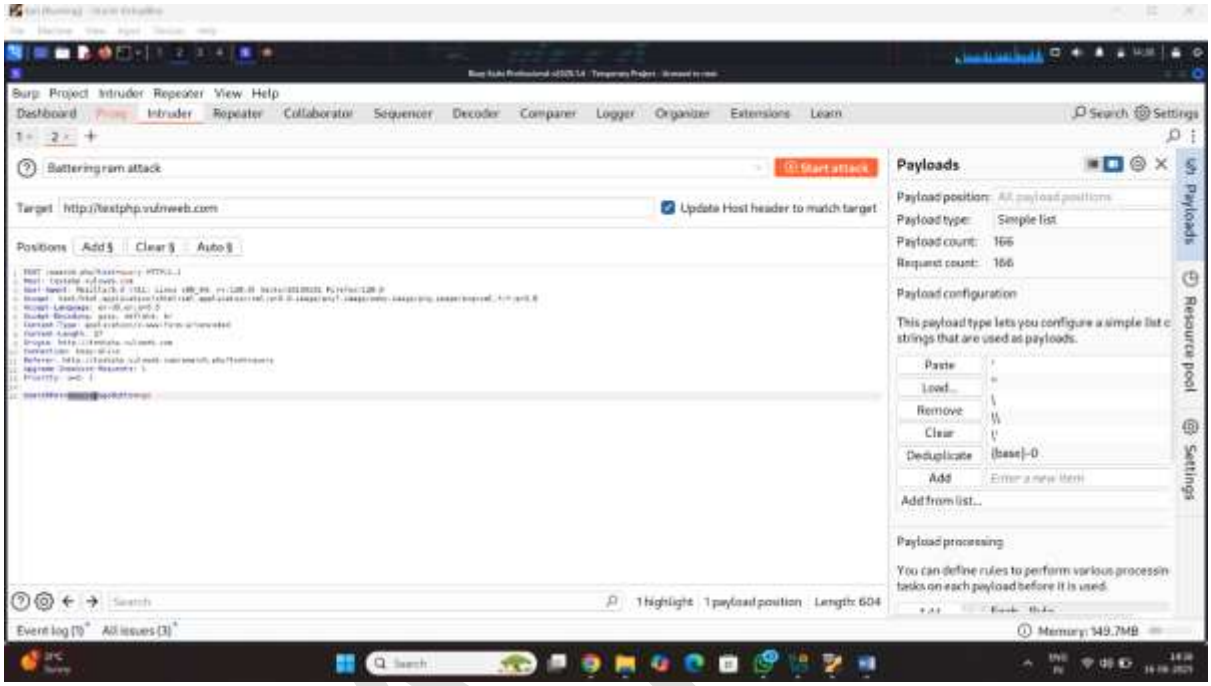


Step6: select the search box txt add click on the option Because is test the box is vulnerability sql injection

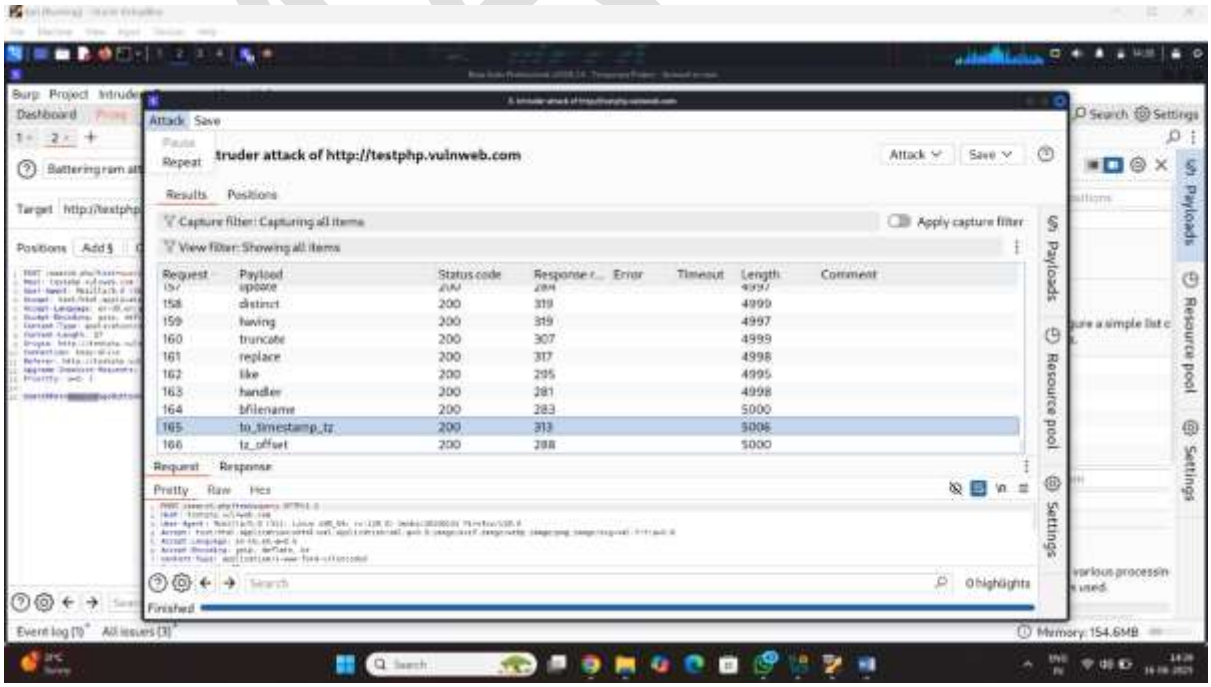


Step7: select the attack type I am select the battering ram

Step8: load sql injection scripts click on start attack



Result:



How to defend against injection attack

🔒 1. Input Validation

- **Whitelist validation**: Only allow expected input (e.g., only numbers for age fields).
- **Reject invalid characters**: Disallow potentially dangerous inputs (e.g., quotes, semicolons

2. Use Parameterized Queries (Prepared Statements)

- Avoid building SQL or command strings with user input.
- Use prepared statements with bound parameters.

🛡 3. Use ORM (Object Relational Mapping)

- Tools like **SQLAlchemy**, **Hibernate**, or **Entity Framework** can help abstract and safely handle database interactions.

4. Sanitize Input

- Escape special characters when inserting user input into queries or scripts.

- Use built-in sanitization functions relevant to the technology/language.

🔍 5. Security Headers & Content Escaping

- Use appropriate headers (e.g., `Content-Security-Policy`) to help mitigate XSS, which is related to injection.
- Properly escape output when rendering to web pages.

8. Regular Security Testing

- Use tools like:
  - **Burp Suite**, **OWASP ZAP** (for manual & automated testing)
  - **Static code analyzers**
  - **Penetration testing**

How to Defend against web application attacks

🔐 1. Input Validation & Output Encoding

- **Validate all user input** (whitelist preferred).
- **Sanitize and encode output** to prevent:
  - **XSS (Cross-Site Scripting)**
  - **SQL Injection**
  - **Command Injection**

□ 2. Use Web Application Firewalls (WAF)

- Deploy a WAF (like **AWS WAF**, **Cloudflare**, or **ModSecurity**) to block known attack patterns automatically.

🔑 3. Authentication and Authorization

- Use **strong authentication** (e.g., MFA).
- Apply **role-based access control (RBAC)**.
- Never trust user roles or credentials from the client-side.

🔒 4. Secure Session Management

- Use **secure cookies** (`HttpOnly`, `Secure`, `SameSite` flags).
- Set **session timeouts** and **regenerate session IDs** after login.

□ 5. Security Testing

- Perform **regular vulnerability scans**, **penetration tests**, and **code reviews**.
- Use tools like:
  - **OWASP ZAP**, **Burp Suite**
  - **Static Application Security Testing (SAST)**
  - **Dynamic Application Security Testing (DAST)**

🔧 7. Patch and Update

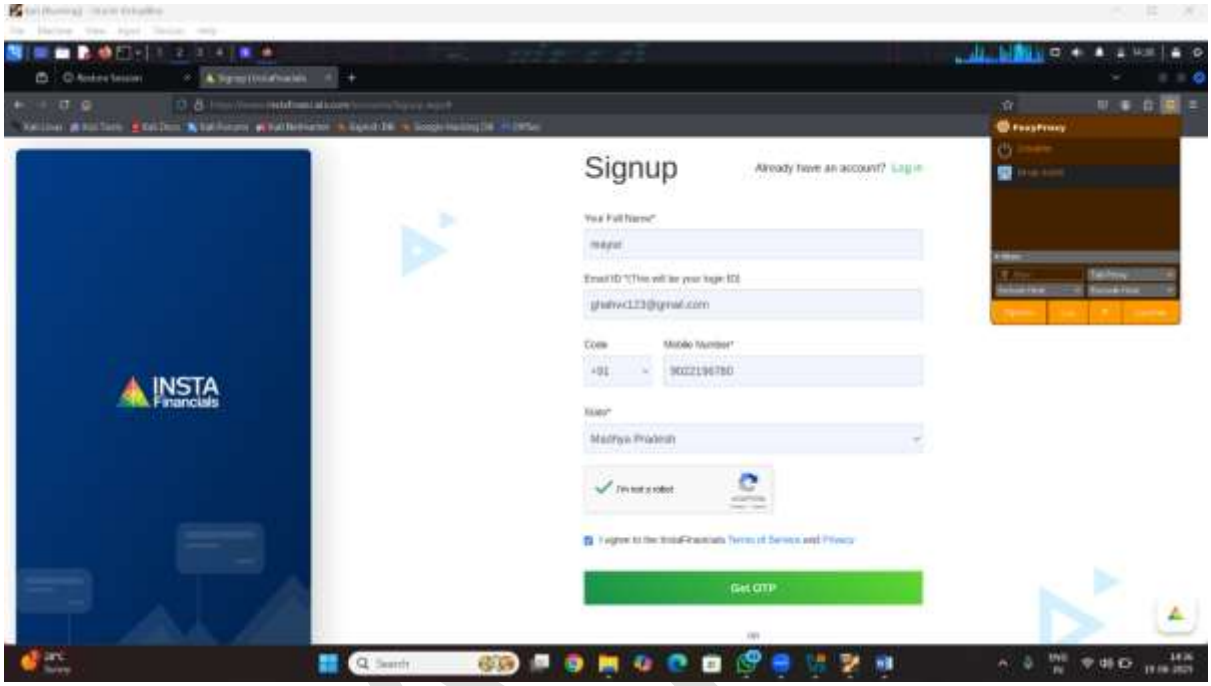
- Keep your web server, frameworks, libraries, and CMS **up to date** to avoid known vulnerabilities.

Extra activity task 11 who to otp bypass using burp suite there is two method of otp bypass

1 st method is server is response manipulate

Step1: select the web site I am select the web site

www.instafincial.com

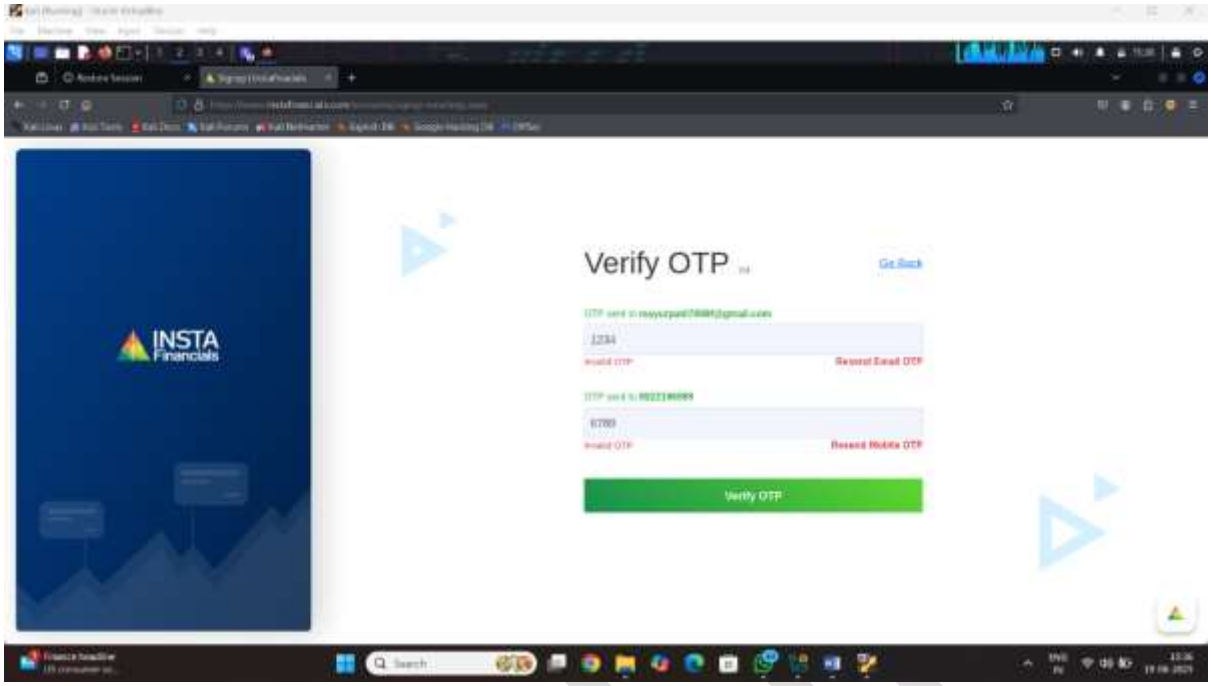


Step2: fill the all information like name ,email and etc

Click on get otp type the any random otp

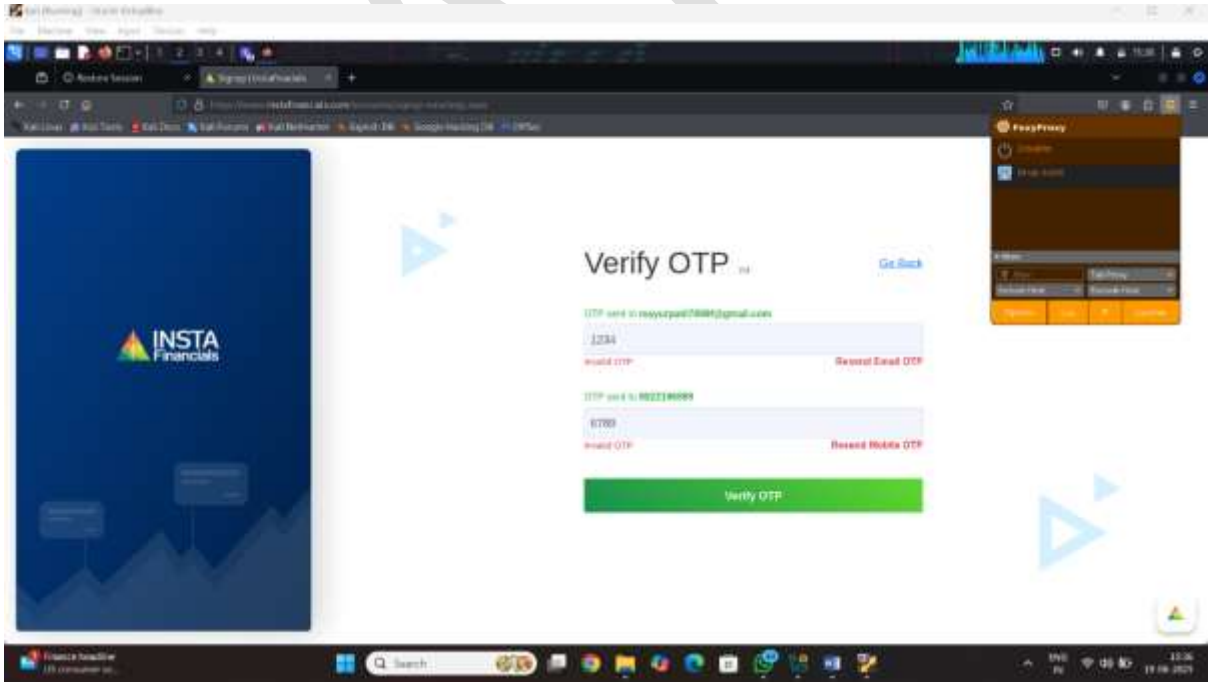
Step3: I am fill the otp email id 1234 and mobile number otp 6789

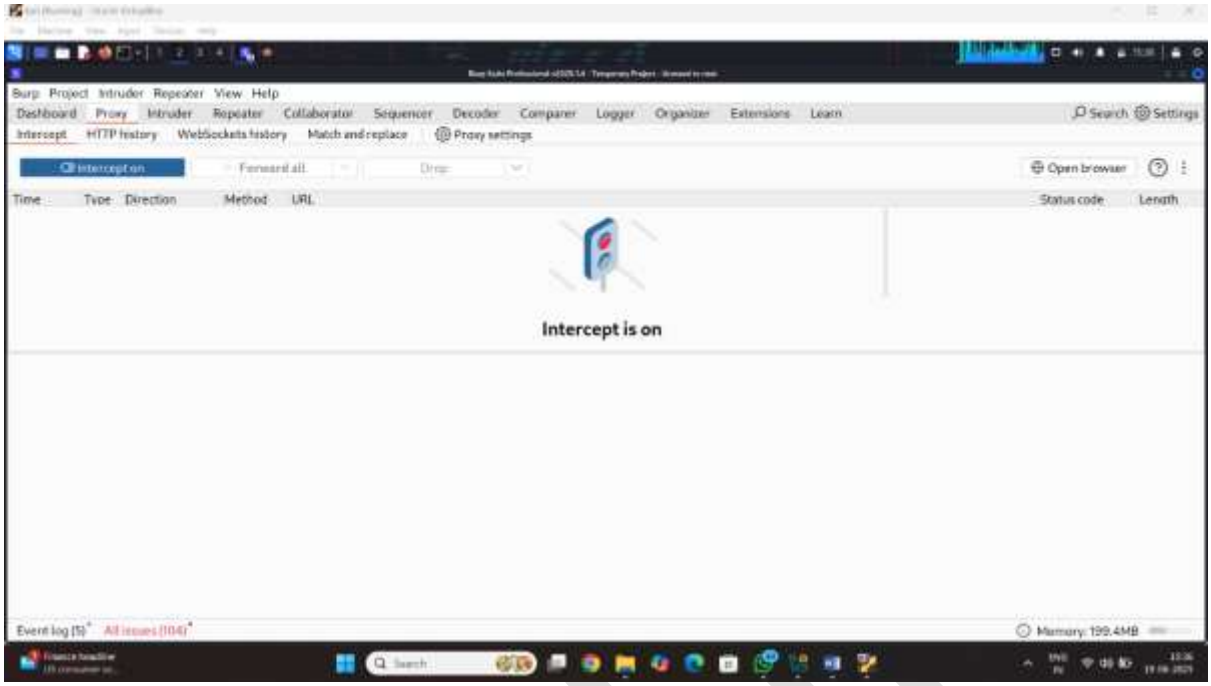
And verify the otp is invalid otp nofication



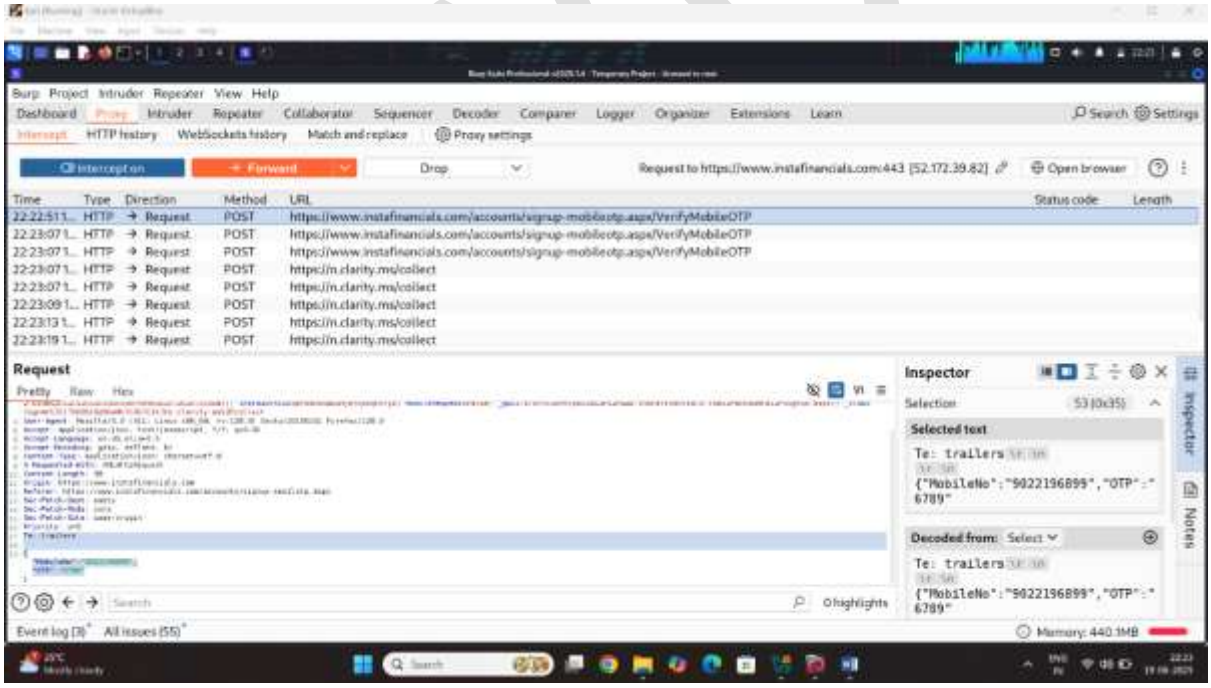
He is otp baypass method include step by step

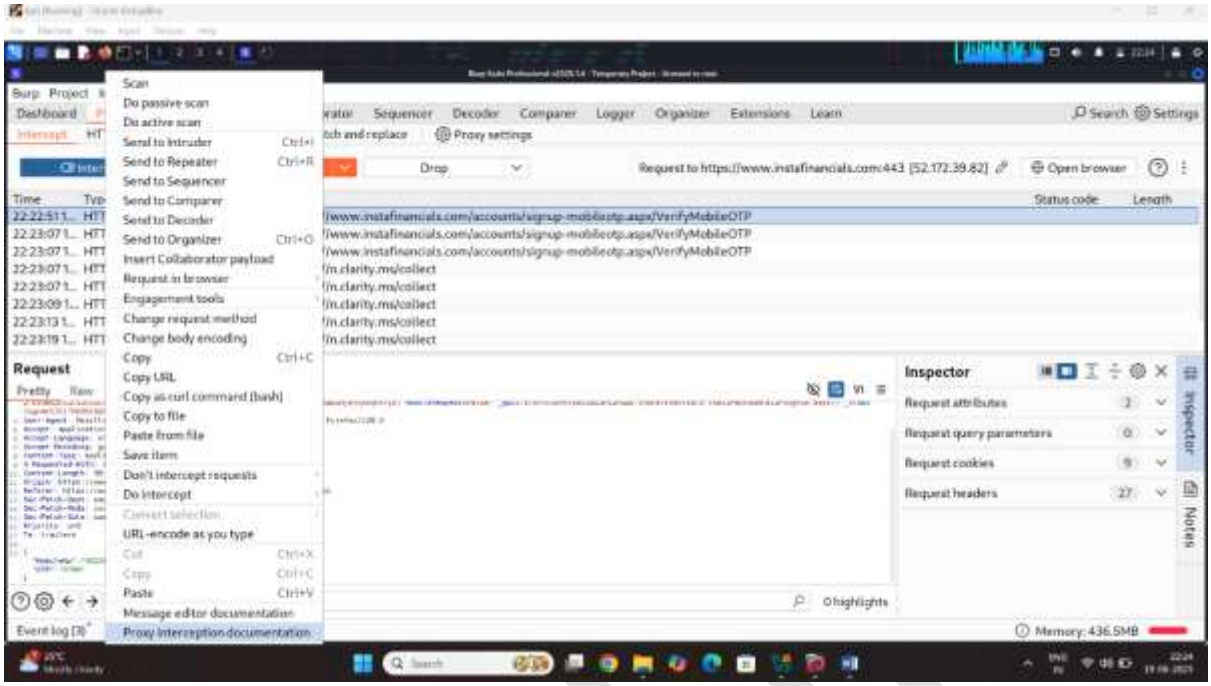
Step4: on the proxy and go to burp suite interception option is on





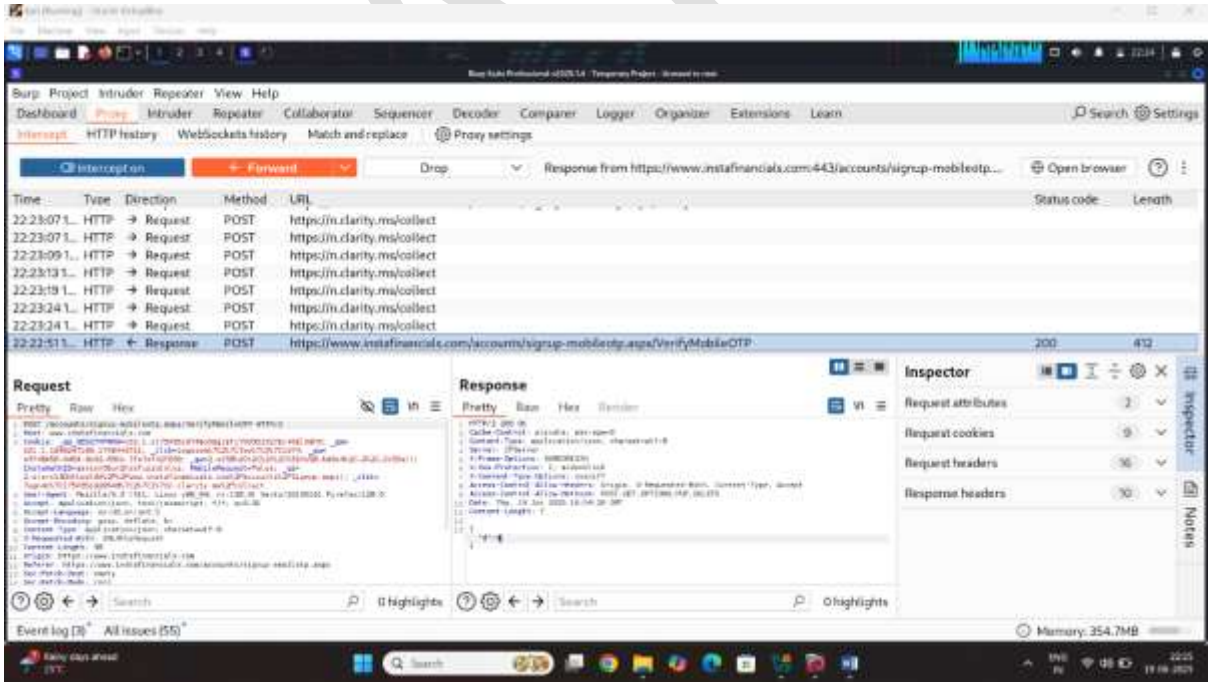
Step6: interception the request on burp suite



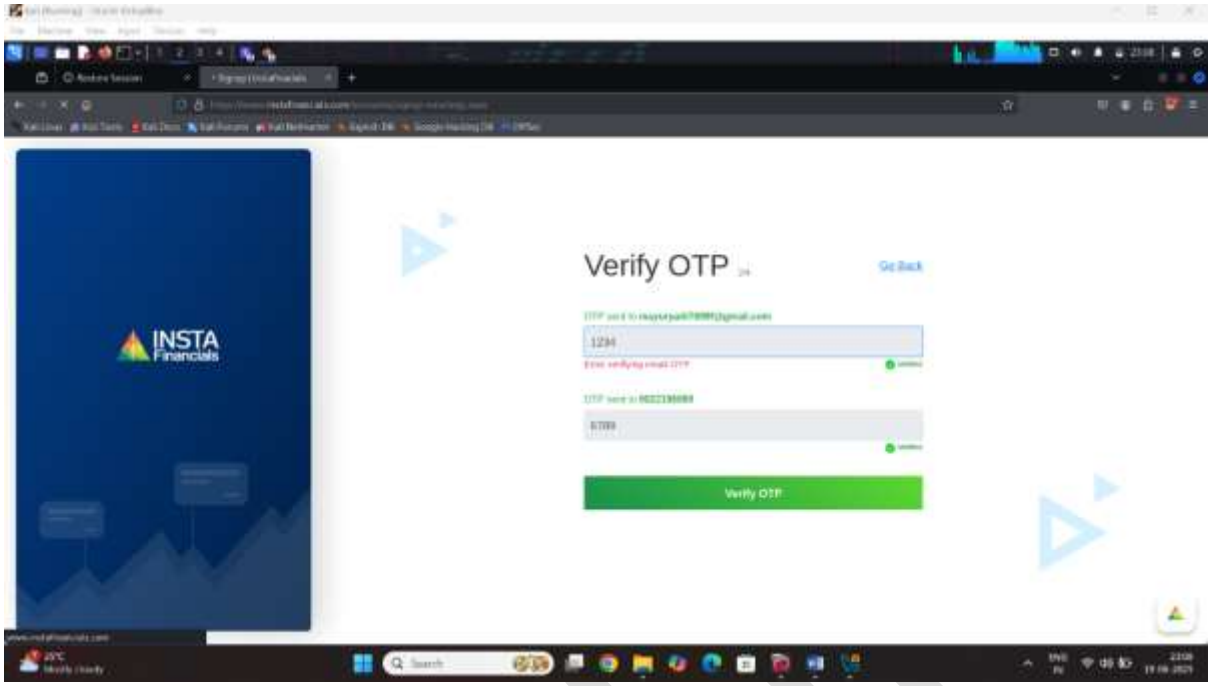


Step5: send to do intercept and click on forward

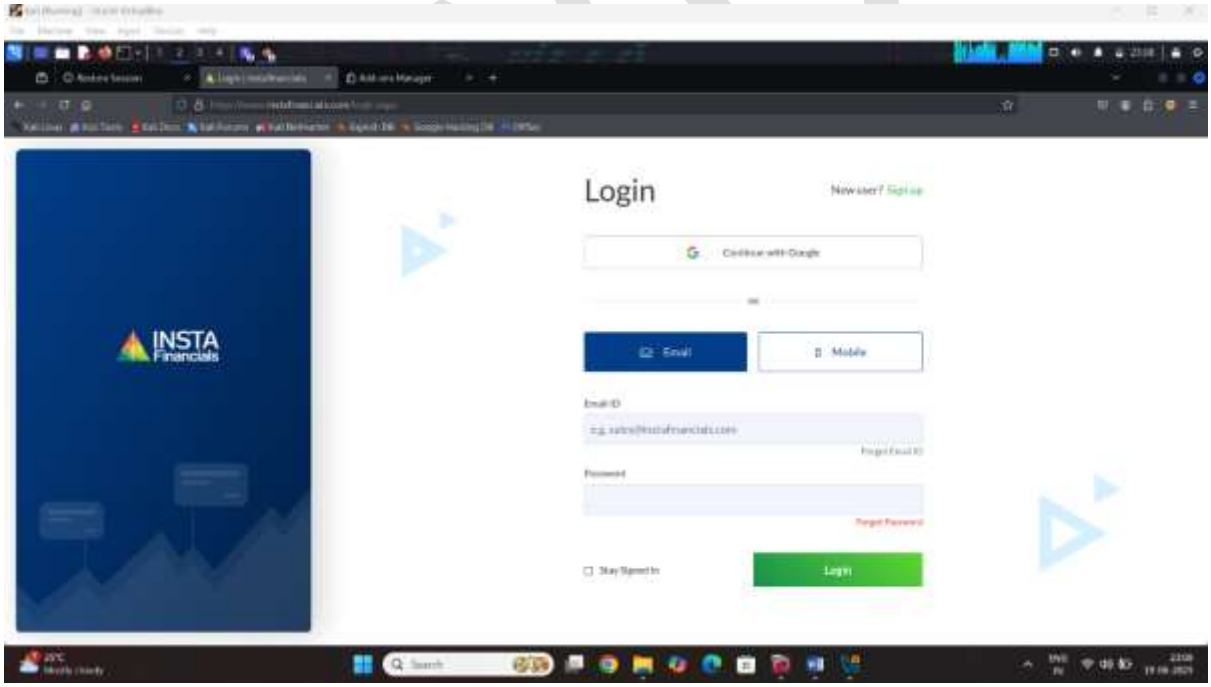
And come to response in sarver to manipulate the otp



Step6: manipulation in server response select D value is server 4 change the manuliye



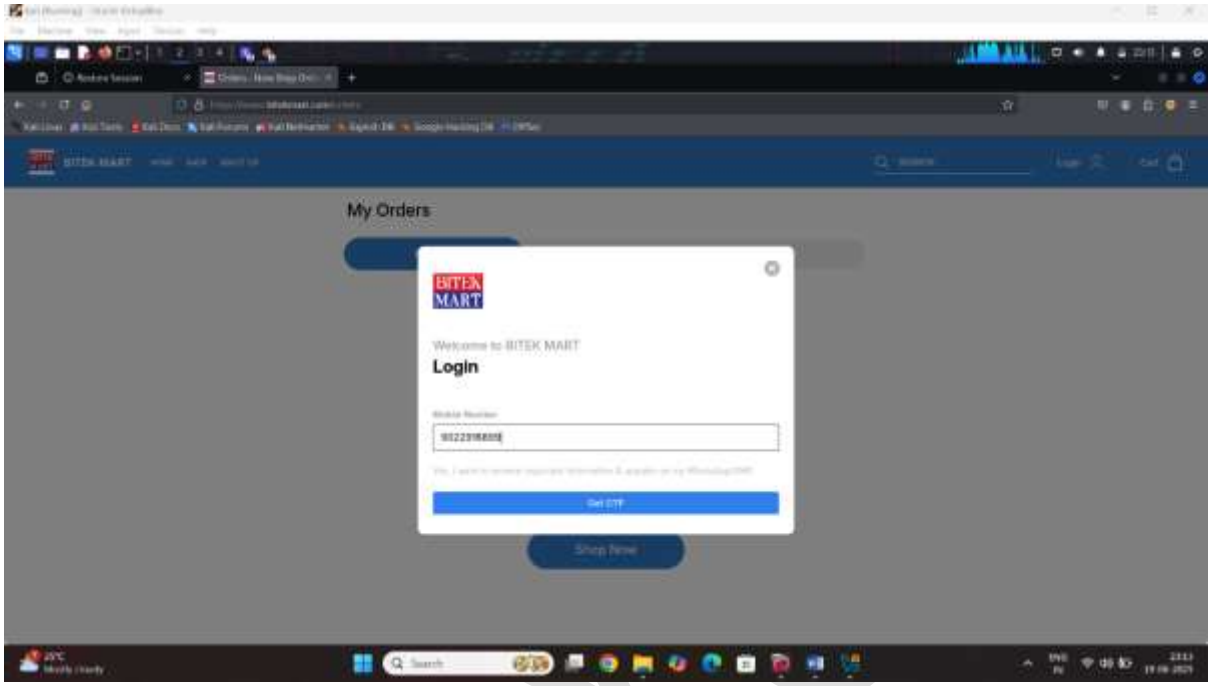
Click on verify the otp and display the login page



2 method is otp bypass in brute force attack

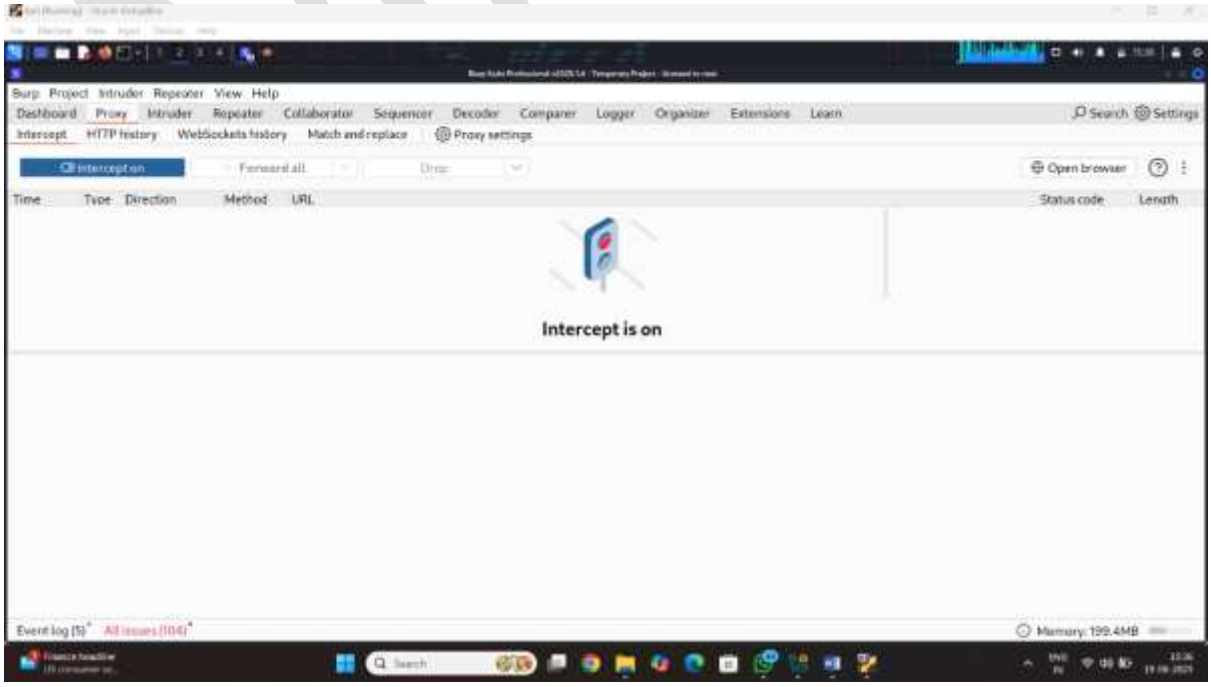
Step1: select the web site I am select the web site

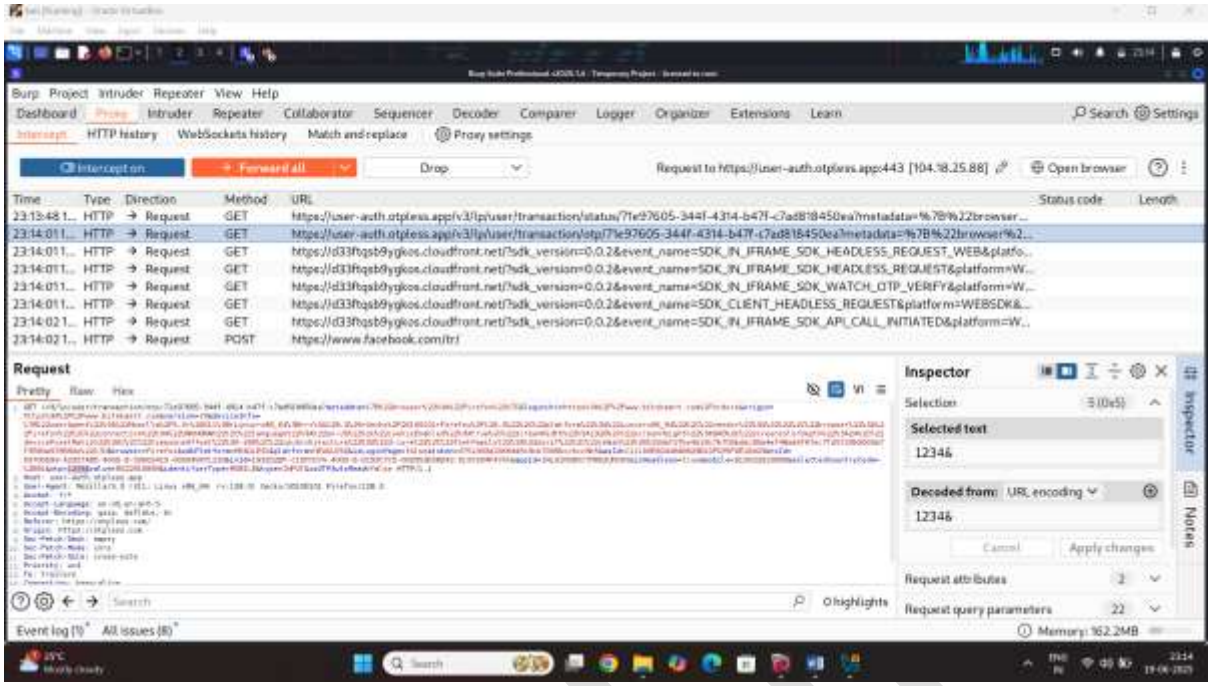
www.bitekmart.com



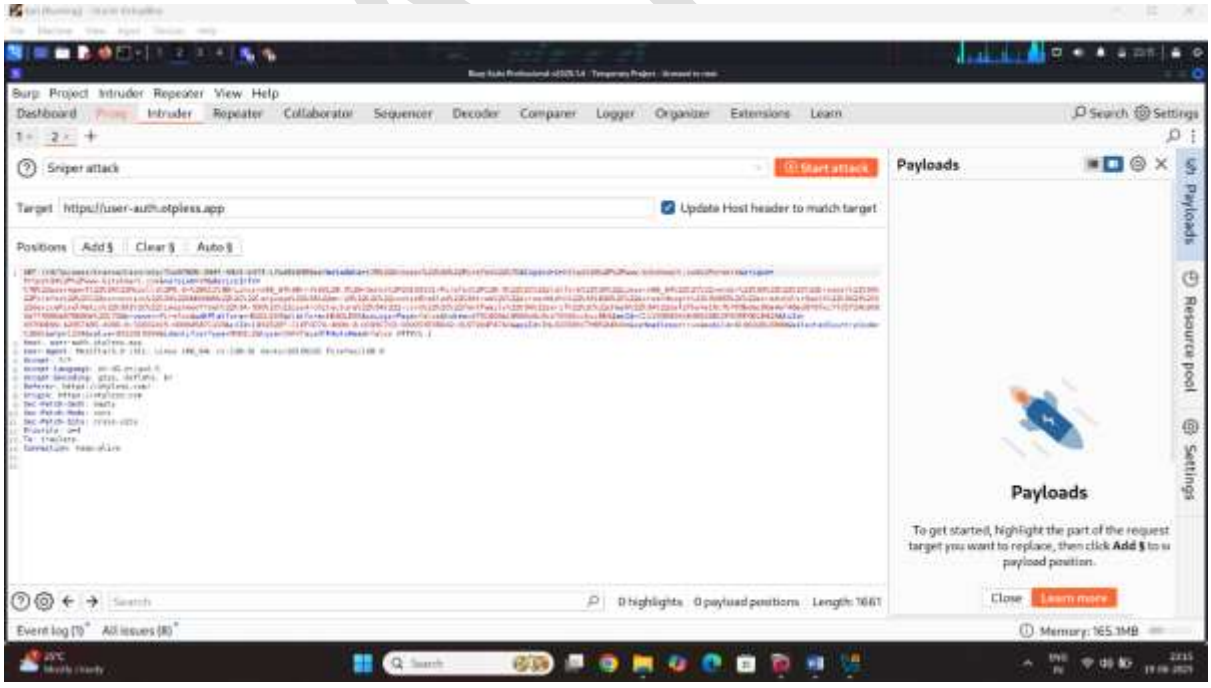
Step2: type the mobile number click on the get otp

Step3: start the burp suite and interception the request of otp

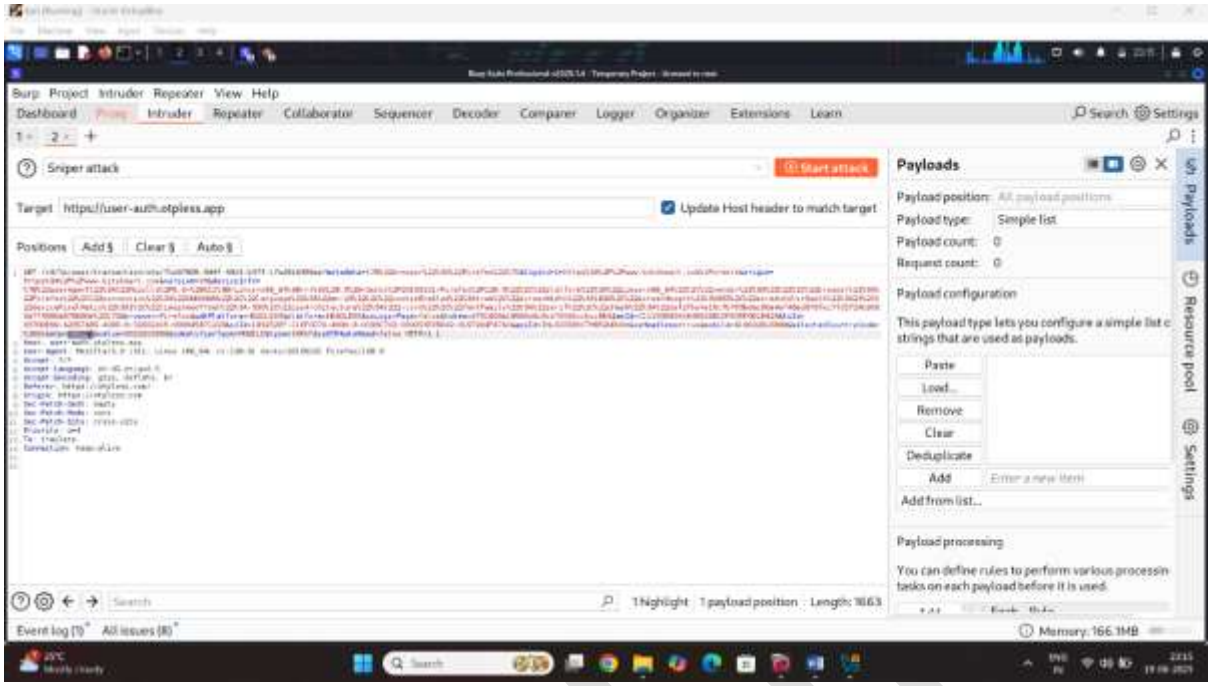




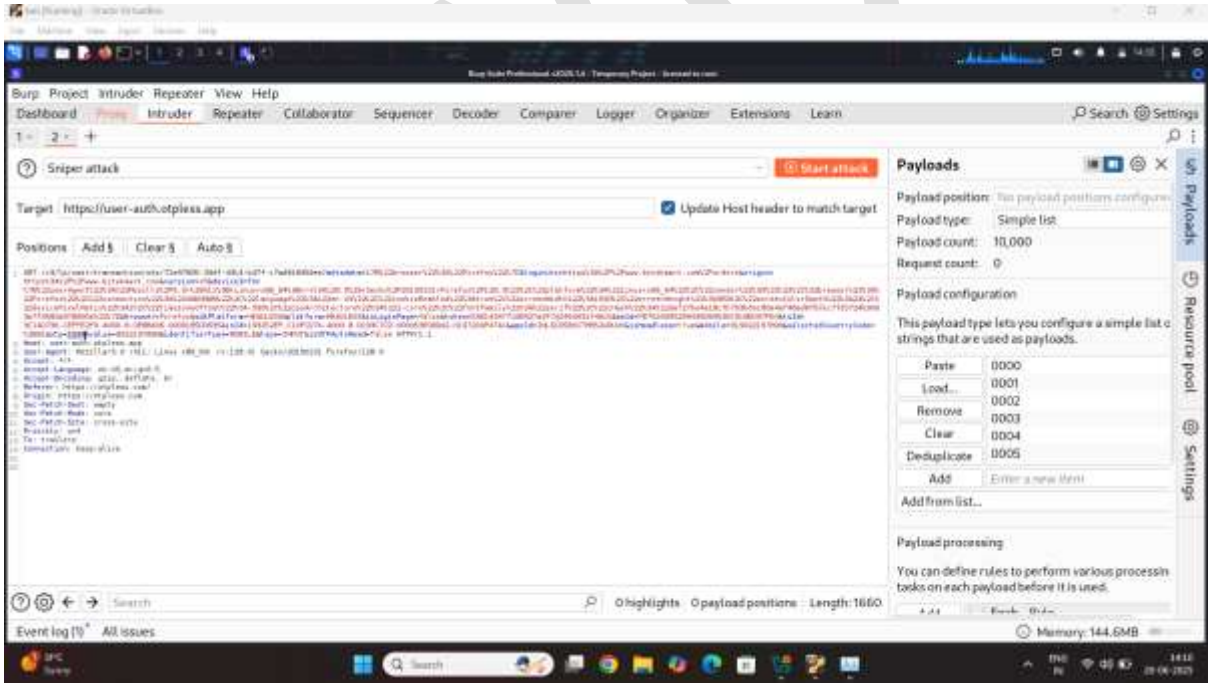
Step4: interception the request and send to intruder module because is try the brute force attack



Step5:select the otp option click on the add otp section

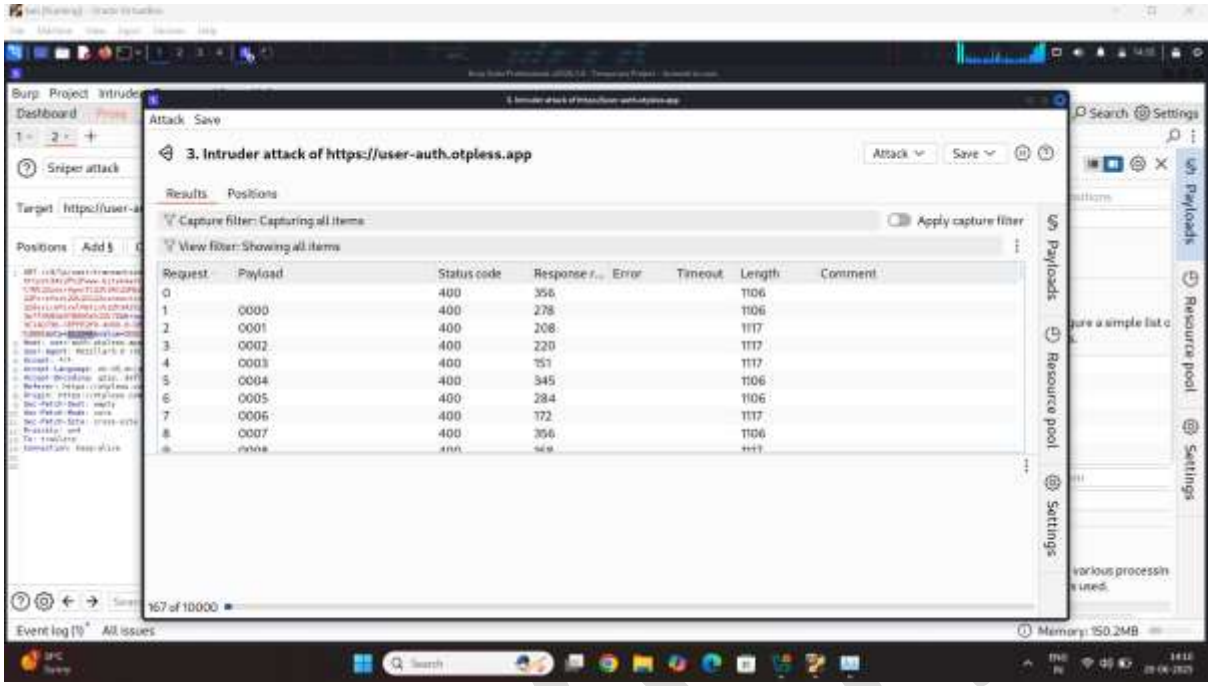


Step6: load dictionary in otp password



Step7:click on the attack and start the attack

result:



and otp success full by pass in brute force attack