Experiment No. 1

Batch: A2 **Roll No.:16010421073** **Experiment No.:1**

Aim: Comprehending security tools

Resources needed: Kali Linux Documentation, Kali Linux Forums, Burp Suite Documentation, Wireshark User Guide, Wireshark Sample Captures

Pre Lab/ Prior Concepts:

Students should have prior knowledge of Networking Basics, Linux Fundamentals, Web Technologies, Network Packet Analysis, Linux Command-Line Skills, Web Application Basics, Understanding of HTTP and HTTPS, Virtualization, Basic Security Concepts.

Theory:

Security tools serve different purposes but are often used together in security assessments to ensure a thorough examination of both web application and network security. In the dynamic landscape of cybersecurity, comprehending and mastering security tools is paramount for professionals engaged in ethical hacking, penetration testing, and network analysis. Three key tools—Kali Linux, Burp Suite, and Wireshark—stand out as indispensable assets in the cybersecurity arsenal.

1. **Kali Linux: The Swiss Army Knife of Security**

Kali Linux provides a comprehensive environment for ethical hackers and security professionals to test and assess the security of systems. Kali Linux, a Debian-based distribution, is purpose-built for penetration testing and security auditing. It encompasses a vast array of pre-installed tools, categorized for diverse tasks, ranging from network reconnaissance to exploitation. Its strength lies not only in its comprehensive suite of tools but also in its community-driven ethos, ensuring constant updates and adaptability to emerging threats. Professionals delve into Kali Linux to explore vulnerabilities, assess network defenses, and simulate real-world attacks in a controlled environment.

Purpose:

Kali Linux, a specialized Linux distribution, is meticulously crafted for the primary purpose of facilitating penetration testing, ethical hacking, and security auditing. Originally derived from Debian, Kali Linux has evolved into a comprehensive platform that provides security professionals, ethical hackers, and enthusiasts with a robust and versatile toolkit for identifying vulnerabilities, conducting security assessments, and fortifying systems against potential threats.

Features of Kali Linux

1. Vast Toolset
2. Open Source
3. Community-Driven Development
4. Targeted for Penetration Testing
5. Versatility in Deployment
6. Live Boot Capability
7. User-Friendly Interface
8. Security and Privacy Tools
9. Extensive Documentation

2. **Burp Suite: The Web Application Security Champion**

Burp Suite helps in identifying and fixing vulnerabilities in web applications by intercepting and manipulating HTTP/S traffic. Burp Suite emerges as a cornerstone in web application security testing. Its capabilities extend from mapping application architecture to identifying vulnerabilities, enabling ethical hackers to assess the security posture of web applications. The suite includes tools for scanning, crawling, and manipulating web requests, providing a robust platform for understanding, exploiting, and fortifying web-based systems. Its extensibility through custom plugins further enhances its adaptability to various testing scenarios.

Purpose of Burp Suite:

Burp Suite is a leading cybersecurity tool designed for web application security testing. Its primary purpose is to assist security professionals, ethical hackers, and penetration testers in identifying and addressing vulnerabilities within web applications. Burp Suite provides a comprehensive platform for manual and automated testing of web applications' security, helping organizations enhance their overall security posture.

Features of Burp Suite:

1. Intercept and Modify Requests
2. Automated Crawling and Discovery
3. Automated Vulnerability Scanner
4. Automated Attack Patterns
5. Manual Request Modification and Resubmission
6. Session Token and Cryptographic Analysis
7. Encoding and Decoding of Data
8. Comparison of Responses
9. Customization through Extension
10. Automated Analysis of Web Application
11. Detection of Out-of-Band Vulnerabilities

3. Wireshark: Unveiling the Secrets of Network Traffic

Wireshark, on the other hand, captures and analyzes network packets, enabling users to examine the flow of data and troubleshoot network issues. Wireshark, a powerful network protocol analyzer, provides a lens into network communication's intricacies. From capturing and analyzing packets to deciphering protocols and detecting anomalies, Wireshark is an invaluable tool for understanding network behavior. Security professionals leverage Wireshark to identify potential security threats, troubleshoot network issues, and gain insights into the flow of data. Its graphical interface and extensive filtering capabilities make it accessible to both novices and seasoned analysts.

Purpose of Wireshark:

Wireshark is a powerful open-source network protocol analyzer designed for network troubleshooting, analysis, software and communication protocol development, and education. Its primary purpose is to capture and examine the data traveling back and forth on a network, providing deep insights into network activity, performance, and potential security issues. Wireshark allows users to analyze and interpret packet-level data to diagnose network problems, understand network behavior, and ensure the efficient operation of networked systems.

Features of Wireshark

1. Real-Time Traffic Analysis
2. Examine Captured Files
3. Customize Views with Filters
4. Wide Range of Supported Protocols
5. Inspect Packet Contents
6. Visual Identification of Packets
7. Analyze Network Statistics
8. Reassemble TCP Streams
9. Voice over IP (VoIP) Analysis
10. Investigate Security Incidents

Procedure:

Kali Linix : Exploring Kali Linux involves getting familiar with its interface, understanding basic commands, and exploring the pre-installed tools.

Stepwise Procedure to Explore Kali Linux:

1. **Boot into Kali Linux**: Start by booting system into Kali Linux Familiarize with the Desktop Environment: Kali Linux typically uses the GNOME desktop environment
2. **Open Terminal and Learn Basic Commands**: Open the terminal, either by clicking on the terminal icon. Some essential commands include ls (list files), cd (change directory), pwd (print working directory), and sudo (execute commands with superuser privileges).
3. **Explore System Information:** Use commands like uname -a to display system information, df -h to show disk space usage, and free -m to view available memory.
4. **Check Network Configuration**: Use the ifconfig command to display network interface configurations. Explore the ping command to test network connectivity.
5. **Navigate through File System**: Use the cd command to navigate through the file system. Explore directories and files using ls and cd commands. Understand the concept of the root directory (/), home directory (~), and other essential directories.
6. **Explore Pre-Installed Tools**: Kali Linux comes with a plethora of pre-installed security tools. Explore the tools categorized in the Kali menu.

Burp Suite: Exploring Burp Suite involves familiarizing with its interface, understanding key features, and learning how to perform basic tasks related to web application security testing.

Stepwise Procedure to Explore Burp Suite:

1. **Download and Install Burp Suite**: Start by downloading Burp Suite from the official website (PortSwigger). Follow the installation instructions provided for operating system.
2. **Launch Burp Suite**: Once installed, launch Burp Suite. The interface consists of various tabs, including Proxy, Target, Intruder, Repeater, and more.
3. **Configure Browser to Use Burp Proxy**: To intercept and analyze HTTP/S traffic, configure web browser to use the Burp Proxy. Set the browser's proxy settings to match Burp Suite's proxy settings. The default proxy listener in Burp Suite is on 127.0.0.1 (localhost) and port 8080.
4. **Explore the Proxy Tab**: Go to the "Proxy" tab in Burp Suite. Observe the intercepted requests and responses in real-time. Get familiarize with the various options, such as turning interception on/off, forwarding requests, and responding to intercepted requests.
5. **Use the Target Tab**: Navigate to the "Target" tab. This tab provides information about the target web application, including site map, discovered content, and scope. Learn how to add a target, configure scope, and view the site map to understand the structure of the web application.
6. **Perform Basic Intruder and Repeater Tasks**: Explore the "Intruder" and "Repeater" tabs. In the "Intruder" tab to automate attacks such as brute force or fuzzing. In the "Repeater" tab, manually modify and re-send individual requests to observe their impact on the application.
7. **Review Other Essential Tabs**: Spend time exploring other essential tabs such as "Scanner," "Sequencer," and "Decoder." The "Scanner" tab allows automated scanning for common vulnerabilities, "Sequencer" analyzes the randomness of tokens, and "Decoder" assists in encoding and decoding data.

Wireshark: Exploring Wireshark involves capturing and analyzing network traffic to gain insights into communication protocols, troubleshoot network issues, and identify potential security threats.
Stepwise Procedure to Explore Wireshark:

1. **Download and Install Wireshark**: Start by downloading Wireshark from the official website (Wireshark Download). Follow the installation instructions provided for operating system.
2. **Launch Wireshark**: Once installed, launch Wireshark. The main interface will display a list of available network interfaces for capturing traffic. Select the appropriate interface for network connection.
3. **Start Capturing Packets**: Click on the selected interface to start capturing packets to see a live stream of captured packets displayed in real-time. Observe the various columns containing information such as source and destination addresses, protocols, and packet lengths.
4. **Filter Packets**: Utilize Wireshark's powerful filtering capabilities to focus on specific packets. Use display filters (e.g., ip.addr == 192.168.1.1) to narrow down the displayed packets based on criteria such as source/destination addresses, protocols, or specific keywords.
5. **Analyze Packet Details**: Select a packet from the captured list to view detailed information about its contents. Wireshark provides a hierarchical view of packet data, including protocol layers. Explore the dissected details to understand the structure of different protocols within the packet.
6. **Follow TCP Streams**: Use the "Follow TCP Stream" feature to reconstruct and display the entire conversation between two endpoints for a specific TCP stream. This is particularly useful for understanding the content and context of communication.
7. **Apply Colorization and Marking**: Wireshark uses colorization to highlight different types of packets, making it easier to identify and analyze specific types of network traffic. Take note of the color-coded packets to distinguish between various protocols and activities.

Output (Code with result Snapshot)

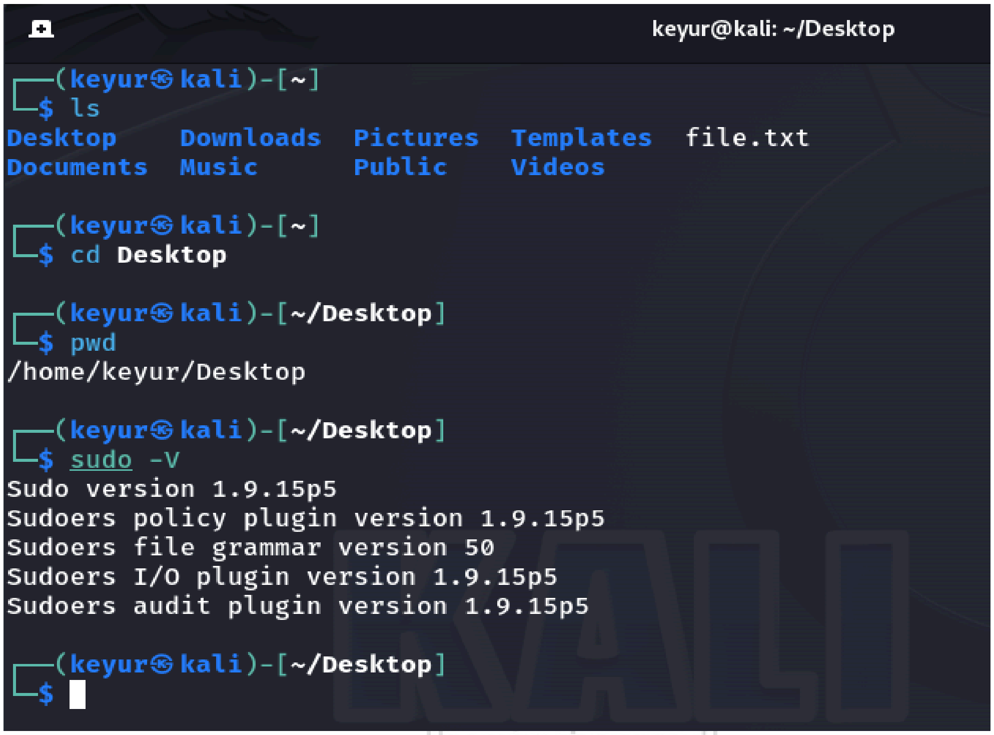
Kali-Linux

1) **Boot into Kali Linux**:

```
┌──(keyur㉿kali)-[~]
└─$ S
```

2) **Open Terminal and Learn Basic Commands**:

- **ls- Displaying various directories and files using ‘ls’ command.**
- **cd - Changing directory to desktop**
- **pwd- printing current working directory.**
- **sudo -V - Displaying version information using (Super Users do command)**

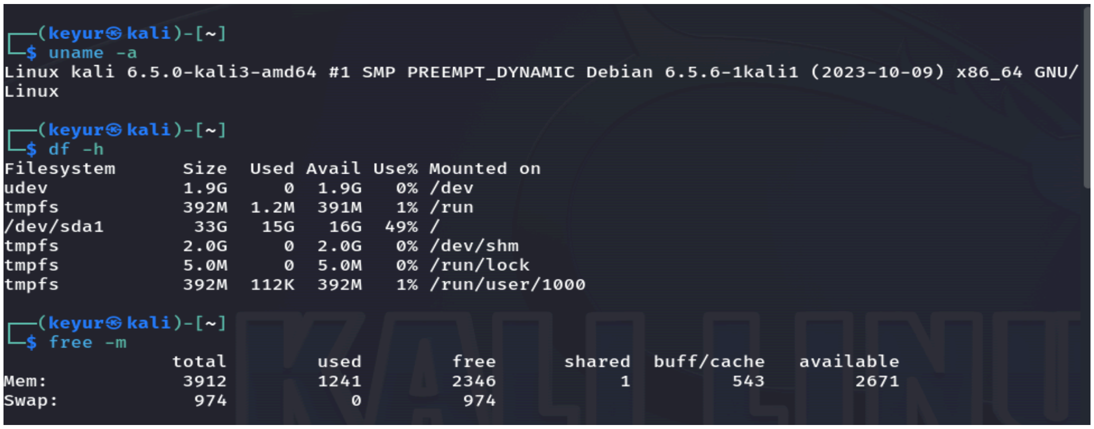


keyur@kali: ~/Desktop

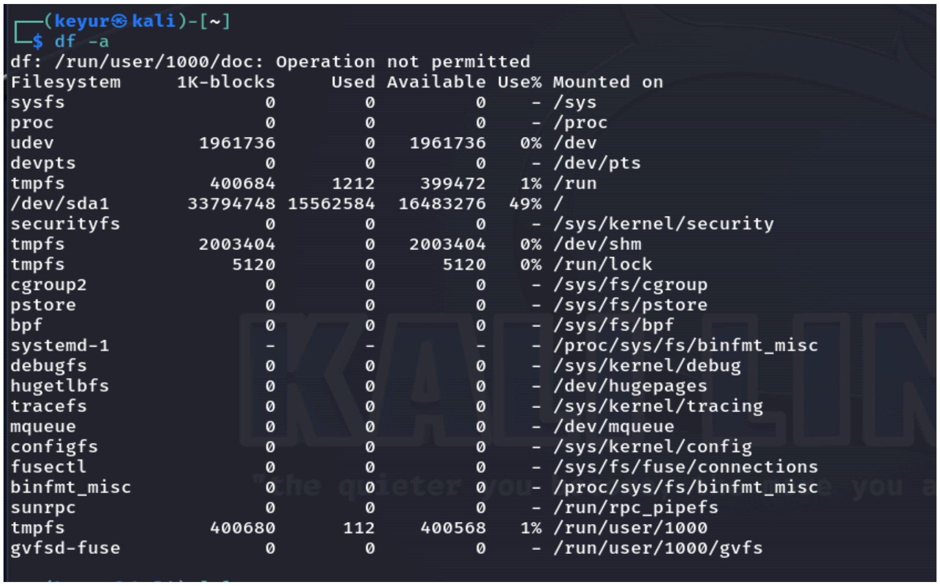
```
┌──(keyur㉿kali)-[~]
└─$ ls
Desktop     Downloads  Pictures  Templates  file.txt
Documents   Music      Public    Videos

┌──(keyur㉿kali)-[~]
└─$ cd Desktop

┌──(keyur㉿kali)-[~/Desktop]
└─$ pwd
/home/keyur/Desktop

┌──(keyur㉿kali)-[~/Desktop]
└─$ sudo -V
Sudo version 1.9.15p5
Sudoers policy plugin version 1.9.15p5
Sudoers file grammar version 50
Sudoers I/O plugin version 1.9.15p5
Sudoers audit plugin version 1.9.15p5

┌──(keyur㉿kali)-[~/Desktop]
└─$ 
```

3) **Explore System Information:**

- **uname -a : Displaying detailed information about system like kernel name, network node hostname, kernel release, kernel version, machine hardware architecture, and the operating system.**
- **df -h : This command is used in Unix-like operating systems to display information about disk space usage on mounted filesystems in a human-readable format.**
- **free -m : This command is used in Unix-like operating systems to display information about system memory usage.**
- **df -a: This command in Unix-like operating systems is used to display information about all filesystems, including those with 0 blocks (i.e., pseudo filesystems like /proc).**



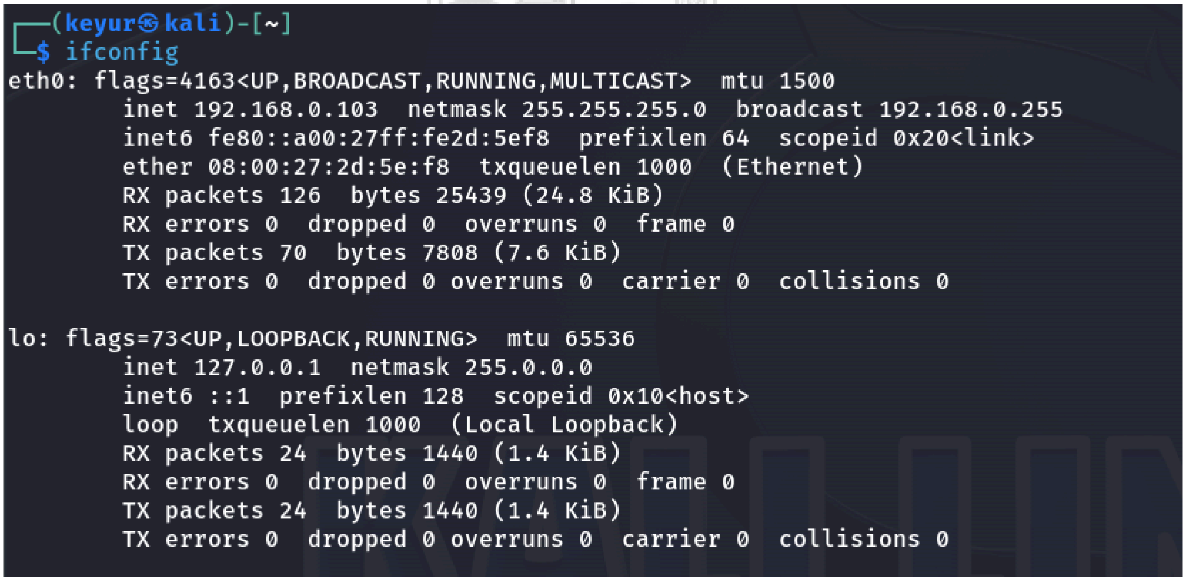
```
┌──(keyur㉿kali)-[~]
└─$ uname -a
Linux kali 6.5.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.6-1kali1 (2023-10-09) x86_64 GNU/
Linux

┌──(keyur㉿kali)-[~]
└─$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  1.2M  391M   1% /run
/dev/sda1        33G   15G   16G  49% /
tmpfs           2.0G     0  2.0G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           392M  112K  392M   1% /run/user/1000

┌──(keyur㉿kali)-[~]
└─$ free -m
               total        used        free      shared  buff/cache   available
Mem:            3912        1241        2346           1         543        2671
Swap:            974           0         974
```



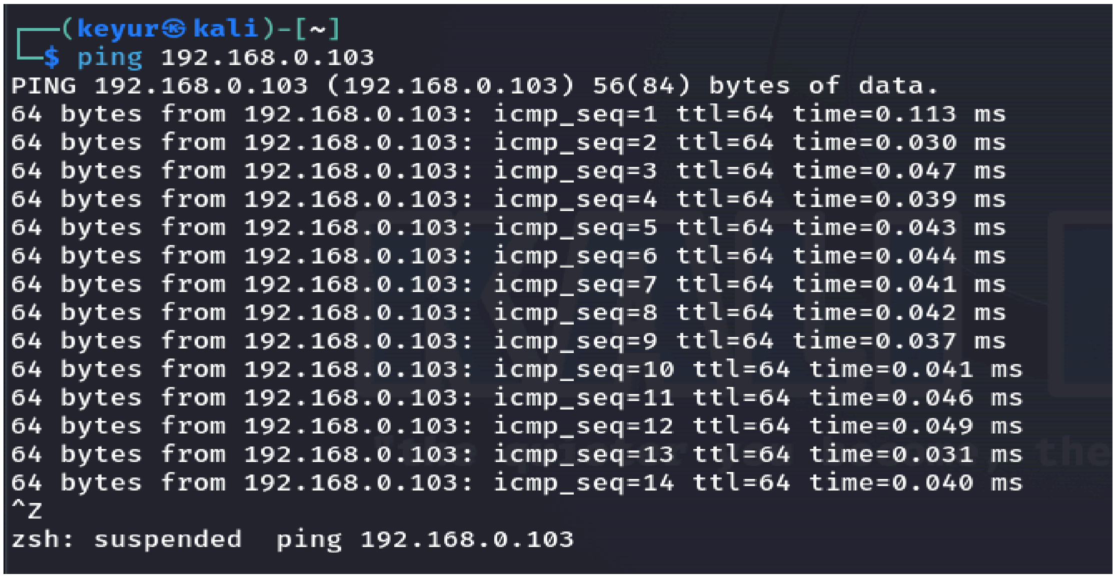
```
┌──(keyur㉿kali)-[~]
└─$ df -a
df: /run/user/1000/doc: Operation not permitted
Filesystem      1K-blocks     Used Available Use% Mounted on
sysfs                   0        0         0    - /sys
proc                    0        0         0    - /proc
udev              1961736        0   1961736   0% /dev
devpts                  0        0         0    - /dev/pts
tmpfs              400684     1212    399472   1% /run
/dev/sda1        33794748 15562584  16483276  49% /
securityfs              0        0         0    - /sys/kernel/security
tmpfs             2003404        0   2003404   0% /dev/shm
tmpfs                5120        0      5120   0% /run/lock
cgroup2                 0        0         0    - /sys/fs/cgroup
pstore                  0        0         0    - /sys/fs/pstore
bpf                     0        0         0    - /sys/fs/bpf
systemd-1               -        -         -    - /proc/sys/fs/binfmt_misc
debugfs                 0        0         0    - /sys/kernel/debug
hugetlbfs               0        0         0    - /dev/hugepages
tracefs                 0        0         0    - /sys/kernel/tracing
mqueue                  0        0         0    - /dev/mqueue
configfs                0        0         0    - /sys/kernel/config
fusectl                 0        0         0    - /sys/fs/fuse/connections
binfmt_misc             0        0         0    - /proc/sys/fs/binfmt_misc
sunrpc                  0        0         0    - /run/rpc_pipefs
tmpfs              400680      112    400568   1% /run/user/1000
gvfsd-fuse              0        0         0    - /run/user/1000/gvfs
```

4) Check Network Configuration:

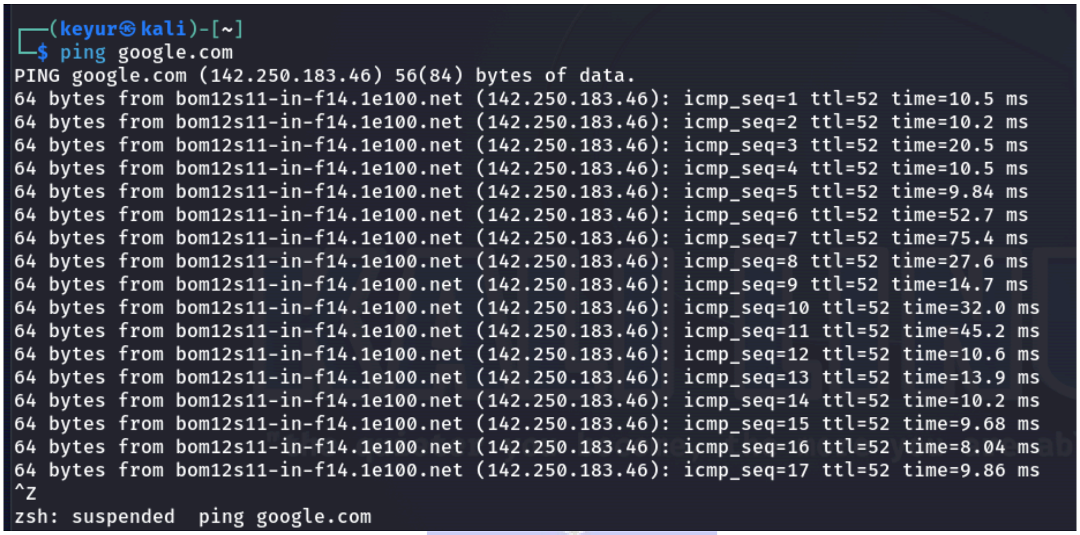
- **Ifconfig: The ifconfig command is used in Unix-like operating systems to display and configure network interfaces on a system. It provides information about the current network configuration, including IP addresses, network masks, hardware addresses (MAC addresses), and more.**
- **ping google.com: The ping command is used to test the reachability of a host on an Internet Protocol (IP) network.**



```
┌──(keyur㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.103  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe2d:5ef8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:2d:5e:f8  txqueuelen 1000  (Ethernet)
        RX packets 126  bytes 25439 (24.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 70  bytes 7808 (7.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 24  bytes 1440 (1.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1440 (1.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



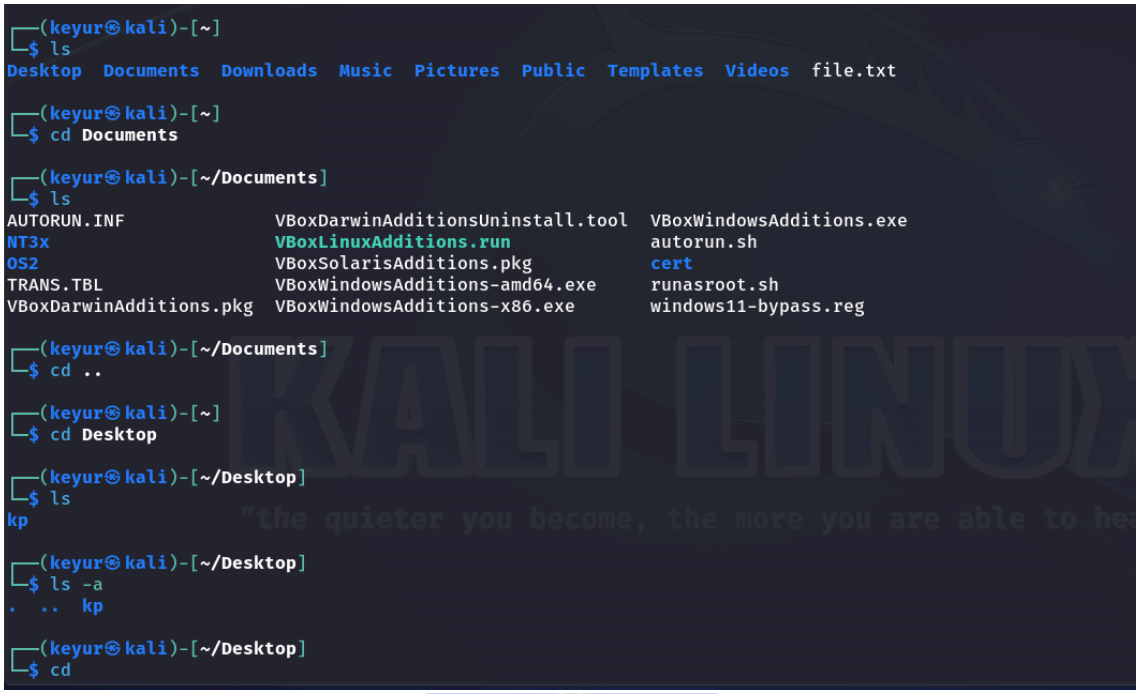
```
┌──(keyur㉿kali)-[~]
└─$ ping 192.168.0.103
PING 192.168.0.103 (192.168.0.103) 56(84) bytes of data.
64 bytes from 192.168.0.103: icmp_seq=1 ttl=64 time=0.113 ms
64 bytes from 192.168.0.103: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 192.168.0.103: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 192.168.0.103: icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from 192.168.0.103: icmp_seq=5 ttl=64 time=0.043 ms
64 bytes from 192.168.0.103: icmp_seq=6 ttl=64 time=0.044 ms
64 bytes from 192.168.0.103: icmp_seq=7 ttl=64 time=0.041 ms
64 bytes from 192.168.0.103: icmp_seq=8 ttl=64 time=0.042 ms
64 bytes from 192.168.0.103: icmp_seq=9 ttl=64 time=0.037 ms
64 bytes from 192.168.0.103: icmp_seq=10 ttl=64 time=0.041 ms
64 bytes from 192.168.0.103: icmp_seq=11 ttl=64 time=0.046 ms
64 bytes from 192.168.0.103: icmp_seq=12 ttl=64 time=0.049 ms
64 bytes from 192.168.0.103: icmp_seq=13 ttl=64 time=0.031 ms
64 bytes from 192.168.0.103: icmp_seq=14 ttl=64 time=0.040 ms
^Z
zsh: suspended  ping 192.168.0.103
```



```
┌──(keyur㉿kali)-[~]
└─$ ping google.com
PING google.com (142.250.183.46) 56(84) bytes of data.
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=1 ttl=52 time=10.5 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=2 ttl=52 time=10.2 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=3 ttl=52 time=20.5 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=4 ttl=52 time=10.5 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=5 ttl=52 time=9.84 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=6 ttl=52 time=52.7 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=7 ttl=52 time=75.4 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=8 ttl=52 time=27.6 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=9 ttl=52 time=14.7 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=10 ttl=52 time=32.0 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=11 ttl=52 time=45.2 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=12 ttl=52 time=10.6 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=13 ttl=52 time=13.9 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=14 ttl=52 time=10.2 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=15 ttl=52 time=9.68 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=16 ttl=52 time=8.04 ms
64 bytes from bom12s11-in-f14.1e100.net (142.250.183.46): icmp_seq=17 ttl=52 time=9.86 ms
^Z
zsh: suspended  ping google.com
```

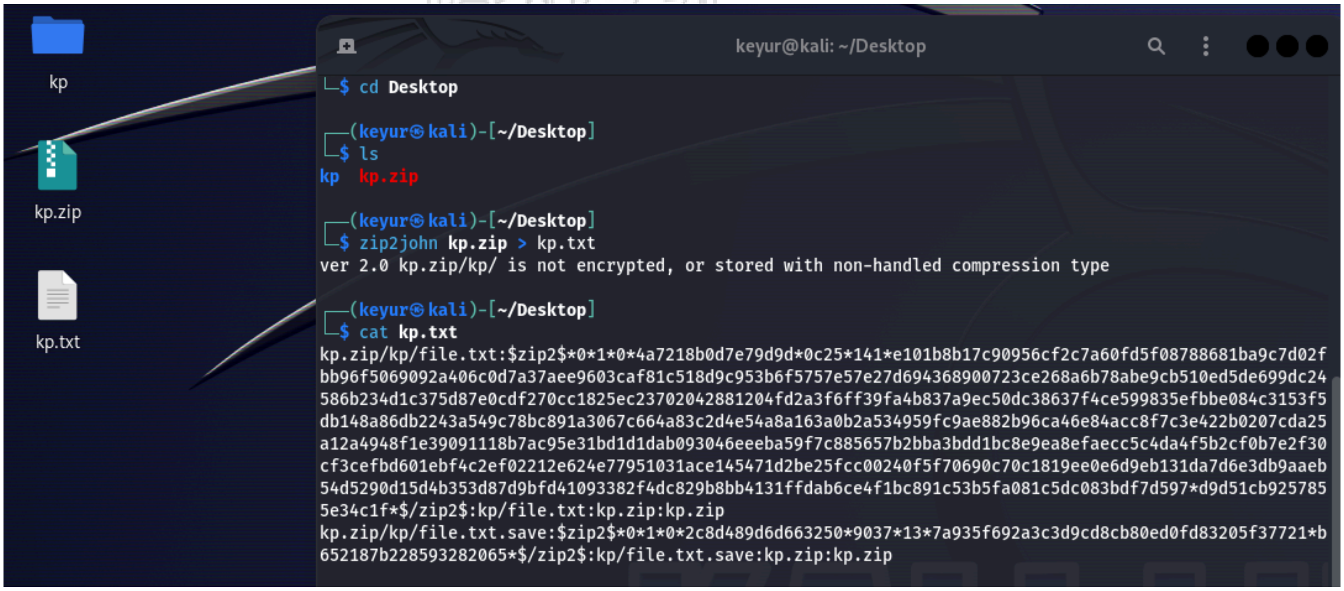
5) **Navigate through File System**:

- **cd - changing directories to documents and desktop.**
- **cd .. - previous directory**
- **ls -a - listing directory and files.**

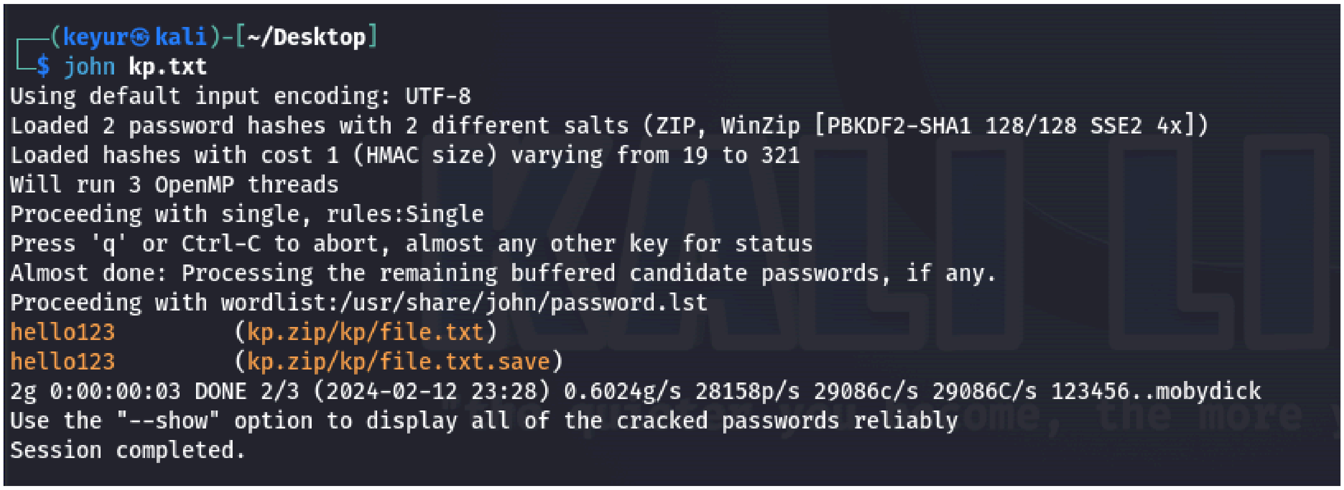


```
┌──(keyur㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  file.txt

┌──(keyur㉿kali)-[~]
└─$ cd Documents

┌──(keyur㉿kali)-[~/Documents]
└─$ ls
AUTORUN.INF               VBoxDarwinAdditionsUninstall.tool  VBoxWindowsAdditions.exe
NT3x                      VBoxLinuxAdditions.run             autorun.sh
OS2                       VBoxSolarisAdditions.pkg           cert
TRANS.TBL                 VBoxWindowsAdditions-amd64.exe     runasroot.sh
VBoxDarwinAdditions.pkg   VBoxWindowsAdditions-x86.exe       windows11-bypass.reg

┌──(keyur㉿kali)-[~/Documents]
└─$ cd ..

┌──(keyur㉿kali)-[~]
└─$ cd Desktop

┌──(keyur㉿kali)-[~/Desktop]
└─$ ls
kp

┌──(keyur㉿kali)-[~/Desktop]
└─$ ls -a
.  ..  kp

┌──(keyur㉿kali)-[~/Desktop]
└─$ cd
```

6) **Explore Pre-Installed Tools**:
John The Ripper(Password cracking for a simple zip file)



```
└─$ cd Desktop

┌──(keyur㉿kali)-[~/Desktop]
└─$ ls
kp  kp.zip

┌──(keyur㉿kali)-[~/Desktop]
└─$ zip2john kp.zip > kp.txt
ver 2.0 kp.zip/kp/ is not encrypted, or stored with non-handled compression type

┌──(keyur㉿kali)-[~/Desktop]
└─$ cat kp.txt
kp.zip/kp/file.txt:$zip2$*0*1*0*4a7218b0d7e79d9d*0c25*141*e101b8b17c90956cf2c7a60fd5f08788681ba9c7d02f
bb96f5069092a406c0d7a37aee9603caf81c518d9c953b6f5757e57e27d694368900723ce268a6b78abe9cb510ed5de699dc24
586b234d1c375d87e0cdf270cc1825ec23702042881204fd2a3f6ff39fa4b837a9ec50dc38637f4ce599835efbbe084c3153f5
db148a86db2243a549c78bc891a3067c664a83c2d4e54a8a163a0b2a534959fc9ae882b96ca46e84acc8f7c3e422b0207cda25
a12a4948f1e39091118b7ac95e31bd1d1dab093046eeeba59f7c885657b2bba3bdd1bc8e9ea8efaecc5c4da4f5b2cf0b7e2f30
cf3cefbd601ebf4c2ef02212e624e77951031ace145471d2be25fcc00240f5f70690c70c1819ee0e6d9eb131da7d6e3db9aaeb
54d5290d15d4b353d87d9bfd41093382f4dc829b8bb4131ffdab6ce4f1bc891c53b5fa081c5dc083bdf7d597*d9d51cb925785
5e34c1f*$/zip2$:kp/file.txt:kp.zip:kp.zip
kp.zip/kp/file.txt.save:$zip2$*0*1*0*2c8d489d6d663250*9037*13*7a935f692a3c3d9cd8cb80ed0fd83205f37721*b
652187b228593282065*$/zip2$:kp/file.txt.save:kp.zip:kp.zip
```



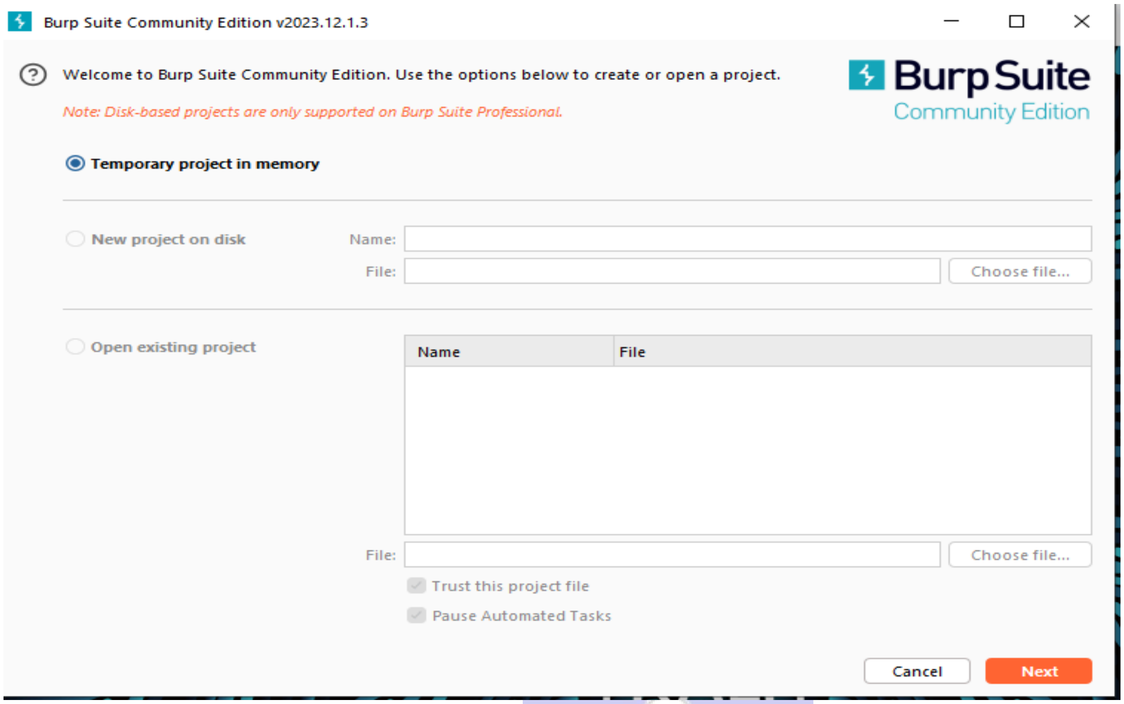
```
┌──(keyur㉿kali)-[~/Desktop]
└─$ john kp.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (ZIP, WinZip [PBKDF2-SHA1 128/128 SSE2 4x])
Loaded hashes with cost 1 (HMAC size) varying from 19 to 321
Will run 3 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
hello123         (kp.zip/kp/file.txt)
hello123         (kp.zip/kp/file.txt.save)
2g 0:00:00:03 DONE 2/3 (2024-02-12 23:28) 0.6024g/s 28158p/s 29086c/s 29086C/s 123456..mobydick
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

The yellow text is password for zip file 'kp'.

<u>Burpsuite:</u>

1. **Launch Burp Suite**:

2. **Creating a new project.**



Burp Suite Community Edition v2023.12.1.3

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

● Temporary project in memory

○ New project on disk — Name: — File:

○ Open existing project

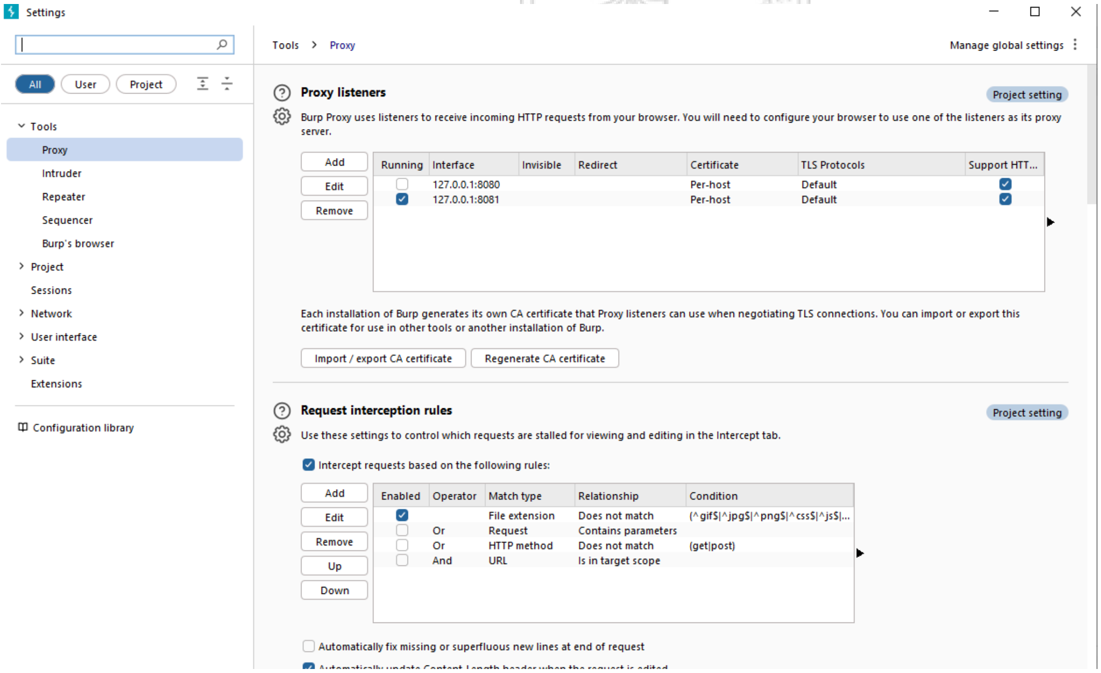
Name	File

File:

☑ Trust this project file

☑ Pause Automated Tasks

3) Getting Familiar with proxy settings.



Settings

Tools › Proxy

Proxy listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols	Support HTT...
☐	127.0.0.1:8080			Per-host	Default	☑
☑	127.0.0.1:8081			Per-host	Default	☑

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools or another installation of Burp.

Request interception rules

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

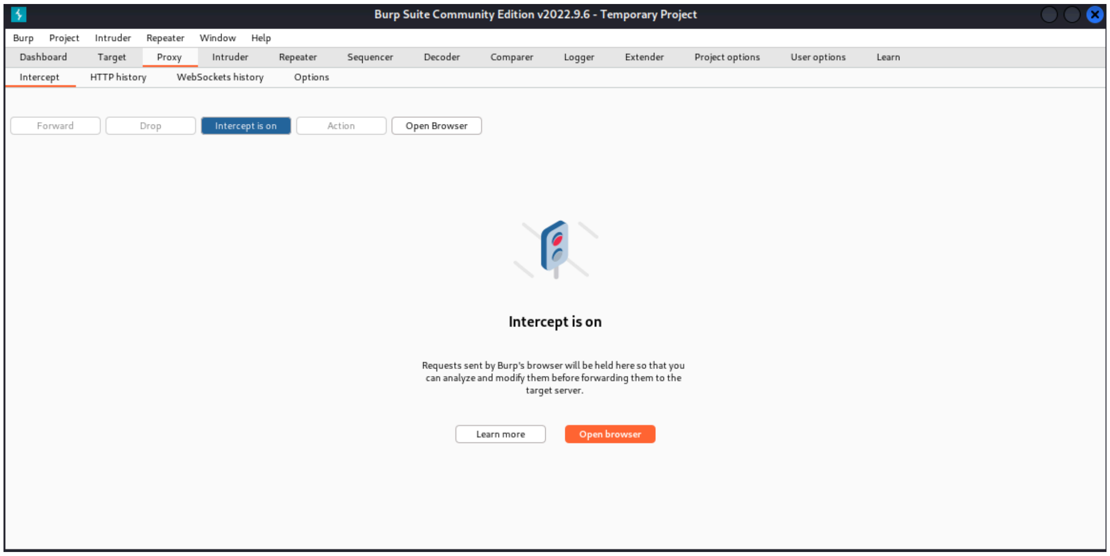
☑ Intercept requests based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$|...
☐	Or	Request	Contains parameters	
☐	Or	HTTP method	Does not match	(get|post)
☐	And	URL	Is in target scope	

☐ Automatically fix missing or superfluous new lines at end of request

☑ Automatically update Content-Length header when the request is edited

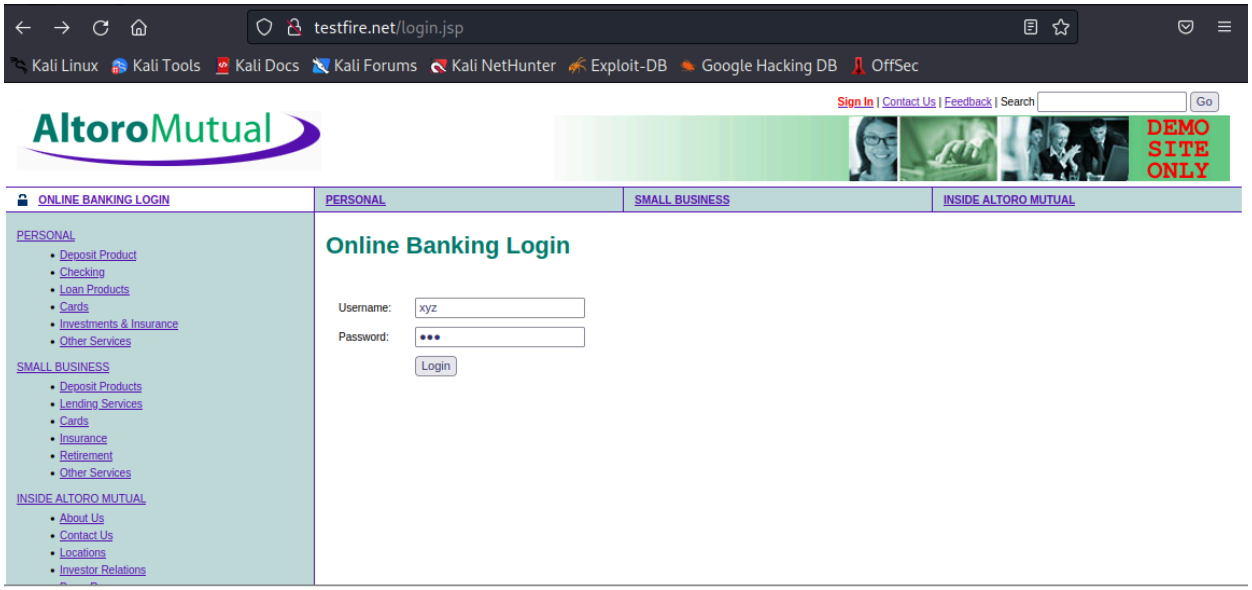
4) After setting it up we start the intercept:


Burp Suite Community Edition v2022.9.6 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Learn more
Open browser

Intercept is on

Requests sent by Burp's browser will be held here so that you can analyze and modify them before forwarding them to the target server.

5) After going to a login page:

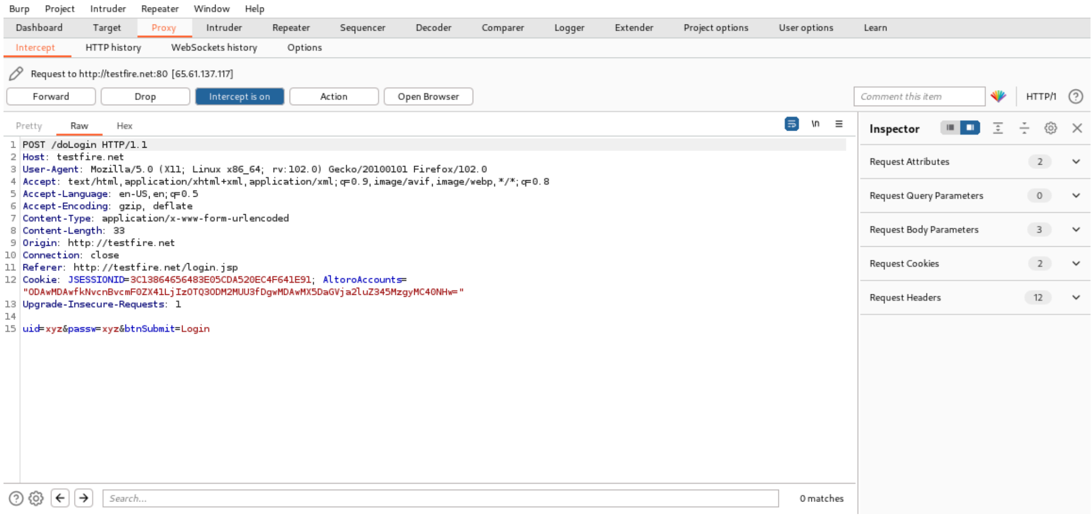


Online Banking Login

Username: xyz

Password: •••

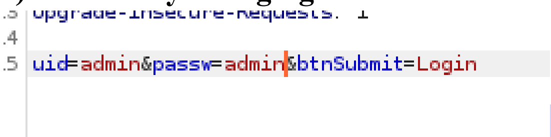
6) First the request is sent into the Burp:


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Comment this item
HTTP/1
Pretty Raw Hex
Inspector
Request Attributes 2
Request Query Parameters 0
Request Body Parameters 3
Request Cookies 2
Request Headers 12
Search...
0 matches

Request to http://testfire.net:80 [65.61.137.117]

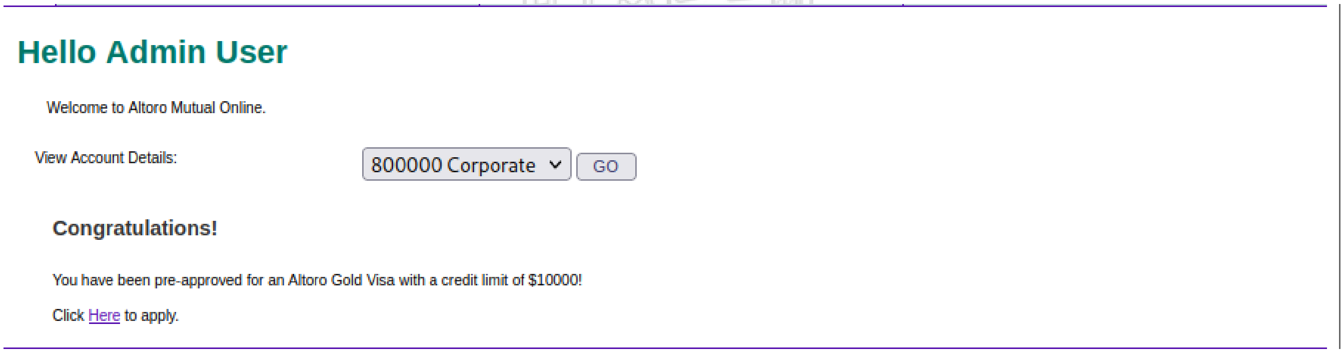
```
POST /doLogin HTTP/1.1
Host: testfire.net
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
Origin: http://testfire.net
Connection: close
Referer: http://testfire.net/login.jsp
Cookie: JSESSIONID=3C13864656483E05CDA520EC4F641E91; AltoroAccounts=
"ODAwMDAwfkNvcnBvcmF0ZX41LjIzOTQ3ODM2MUU3fDgwMDAwMX5DaGVja2luZ345MzgyMC40NHw="
Upgrade-Insecure-Requests: 1

uid=xyz&passw=xyz&btnSubmit=Login
```

7) Manually changing the UID and Password:



```
Upgrade-Insecure-Requests: 1

uid=admin&passw=admin&btnSubmit=Login
```

8) Forward the Request:



Hello Admin User

Welcome to Altoro Mutual Online.

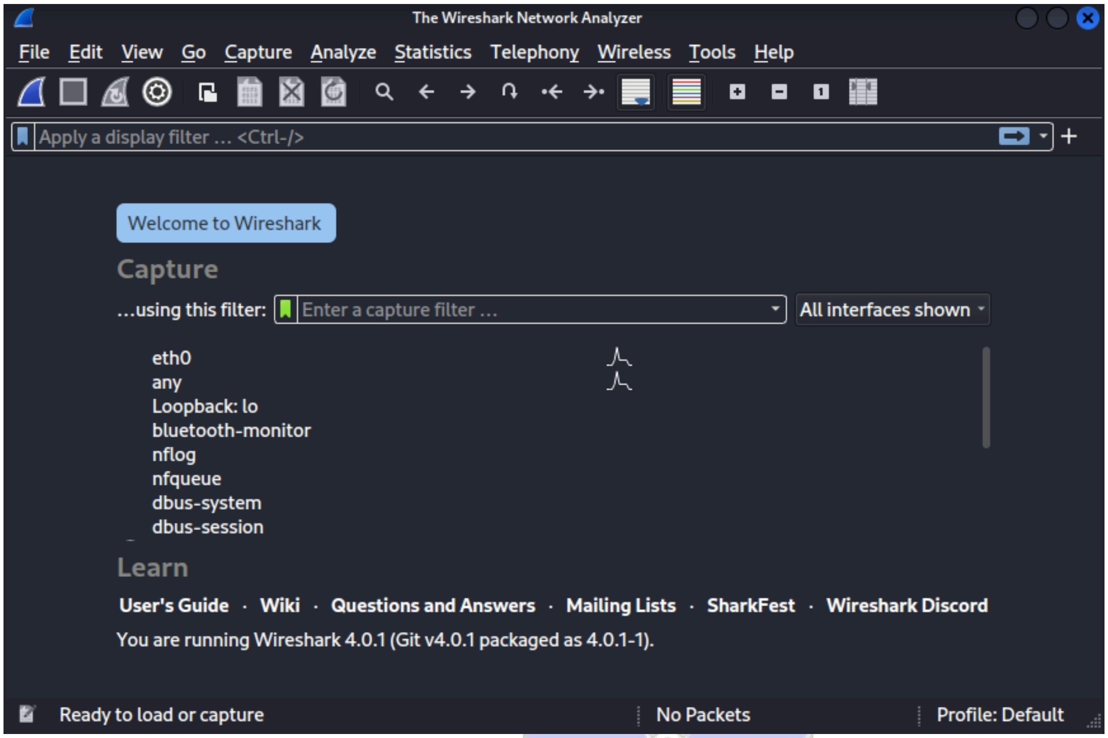
View Account Details: 800000 Corporate GO

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

Wireshark:


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark

Capture

…using this filter: Enter a capture filter … All interfaces shown

```
eth0
any
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
```

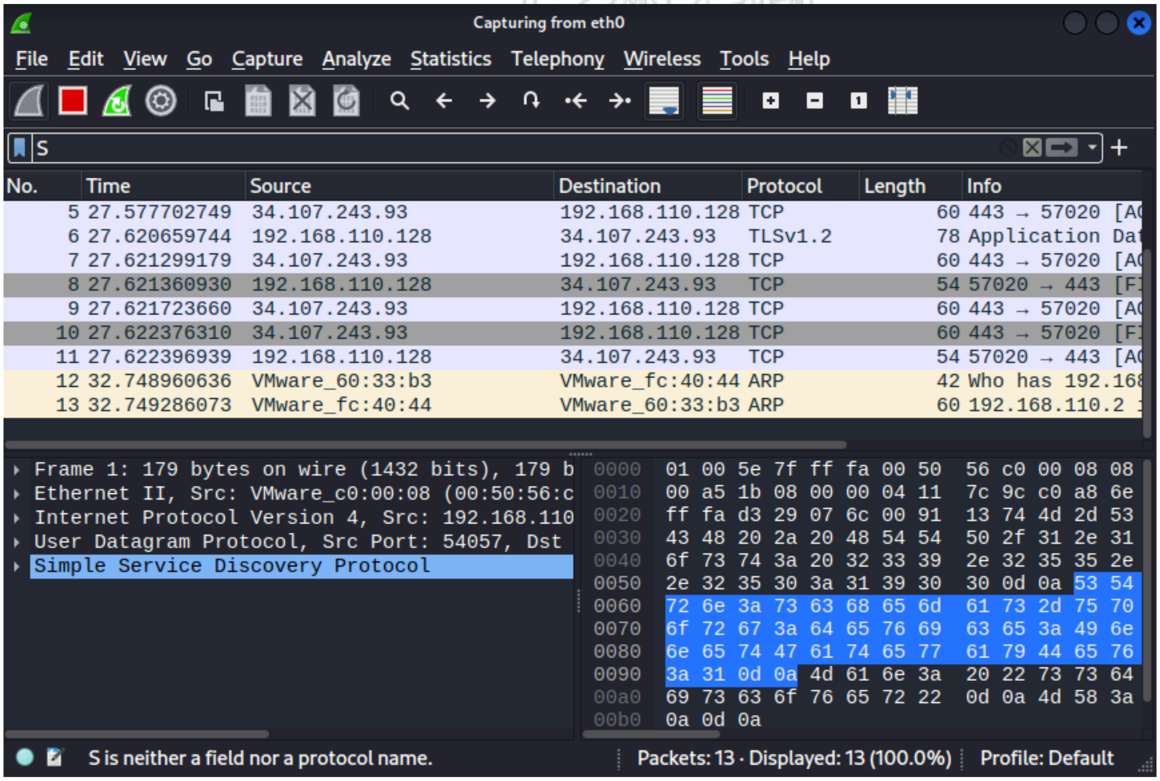
Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord

You are running Wireshark 4.0.1 (Git v4.0.1 packaged as 4.0.1-1).

Ready to load or capture | No Packets | Profile: Default

EXPLORING THE PACKETS:


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

S

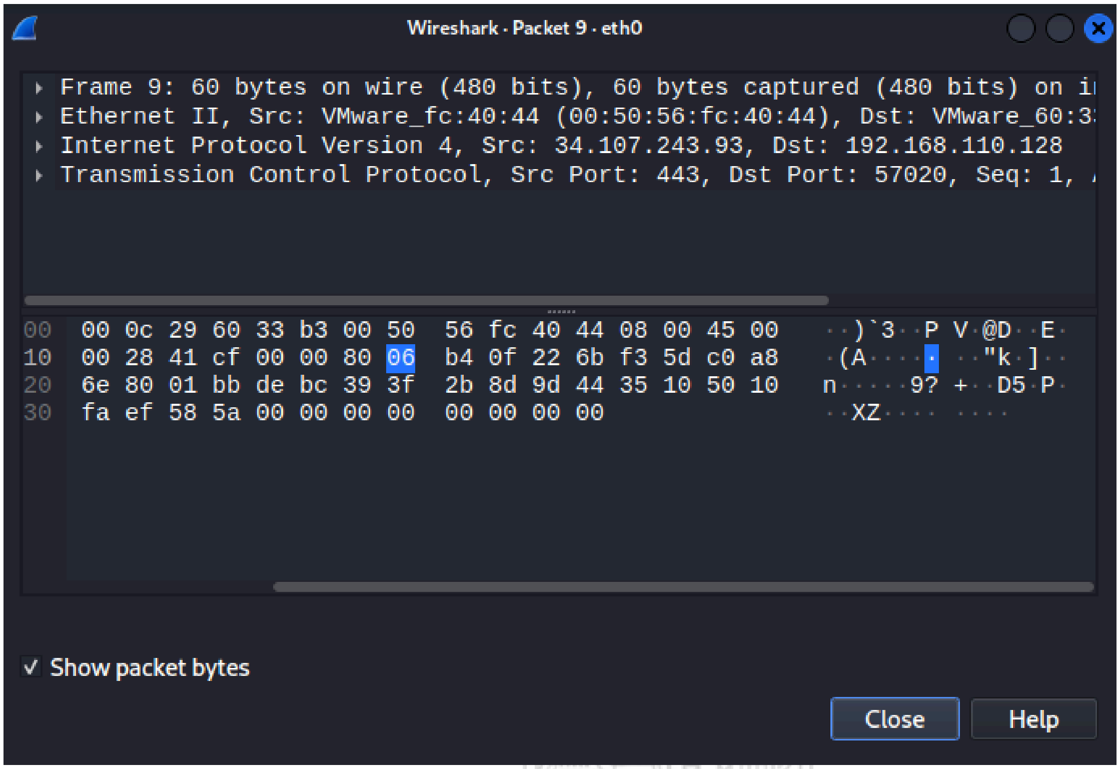
```
No.  Time         Source           Destination      Protocol Length Info
   5 27.577702749 34.107.243.93    192.168.110.128  TCP          60 443 → 57020 [AC
   6 27.620659744 192.168.110.128  34.107.243.93    TLSv1.2      78 Application Dat
   7 27.621299179 34.107.243.93    192.168.110.128  TCP          60 443 → 57020 [AC
   8 27.621360930 192.168.110.128  34.107.243.93    TCP          54 57020 → 443 [FI
   9 27.621723660 34.107.243.93    192.168.110.128  TCP          60 443 → 57020 [AC
  10 27.622376310 34.107.243.93    192.168.110.128  TCP          60 443 → 57020 [FI
  11 27.622396939 192.168.110.128  34.107.243.93    TCP          54 57020 → 443 [AC
  12 32.748960636 VMware_60:33:b3  VMware_fc:40:44  ARP          42 Who has 192.168
  13 32.749286073 VMware_fc:40:44  VMware_60:33:b3  ARP          60 192.168.110.2 i
```

```
› Frame 1: 179 bytes on wire (1432 bits), 179 b
› Ethernet II, Src: VMware_c0:00:08 (00:50:56:c
› Internet Protocol Version 4, Src: 192.168.110
› User Datagram Protocol, Src Port: 54057, Dst
› Simple Service Discovery Protocol
```

```
0000  01 00 5e 7f ff fa 00 50  56 c0 00 08 08
0010  00 a5 1b 08 00 00 04 11  7c 9c c0 a8 6e
0020  ff fa d3 29 07 6c 00 91  13 74 4d 2d 53
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31
0040  6f 73 74 3a 20 32 33 39  2e 32 35 35 2e
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 53 54
0060  72 6e 3a 73 63 68 65 6d  61 73 2d 75 70
0070  6f 72 67 3a 64 65 76 69  63 65 3a 49 6e
0080  6e 65 74 47 61 74 65 77  61 79 44 65 76
0090  3a 31 0d 0a 4d 61 6e 3a  20 22 73 73 64
00a0  69 73 63 6f 76 65 72 22  0d 0a 4d 58 3a
00b0  0a 0d 0a
```

S is neither a field nor a protocol name. | Packets: 13 · Displayed: 13 (100.0%) | Profile: Default

SHOWING A PACKET DETAILS:

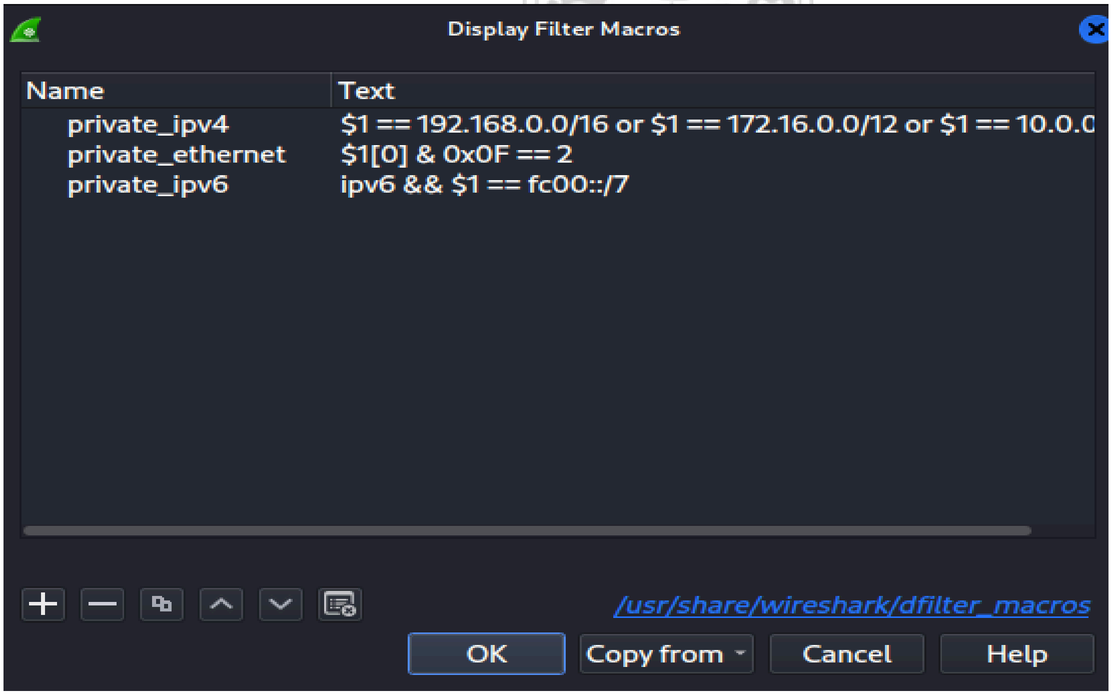


```
▸ Frame 9: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on i
▸ Ethernet II, Src: VMware_fc:40:44 (00:50:56:fc:40:44), Dst: VMware_60:3
▸ Internet Protocol Version 4, Src: 34.107.243.93, Dst: 192.168.110.128
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 57020, Seq: 1,
```

```
00  00 0c 29 60 33 b3 00 50  56 fc 40 44 08 00 45 00   ··)`3··P V·@D··E·
10  00 28 41 cf 00 00 80 06  b4 0f 22 6b f3 5d c0 a8   ·(A····· ··"k·]··
20  6e 80 01 bb de bc 39 3f  2b 8d 9d 44 35 10 50 10   n·····9? +··D5·P·
30  fa ef 58 5a 00 00 00 00  00 00 00 00               ··XZ···· ····
```

☑ Show packet bytes

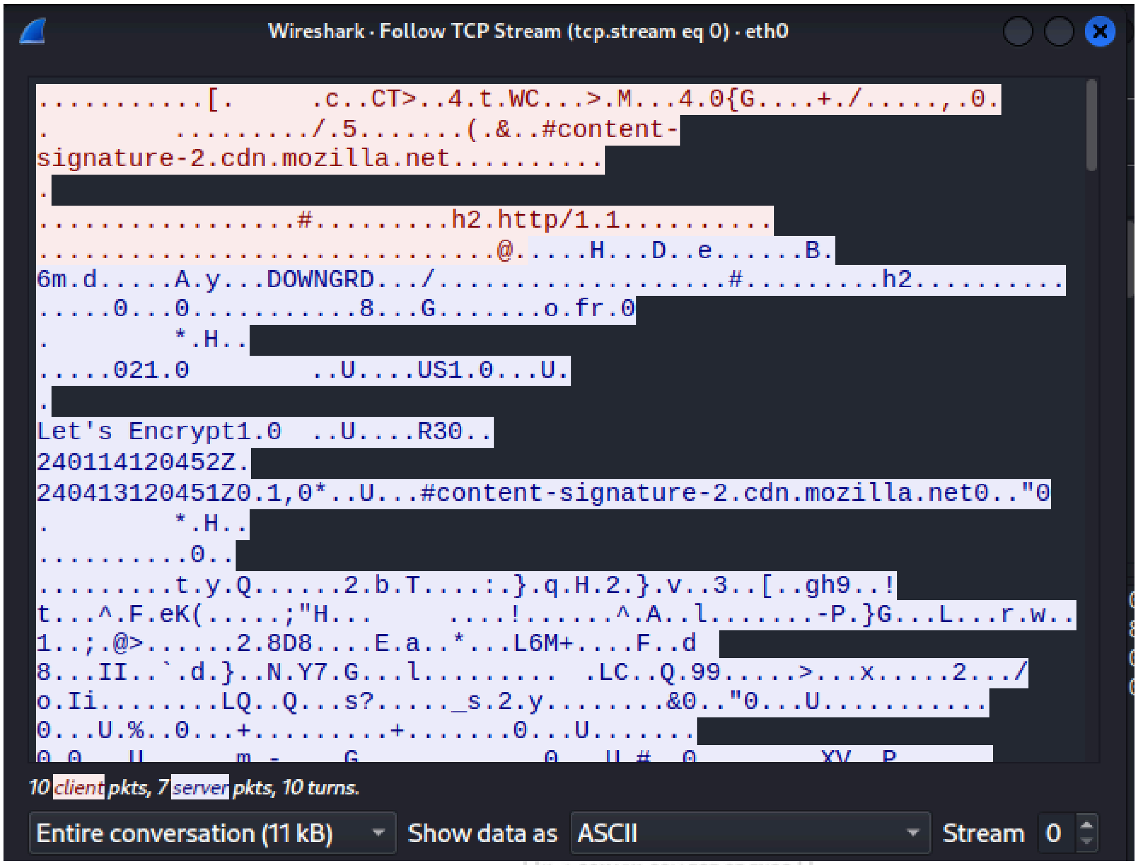
ANALYZING THE PACKETS:



Name	Text
private_ipv4	$1 == 192.168.0.0/16 or $1 == 172.16.0.0/12 or $1 == 10.0.0
private_ethernet	$1[0] & 0x0F == 2
private_ipv6	ipv6 && $1 == fc00::/7

/usr/share/wireshark/dfilter_macros

FOLLOWING TCP Streams:



```
...........[.     .c..CT>..4.t.WC...>.M...4.0{G....+./.....,.0.
.        ........./.5.......(.&..#content-
signature-2.cdn.mozilla.net..........
.
.................#.........h2.http/1.1..........
..............................@.....H...D..e......B.
6m.d.....A.y...DOWNGRD.../.....................#.........h2..........
.....0...0...........8...G.......o.fr.0
.        *.H..
.....021.0        ..U....US1.0...U.
.
Let's Encrypt1.0  ..U....R30..
240114120452Z.
240413120451Z0.1,0*..U...#content-signature-2.cdn.mozilla.net0.."0
.        *.H..
..........0..
.........t.y.Q......2.b.T....:.}.q.H.2.}.v..3..[..gh9..!
t...^.F.eK(.....;"H...     ....!......^.A..l.......-P.}G...L...r.w..
1..;.@>......2.8D8....E.a..*...L6M+....F..d
8...II..`.d.}..N.Y7.G...l.........  .LC..Q.99.....>...x.....2.../
o.Ii........LQ..Q...s?....._s.2.y........&0.."0...U..........
0...U.%..0...+.........+.......0...U.......
```

10 client pkts, 7 server pkts, 10 turns.

Entire conversation (11 kB) | Show data as: ASCII | Stream: 0

Post Lab Questions: -

1. **Describe a specific scenario to utilize Kali Linux tools for a penetration testing engagement. Identify at least two tools from Kali's arsenal and explain their roles in addressing security vulnerabilities. How to approach a simulated network assessment using these tools?**

   Ans: Sure, let's consider a scenario where a company has hired you to perform a penetration testing engagement on their internal network to identify and address potential security vulnerabilities. For this simulation, we'll focus on using Kali Linux tools to conduct the assessment.

   Scenario:
   You have been given the task to assess the security of a company's internal network, which includes servers, workstations, and network infrastructure. The goal is to identify vulnerabilities that could be exploited by malicious actors.

   Kali Linux Tools and Their Roles:

   1. Nmap (Network Mapper):

   - **Role**: Nmap is a powerful network scanning tool that can be used to discover hosts, services, and open ports on a network.
   - **Utilization**: Start with an initial reconnaissance by running an Nmap scan on the target network to identify live hosts and open ports. This information will help you understand the network topology and potential entry points.

   2. Metasploit:

   - **Role:** Metasploit is a penetration testing framework that allows you to develop, test, and execute exploits against a target system.
   - **Utilization:** After identifying open ports and services with Nmap, use Metasploit to find and exploit vulnerabilities. For example, if a vulnerable version of a service is running on a target machine, Metasploit can automate the process of exploiting it.

   Simulated Network Assessment Approach:

   1. Reconnaissance:

   - Use Nmap to conduct an initial scan to identify live hosts and open ports on the target network.
   - Gather information about the target systems, such as operating systems and services running on open ports.

   2. Vulnerability Scanning:

   - Utilize tools like OpenVAS or Nessus from Kali Linux to perform vulnerability scans on the identified hosts.

- Identify potential weaknesses and prioritize them based on their severity.

3. Exploitation:

- Use Metasploit to exploit known vulnerabilities on the target systems.
- Gain access to vulnerable machines and escalate privileges if possible.

4. Post-exploitation:

- Once access is gained, perform post-exploitation activities to assess the extent of the compromise.
- Evaluate the security measures in place to detect and respond to the intrusion.

5. Documentation:

- Document the findings, including identified vulnerabilities, exploited systems, and recommendations for remediation.
- Provide a detailed report to the client, outlining the security posture of their network and suggesting improvements.

It's crucial to conduct penetration testing within a legal and authorized framework. Always ensure you have explicit permission from the client before attempting any penetration testing activities.

2. **In a web application security testing scenario, outline the steps to be taken using Burp Suite's Proxy and Scanner modules. How does intercepting and modifying requests contribute to the assessment, and what types of vulnerabilities can the automated scanner detect? Provide a step-by-step walkthrough.**

Ans:

1. **Configure Browser to Use Burp Proxy:**

- Set up your browser to use Burp Suite as a proxy. Configure the browser's proxy settings to point to Burp's proxy listener (default is usually `127.0.0.1:8080`).

2. **Enable Intercept in Burp Proxy:**

- In Burp Suite, go to the "Proxy" tab and make sure the "Intercept is on" button is clicked. This allows you to intercept and modify HTTP requests and responses.

3. **Explore the Application:**

- Navigate through the web application, allowing Burp to capture and display requests and responses in the "Proxy" tab. This helps you understand the application's functionality and flow.

4. **Intercept and Modify Requests**:

- Identify interesting requests (login, form submissions, etc.) and intercept them using Burp.
- Modify parameters, headers, or other data in the request to observe how the application responds.
- This helps in identifying potential security flaws such as injection vulnerabilities.

5. **Use Burp Scanner:**

- In Burp Suite, go to the "Scanner" tab and configure your scan settings (target, scope, etc.).
- Start an automated scan using the Burp Scanner to identify common web application vulnerabilities.

6. **Review Scanner Findings:**

- Burp Scanner will perform automated checks for various vulnerabilities such as SQL injection, cross-site scripting (XSS), and more.
- Review the scanner findings in the "Scanner" tab, and verify the identified vulnerabilities manually.

Intercepting and Modifying Requests:

- **Purpose:** Intercepting requests allows you to observe and modify data before it reaches the server. This is crucial for identifying security vulnerabilities, such as injection attacks or insecure direct object references.

- **Example**: SQL Injection
  - Intercept a login request.
  - Modify the username or password field by injecting SQL code to see if the application is vulnerable.
  - Analyze how the application responds to manipulated input.

Automated Scanner Capabilities:

1. SQL Injection (SQLi):
- Automated scanner checks for SQL injection vulnerabilities by injecting malicious SQL code into input fields.

2. Cross-Site Scripting (XSS):
- Identifies places where user input is not properly sanitized, allowing for the injection of malicious scripts.

3. Cross-Site Request Forgery (CSRF)
- Detects potential CSRF vulnerabilities by analyzing how the application handles state-changing requests.

4. Security Misconfigurations:
- Scans for misconfigurations in the web application or server settings that could expose sensitive information.

5. Out-of-date Software:
- Identifies outdated software components or libraries that may have known vulnerabilities.

By following these steps and combining automated scanning with manual testing, you can comprehensively assess the security of a web application using Burp Suite.

3. **Explain how Wireshark could be instrumental in identifying the root cause of the network connectivity issue. Provide specific examples of network anomalies or errors that Wireshark can help uncover and the corresponding steps to be taken form mitigation.**
   Ans: Wireshark is a powerful network protocol analyzer that allows you to capture and inspect the traffic on a network. It can be instrumental in identifying the root cause of network connectivity issues by analyzing the packets exchanged between devices. Here are some examples of network anomalies or errors that Wireshark can help uncover and steps to mitigate them:

Example Scenarios and Mitigation Steps:

1. Packet Loss:

- Symptoms: Users report intermittent connectivity issues or slow network performance.

- Wireshark Analysis:

- Capture traffic and look for a high number of retransmissions, duplicate ACKs, or ICMP messages indicating packet loss.

- Mitigation:

- Identify and resolve the cause of packet loss, which could be due to network congestion, faulty hardware, or interference.

2. Network Latency:

- Symptoms: Users experience delays when accessing applications or services.

- Wireshark Analysis:

- Analyze round-trip times (RTTs) in the captured packets to identify delays.

- Look for TCP handshake delays or delayed responses from servers.

- Mitigation:

- Optimize network configurations, address routing issues, or consider upgrading network infrastructure to reduce latency.

3. DNS Resolution Issues:

- Symptoms: Users report difficulty accessing websites or services by domain names.

- Wireshark Analysis:

- Look for DNS queries and responses to identify any delays or failed resolutions.

- Mitigation:

- Investigate DNS server configurations, check for DNS server availability, and resolve any issues affecting DNS resolution.

4. TCP Connection Failures:

- Symptoms: Users encounter errors when trying to establish TCP connections to specific services.

- **Wireshark Analysis:**

  - Inspect TCP handshake packets and check for any anomalies in the connection establishment process.

- **Mitigation:**

  - Address firewall rules, network ACLs, or routing issues that may be blocking the establishment of TCP connections.

5. Broadcast Storms:

- **Symptoms:** Network becomes slow or unresponsive for all users.

- **Wireshark Analysis:**

  - Observe a high volume of broadcast or multicast traffic in the captured packets.

- **Mitigation:**

  - Identify and isolate the source of the broadcast storm, which could be caused by a misconfigured device or a malfunctioning network component.

6. ARP Spoofing or Poisoning:

- **Symptoms:** Unexpected network behavior, such as unauthorized access or man-in-the-middle attacks.

- **Wireshark Analysis:**

  - Monitor Address Resolution Protocol (ARP) traffic for unusual or inconsistent MAC address mappings.

- **Mitigation:**

  - Implement ARP spoofing detection tools or use static ARP entries to prevent unauthorized address resolutions.

General Steps for Mitigation:

1. Identify the Root Cause:

- Use Wireshark to pinpoint the specific network anomaly or error causing connectivity issues.

2. Isolate the Affected Component:

- Determine whether the issue is localized to a specific device, network segment, or application.

3. Address Configuration Issues:

- Correct misconfigurations in routers, switches, firewalls, or DNS servers identified through Wireshark analysis.

4. Optimize Network Infrastructure:

- Upgrade hardware, address bandwidth limitations, and optimize routing to improve overall network performance.

5. Implement Security Measures:

- Apply security measures to mitigate issues such as broadcast storms, ARP spoofing, or other security-related anomalies.

6. Monitor and Test:

- Continuously monitor the network and perform periodic tests to ensure that the identified issues have been resolved.

Wireshark serves as a valuable tool in the troubleshooting process by providing detailed insights into network traffic, helping network administrators identify and resolve connectivity issues effectively.

Outcomes:

CO 1: Realize that premise of vulnerability analysis and penetration testing (VAPT).

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Learned the different types and uses of multiple vulnerability tools and also setting them up to understand the different uses and functions of them.

Signature of faculty in charge with date

References:

1. https://www.simplilearn.com/top-cyber-security-tools-article
2. https://www.everand.com/book/628472835/Penetration-Testing-of-Computer-Networks-Using-BurpSuite-and-Various-Penetration-Testing-Tools
3. https://www.dummies.com/article/technology/cybersecurity/penetration-testing-with-burp-suite-and-wireshark-to-uncover-vulnerabilities-270960/