Experiment - 1

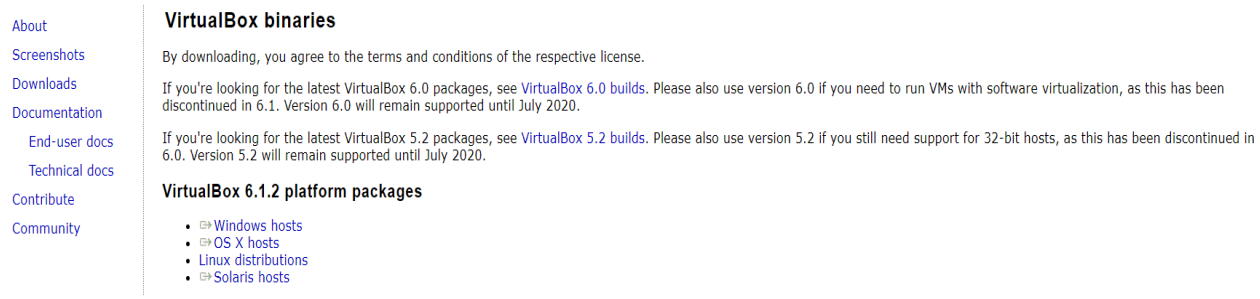
Aim: Installation of virtual Box and Kali Linux on the virtual Machine.

Introduction: -

Virtual Machine abstracts the hardware of our personal computers such as CPU, disk drives, memory, NIC (Network Interface Card), etc, into many different execution environments as per our requirements, hence giving us a feeling that each execution environment is a single computer. For example, VirtualBox.
We can create a virtual machine for several reasons, all of which are fundamentally related to the ability to share the same basic hardware yet can also support different execution environments, i.e., different operating systems simultaneously.

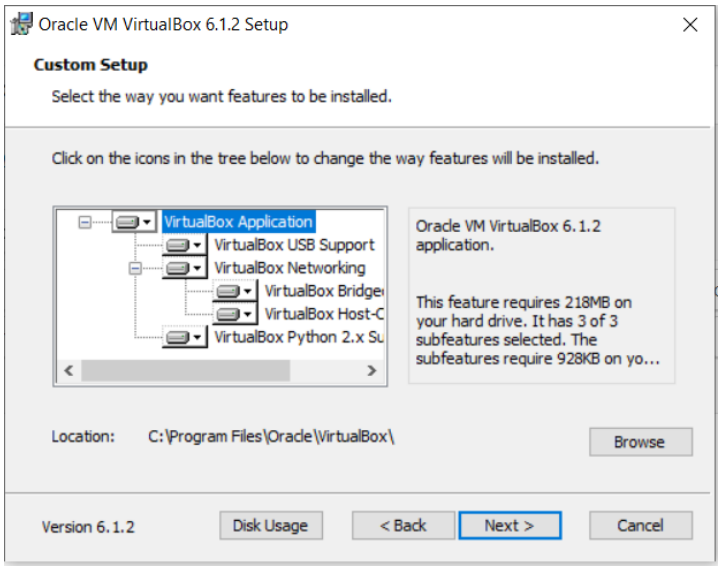
Installation Of Virtual Box: -

Step 1: To download VirtualBox, go to the official site virtualbox.org and download the latest version for windows.

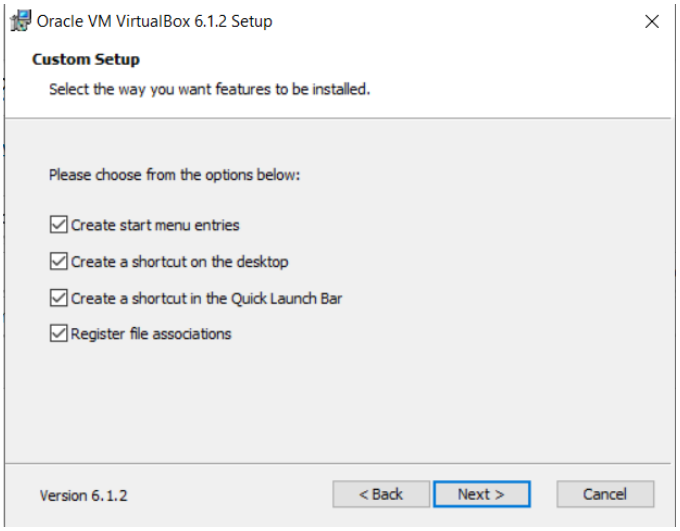


Step 2: Getting Started

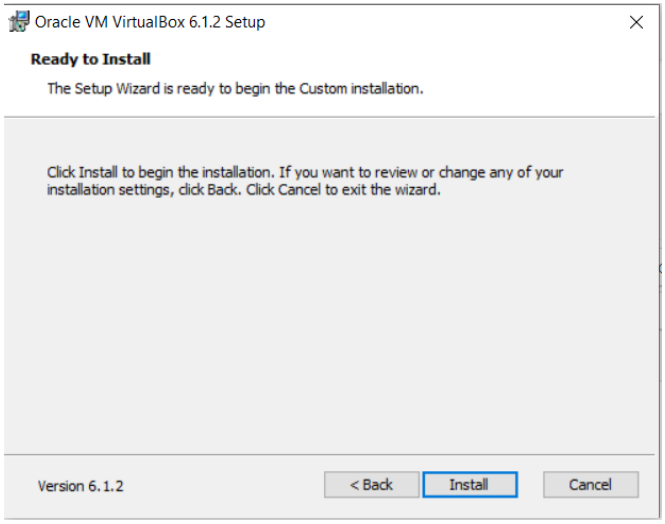
Step 3: Select Installation Location



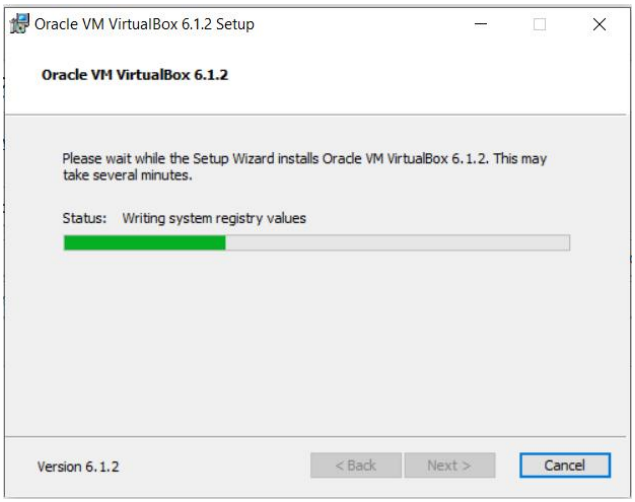
Step 4: - Creating Entries and Shortcuts



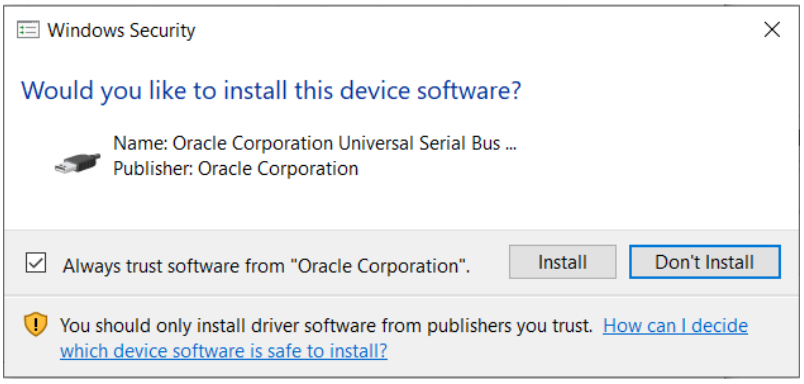
Step 5: Ready to Install



Step 6: Installing Files and packages



Step 7: Installing Certificates



Step 8: Finished Installation

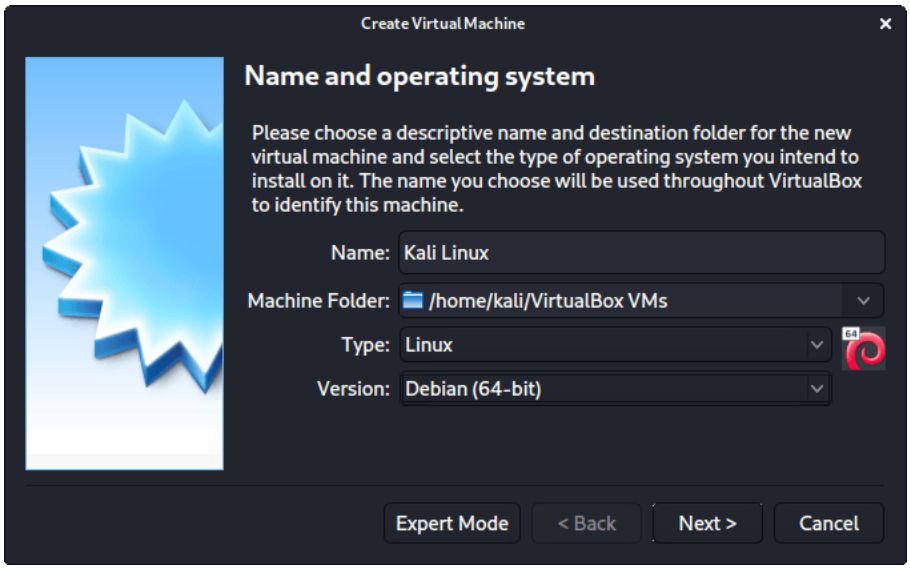
Introduction to Kali Linux: -

Kali Linux is a Debian-based Linux distribution featuring over six hundred preinstalled penetration testing programs. Installing Kali as a virtual machine on a type 2 hypervisor such as VirtualBox provides the isolation and rollback capabilities necessary for advanced security testing.

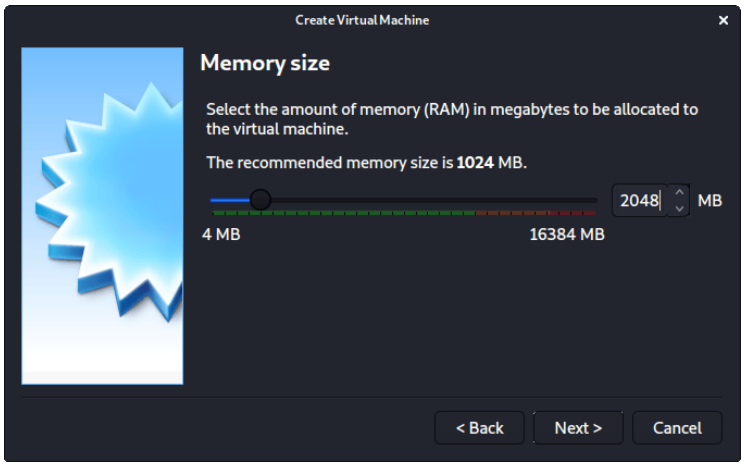
Steps of Installation: -

Step 1: Upon starting up VirtualBox, select "New" (Machine -> New).

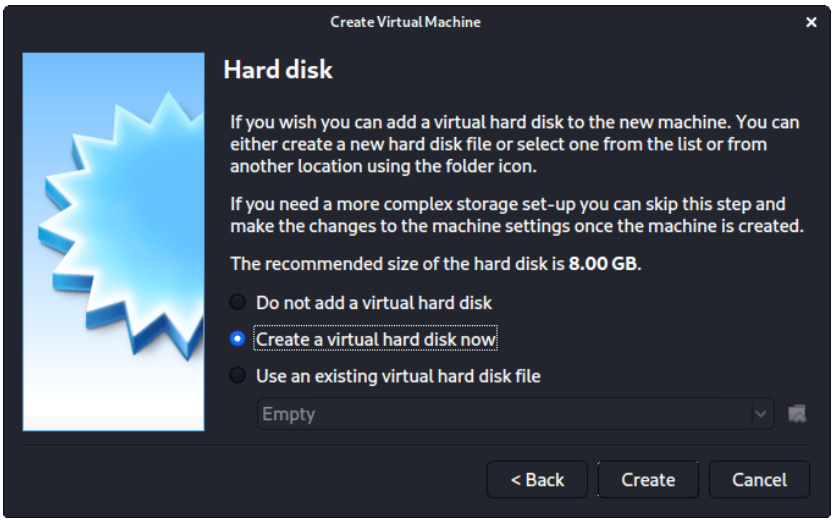
Step 2: Write the name of the virtual machine and select it to Debian based 64bit Linux architecture.



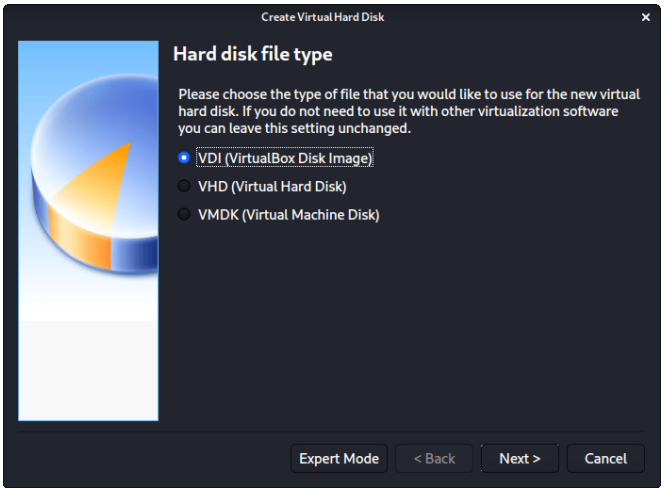
Step 3: Allot the size of RAM memory you want to allocate to the Virtual Machine of Kali Linux. (2048 MB is recommended for normal usage).



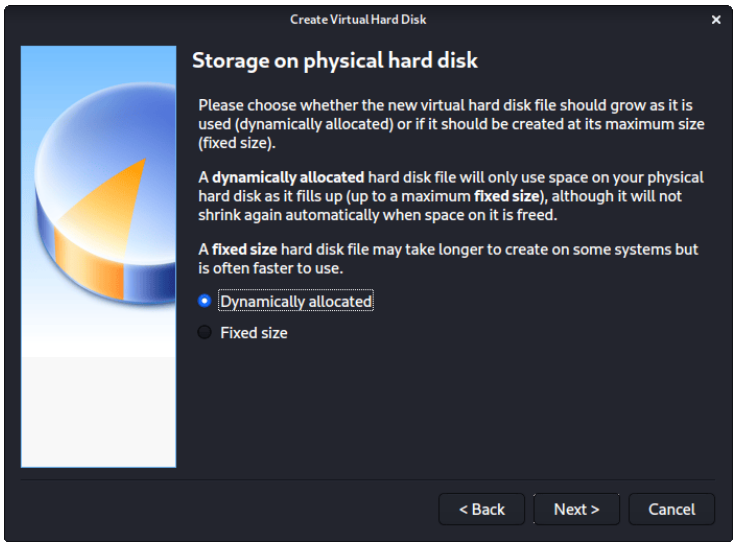
Step 4: Select the option to create a virtual hard disk now and then click on the create button.



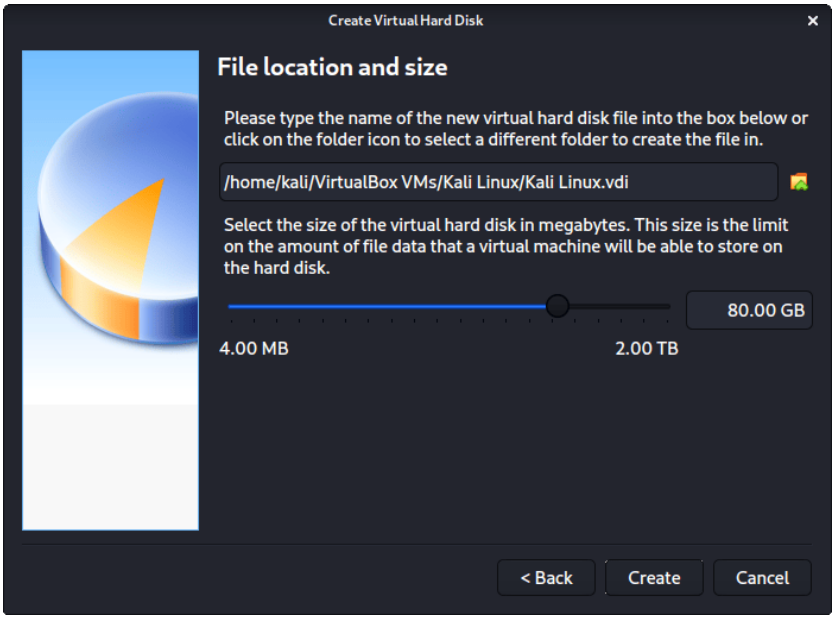
Step 5: Now, Select the Hard Disk File type for Virtual Hard Disk. (VDI is recommended for daily purposes)



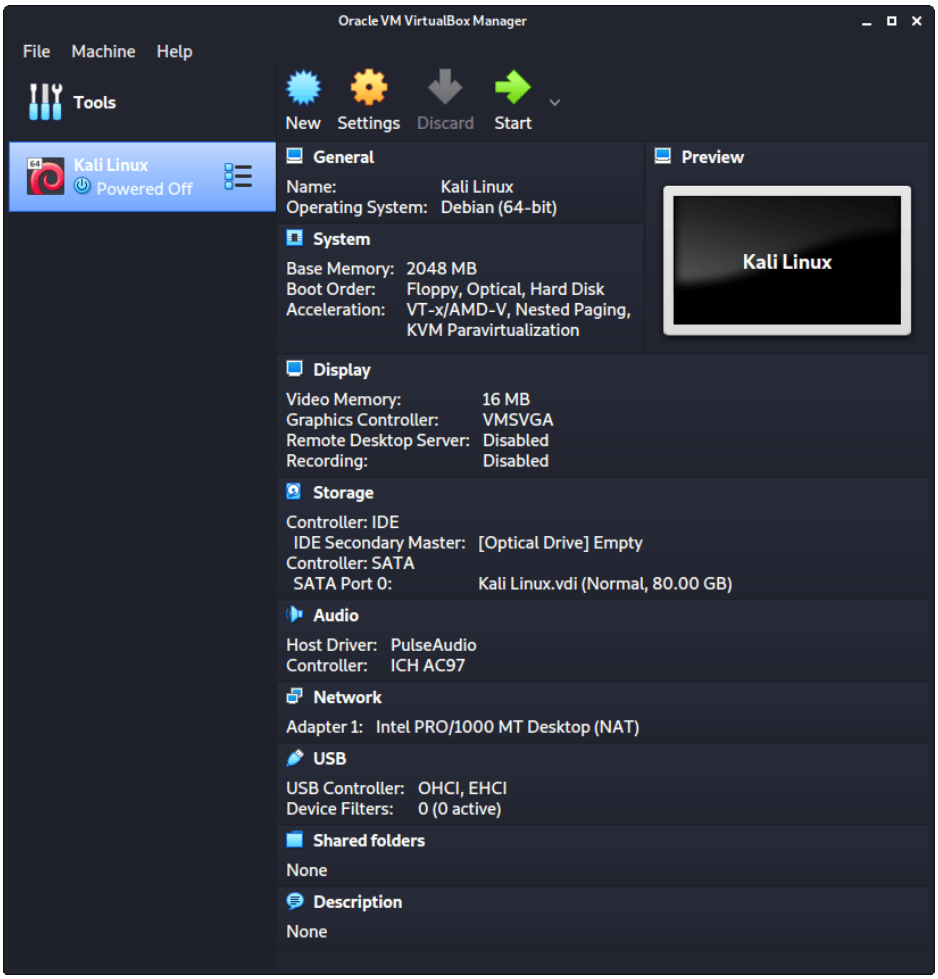
Step 6: Now select the type of Physical Hard Disk Storage. (Dynamically Allocated is recommended for general purposes)



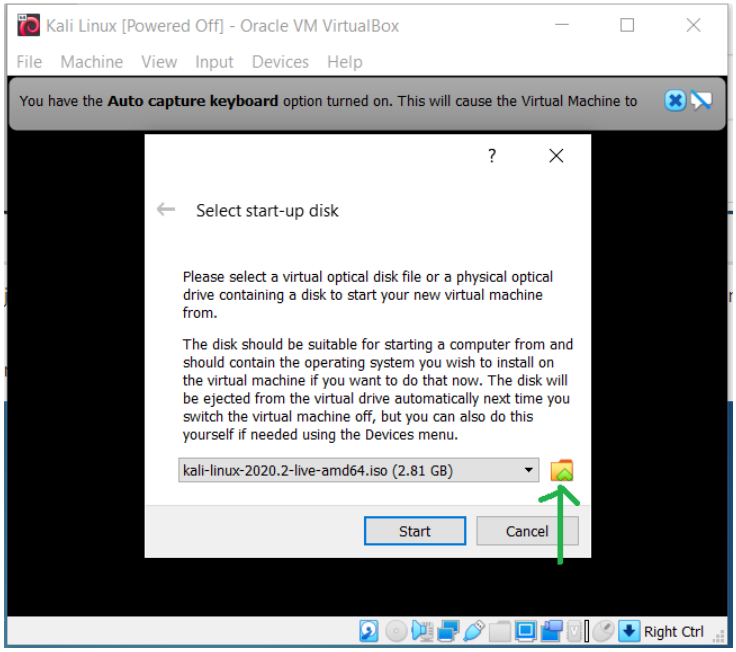
Step 7: Select the size of your virtual hard disk and also the location where you want to save your machine and its files.



Step 8: As soon as the processing is completed click on the Virtual machine name on the left panel and click on the start button from the top.



Step 9: Now click on the Browse icon located just above the cancel button and select your downloaded Kali Linux ISO file and then click on the start button.



Step 10: This will boot our virtual machine from the chosen Kali Linux ISO file. Select the Graphical Install button and hit enter.

Experiment – 2

Aim: Execute the following tools and commands:

NMAP

BURP-SUIT

NIKTO

META SPOILT

AIRCRACK NG

JOHN THE RIPPER

Introduction: -

These tools help to perform a comprehensive security assessment on a target system using various penetration testing method to identify vulnerabilities, test network security, and evaluate password strength.

Nmap - Network scanning and enumeration.

Burp Suite - Web application security testing.

Nikto - Web server vulnerability scanner.

Metasploit - Exploitation framework.

Aircrack-ng - Wireless network penetration testing.

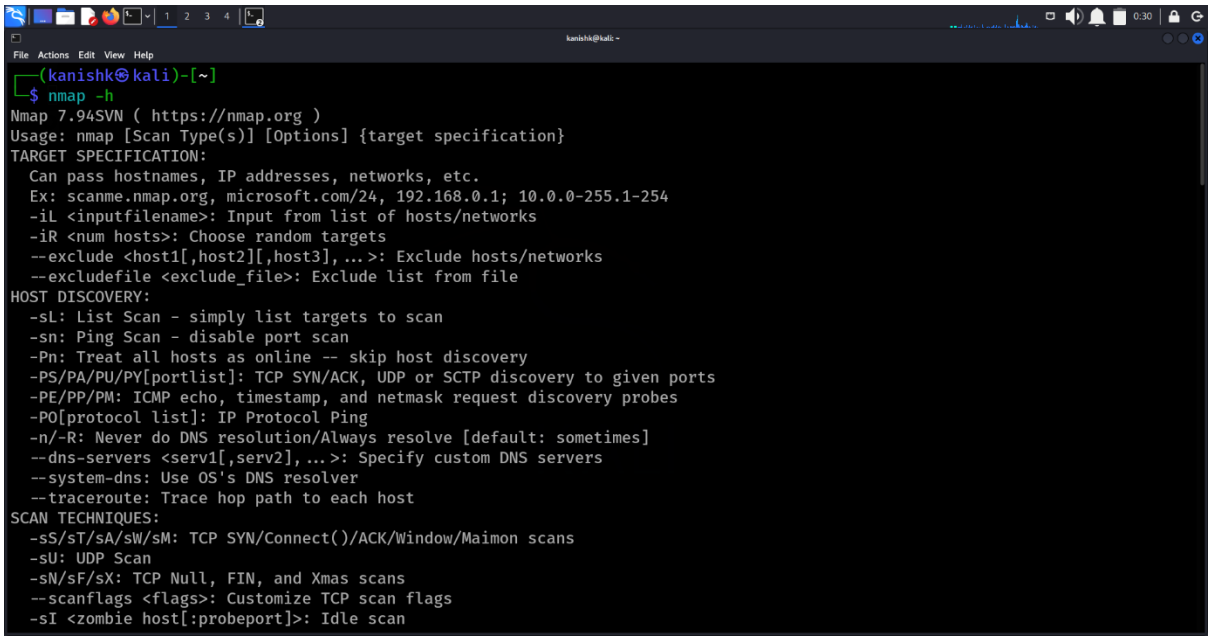
John the Ripper - Password cracking.

1) NETWORK SCANNING AND ENUMERATION WITH NMAP

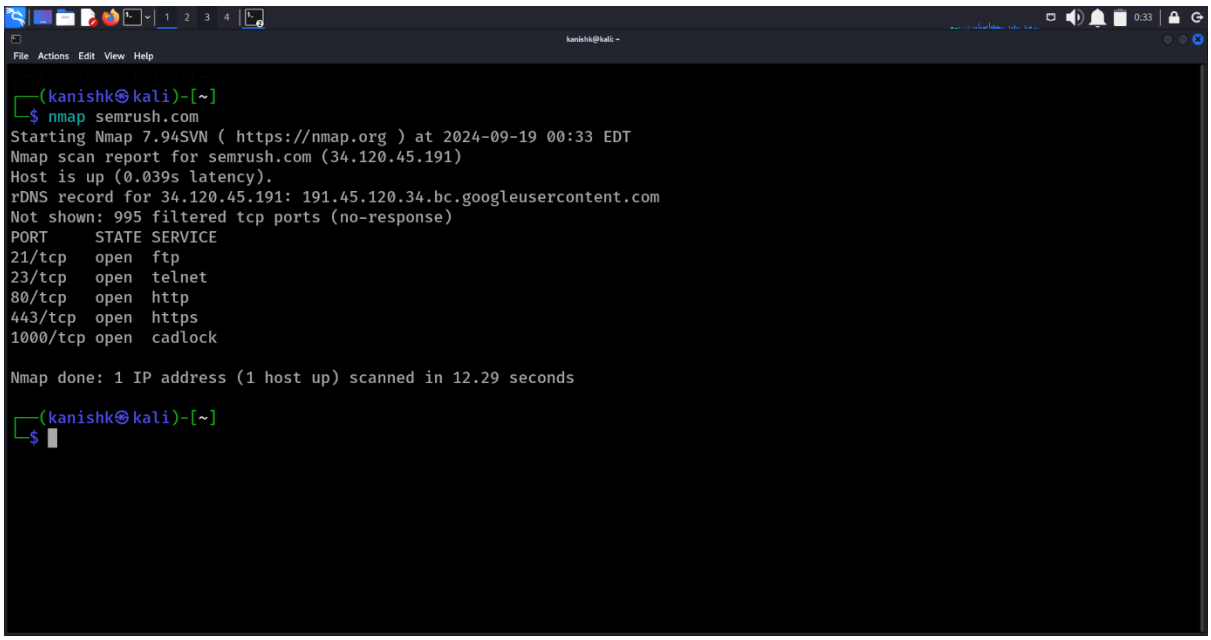
Objective: Identify open ports, services, and potential vulnerabilities on the target system.

Target: https://semrush.com

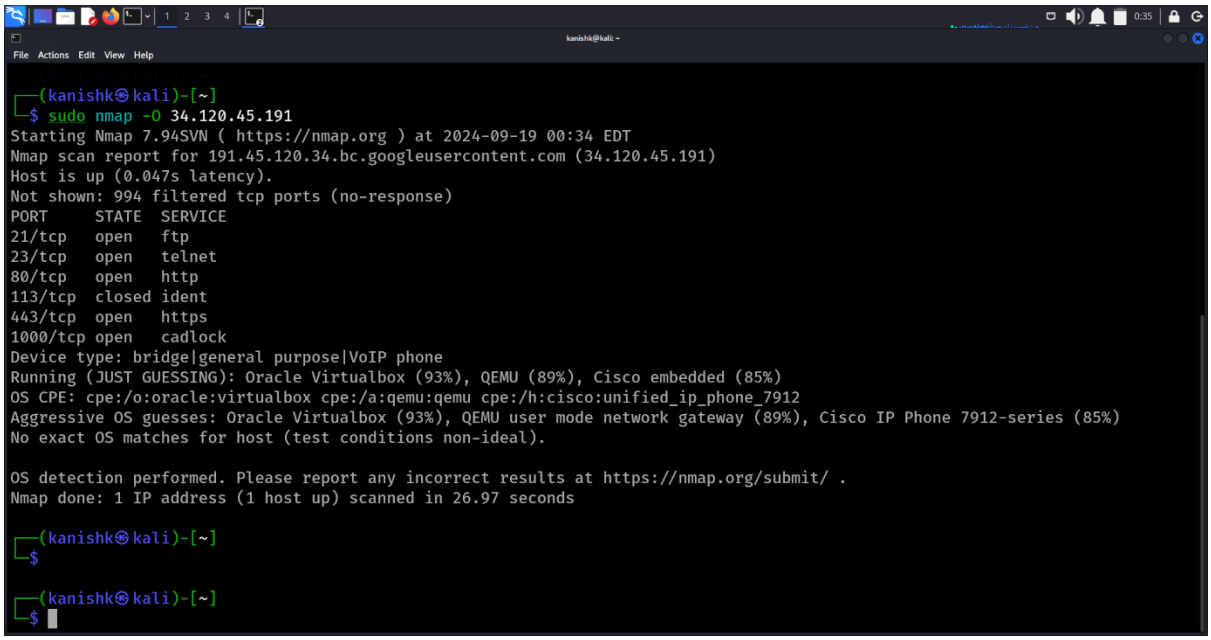
1. Open linux terminal, run **nmap -h** to view all help command prompt of nmap.



2. Now run **nmap semrush.com**
   This will scan for open ports on the domain.

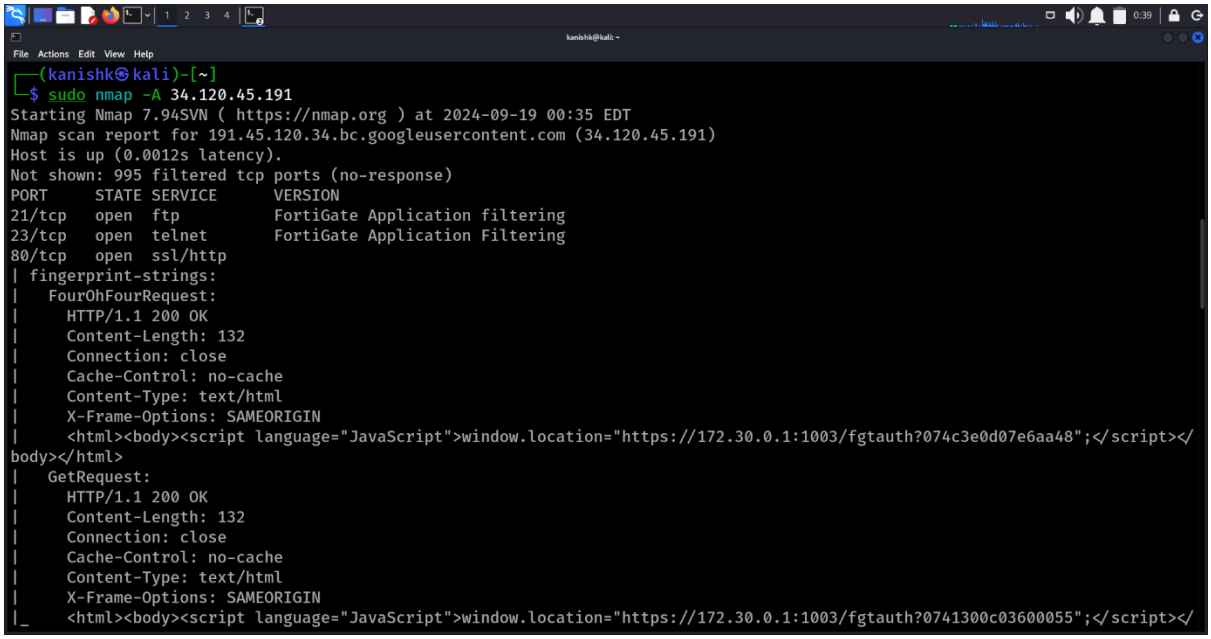


3. Run **sudo nmap -o 34.120.45.191** (The IP address of server which we find from first scan)
   This command Scan for OS Detection

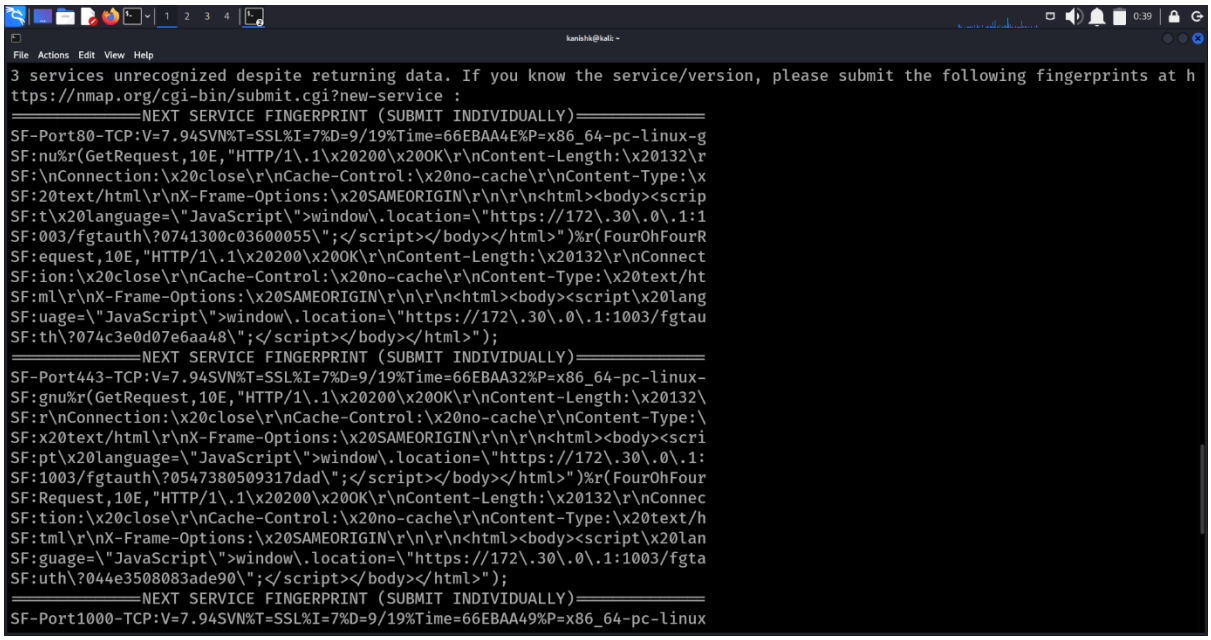


4. Run **sudo nmap -A 45**

This is for full TCP scan with Service Detection and OS Detection
This combines several features for a comprehensive scan:



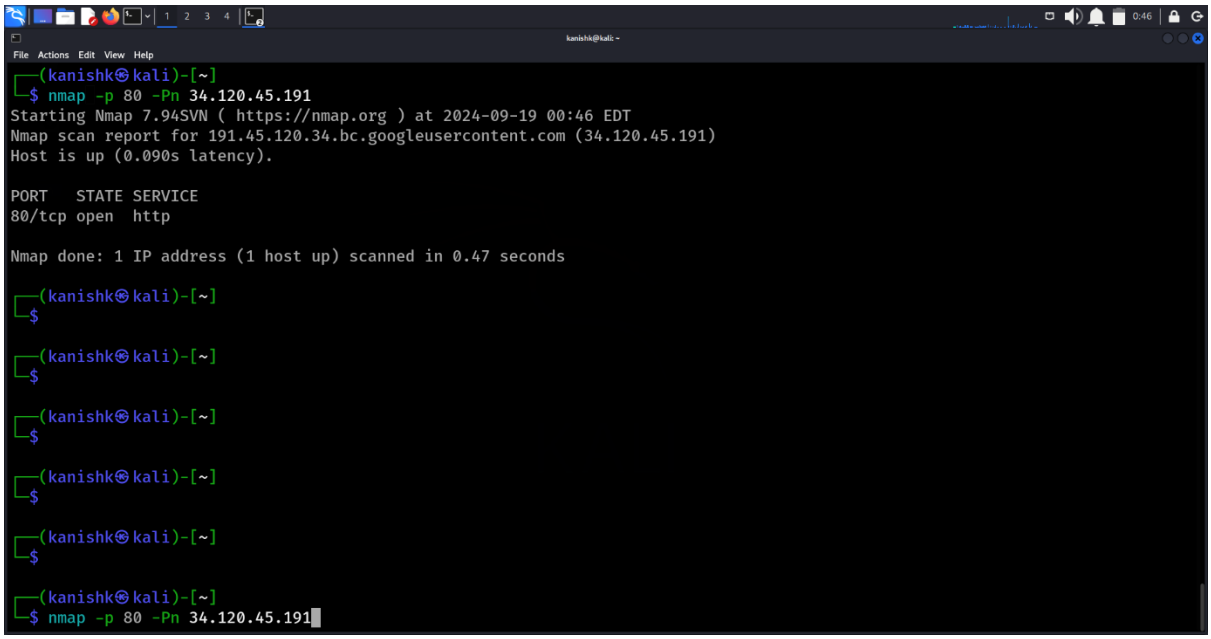
```
File  Actions  Edit  View  Help
┌──(kanishk㉿kali)-[~]
└─$ sudo nmap -A 34.120.45.191
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-19 00:35 EDT
Nmap scan report for 191.45.120.34.bc.googleusercontent.com (34.120.45.191)
Host is up (0.0012s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE SERVICE      VERSION
21/tcp   open  ftp          FortiGate Application filtering
23/tcp   open  telnet       FortiGate Application Filtering
80/tcp   open  ssl/http
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 200 OK
|     Content-Length: 132
|     Connection: close
|     Cache-Control: no-cache
|     Content-Type: text/html
|     X-Frame-Options: SAMEORIGIN
|     <html><body><script language="JavaScript">window.location="https://172.30.0.1:1003/fgtauth?074c3e0d07e6aa48";</script></
body></html>
|   GetRequest:
|     HTTP/1.1 200 OK
|     Content-Length: 132
|     Connection: close
|     Cache-Control: no-cache
|     Content-Type: text/html
|     X-Frame-Options: SAMEORIGIN
|_    <html><body><script language="JavaScript">window.location="https://172.30.0.1:1003/fgtauth?0741300c03600055";</script></
```



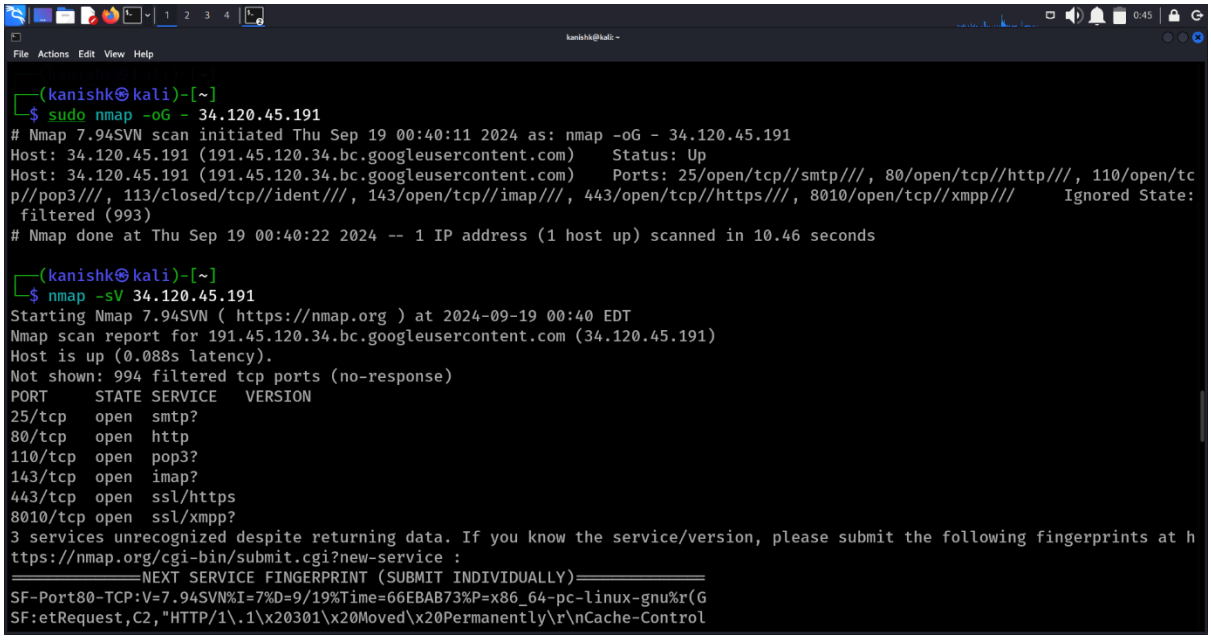
```
File  Actions  Edit  View  Help
3 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at h
ttps://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port80-TCP:V=7.94SVN%T=SSL%I=7%D=9/19%Time=66EBAA4E%P=x86_64-pc-linux-g
SF:nu%r(GetRequest,10E,"HTTP/1\.1\x20200\x20OK\r\nContent-Length:\x20132\r
SF:\nConnection:\x20close\r\nCache-Control:\x20no-cache\r\nContent-Type:\x
SF:20text/html\r\nX-Frame-Options:\x20SAMEORIGIN\r\n\r\n<html><body><scrip
SF:t\x20language=\"JavaScript\">window\.location=\"https://172\.30\.0\.1:1
SF:003/fgtauth\?0741300c03600055\";</script></body></html>")%r(FourOhFourR
SF:equest,10E,"HTTP/1\.1\x20200\x20OK\r\nContent-Length:\x20132\r\nConnect
SF:ion:\x20close\r\nCache-Control:\x20no-cache\r\nContent-Type:\x20text/ht
SF:ml\r\nX-Frame-Options:\x20SAMEORIGIN\r\n\r\n<html><body><script\x20lang
SF:uage=\"JavaScript\">window\.location=\"https://172\.30\.0\.1:1003/fgtau
SF:th\?074c3e0d07e6aa48\";</script></body></html>");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port443-TCP:V=7.94SVN%T=SSL%I=7%D=9/19%Time=66EBAA32%P=x86_64-pc-linux-
SF:gnu%r(GetRequest,10E,"HTTP/1\.1\x20200\x20OK\r\nContent-Length:\x20132\
SF:r\nConnection:\x20close\r\nCache-Control:\x20no-cache\r\nContent-Type:\
SF:x20text/html\r\nX-Frame-Options:\x20SAMEORIGIN\r\n\r\n<html><body><scri
SF:pt\x20language=\"JavaScript\">window\.location=\"https://172\.30\.0\.1:
SF:1003/fgtauth\?0547380509317dad\";</script></body></html>")%r(FourOhFour
SF:Request,10E,"HTTP/1\.1\x20200\x20OK\r\nContent-Length:\x20132\r\nConnec
SF:tion:\x20close\r\nCache-Control:\x20no-cache\r\nContent-Type:\x20text/h
SF:tml\r\nX-Frame-Options:\x20SAMEORIGIN\r\n\r\n<html><body><script\x20lan
SF:guage=\"JavaScript\">window\.location=\"https://172\.30\.0\.1:1003/fgta
SF:uth\?044e3508083ade90\";</script></body></html>");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port1000-TCP:V=7.94SVN%T=SSL%I=7%D=9/19%Time=66EBAA49%P=x86_64-pc-linux
```

5. Run **nmap -p 80 -Pn 34.120.45.219**

   Thess commands scan a specific port and ping them. Here we scan port 80.



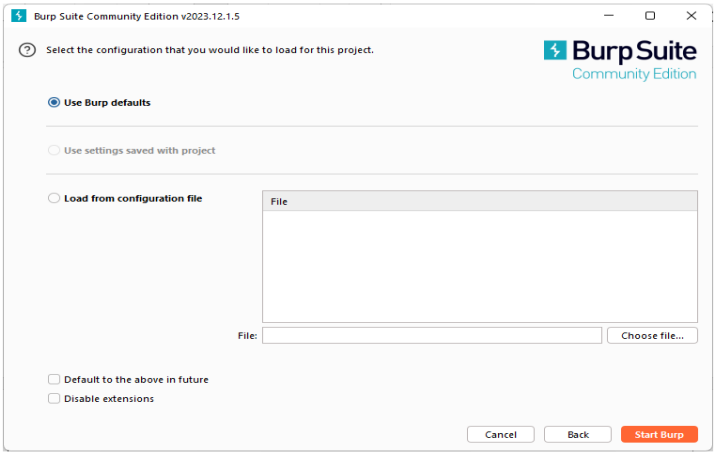
6. Run the **nmap -sV 34.120.45.219** command for service detection on the server



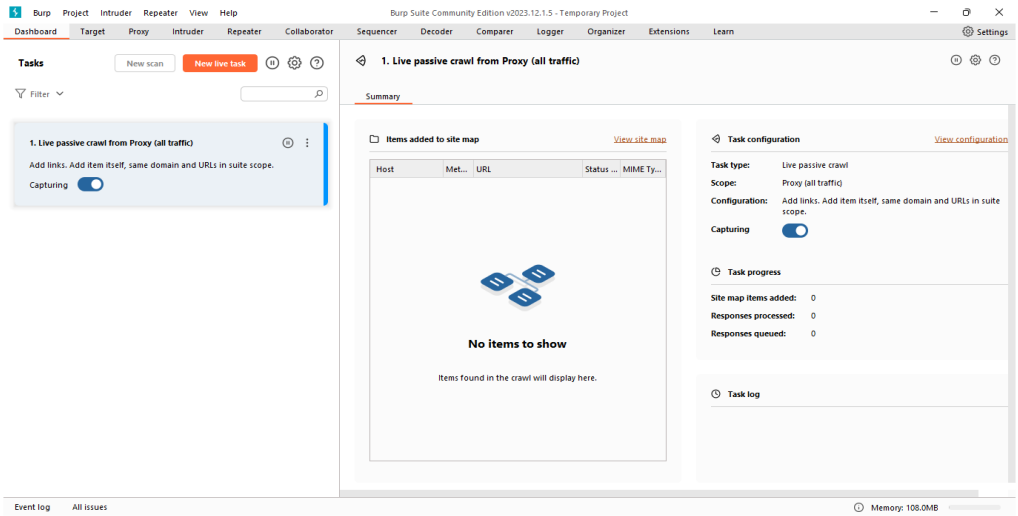
2) WEB APPLICATION SECURITY TESTING WITH BURP SUITE

Objective: Test the target web application for vulnerabilities like XSS, SQL Injection, CSRF, etc.

1. Open Burp suite in Kali Linux. Burp suite is already available with Kali Linux tools. Sometimes, if Burp suite is not available, then please download it and setup.



2. Now click the start button, you were redirected here on Burp suite dashboard.

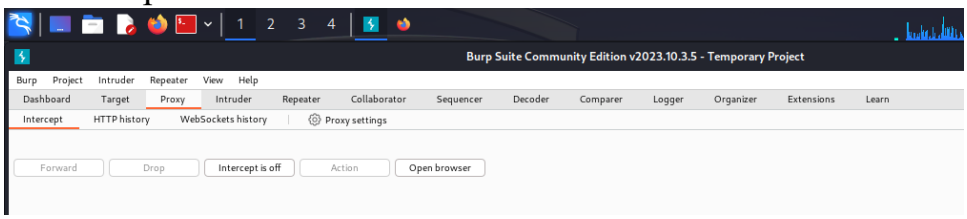


Now we are performing **HTTP traffic intercepting with Burp Proxy.**

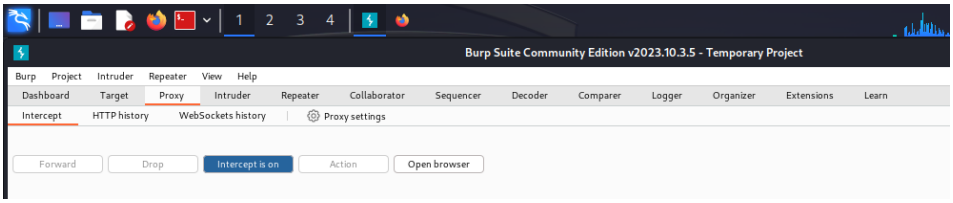
Burp Proxy lets you intercept HTTP requests and responses sent between Burp's browser and the target server. This enables you to study how the website behaves when you perform different actions.

3. Launch **Burp's browser**

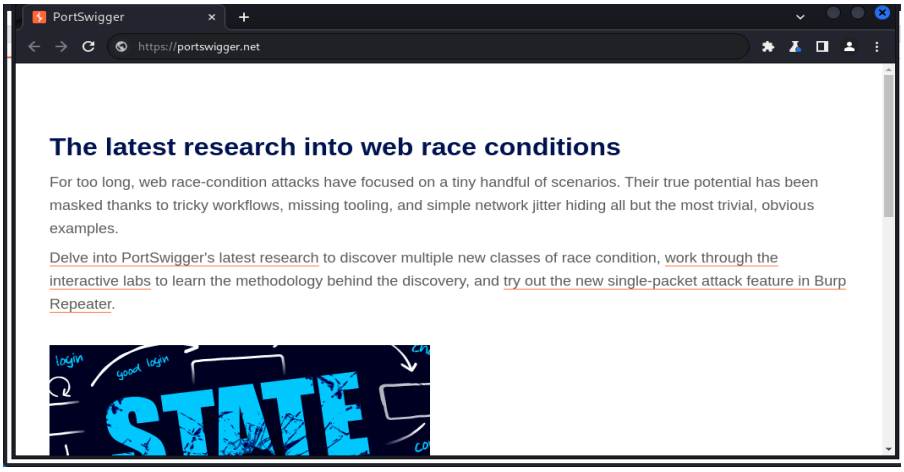
Go to the Proxy > Intercept tab. Set the intercept toggle to Intercept on. Before intercept tab On.



After intercept tab On.

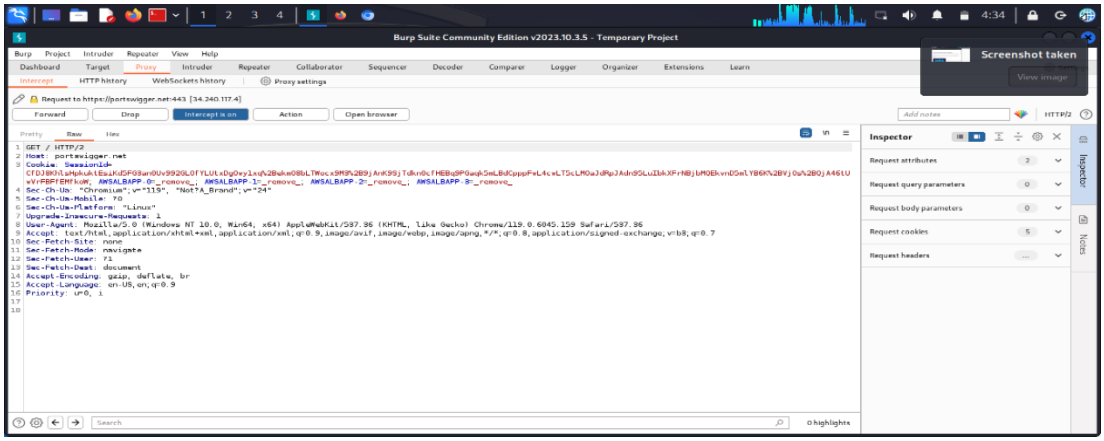


Click **Open Browser**. This launches Burp's browser, which is preconfigured to work with Burp right out of the box.



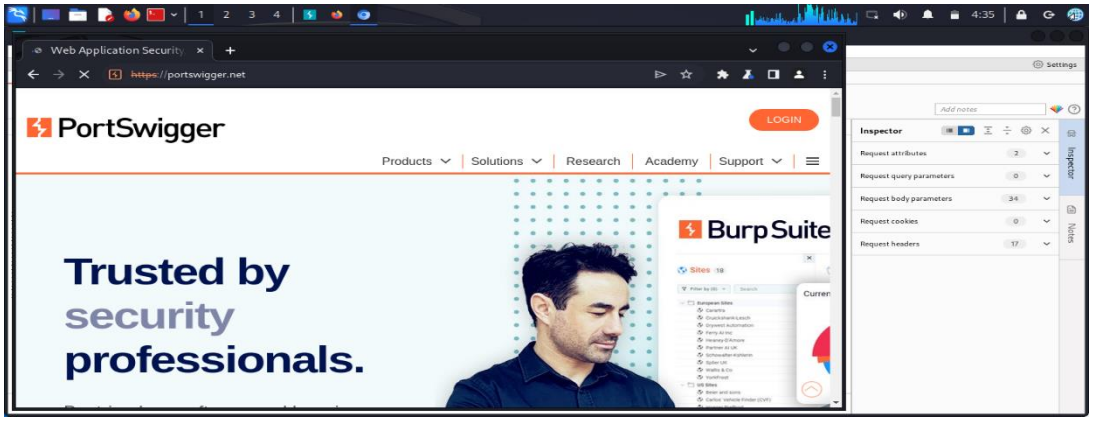
4. Intercept a request

Using Burp's browser, visit **https://portswigger.net** and observe that the site doesn't load. Burp Proxy has intercepted the HTTP request that was issued by the browser before it could reach the server.



5. Forward the request

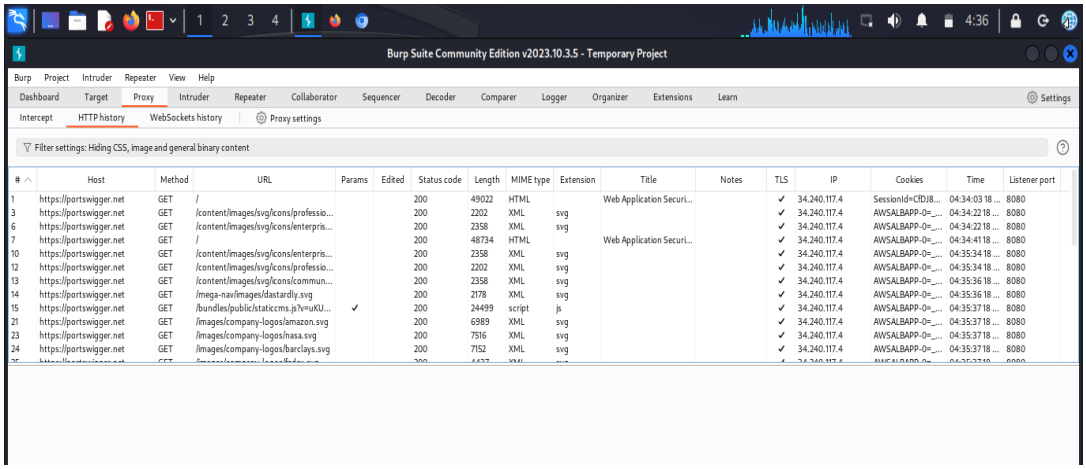
Click the **Forward** button to send the intercepted request. Click **Forward** again to send any subsequent requests that are intercepted, until the page loads in Burp's browser. The **Forward** button sends all the selected requests.



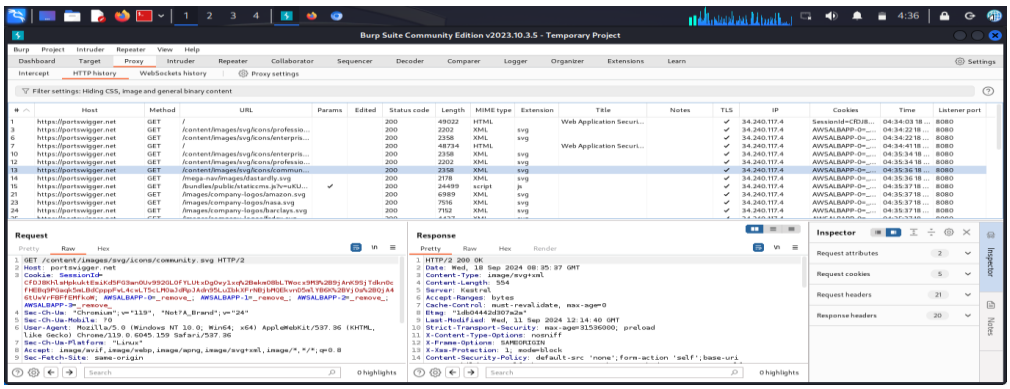
6. Switch off interception

7. View the HTTP history

In Burp, go to the Proxy > HTTP history tab. Here, you can see the history of all HTTP traffic that has passed through Burp Proxy, even while intercept was switched off.



Click on any entry in the history to view the raw HTTP request, along with the corresponding response from the server.

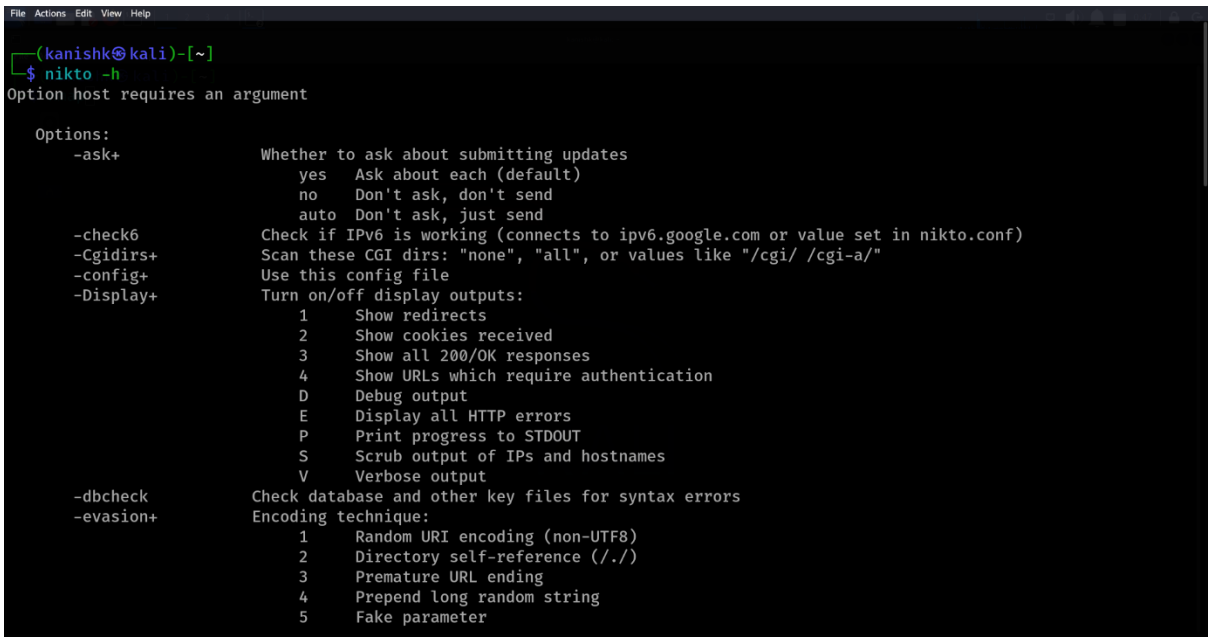


3) WEB SERVER VULNERABILITY SCAN WITH NIKTO

Objective: Scan the web server for known vulnerabilities, misconfigurations, or outdated software.

Target: https://semrush.com

1. Open linux terminal, run **nikto -h** to view all help command prompt of nmap.



2. Run **nikto -h https://semrush.com.**
   This is an basic scan of target.
   -h Specifies the target host or URL.

```
File  Actions  Edit  View  Help

┌──(kanishk㉿kali)-[~]
└─$ nikto -h https://semrush.com
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          34.120.45.191
+ Target Hostname:    semrush.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=*.semrush.com
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA Domain Validation Secure S
erver CA
+ Start Time:         2024-09-19 00:48:01 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx
+ /: Cookie GCLB created without the secure flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Retrieved via header: 1.1 google.
+ /: Uncommon header 'sm-log-id' found, with contents: flb-bf344b6aebf4bfc98cb354065fe37828.
+ /: An alt-svc header was found which is advertising HTTP/3. The endpoint is: ':443'. Nikto cannot test HTTP/3 over QUIC. See
: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/alt-svc
+ Root page / redirects to: https://www.semrush.com/
+ /NGn0zAig.map: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the si
te in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-
content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /database.tar: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /com.war: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /semrushcom.tar: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
```

3. Run **nikto -h https://semrush.com -p 80**

   This specify port and scan that port. Here we scan port number 80 for HTTP but it not completed because of not using full URI.

```
File  Actions  Edit  View  Help

┌──(kanishk㉿kali)-[~]
└─$ nikto -h https://semrush.com -p 80
- Nikto v2.5.0
---------------------------------------------------------------------------
- ERROR: The -port option cannot be used with a full URI
```

4. Run **nikto -h https://semrush.com -Plugins all**

   This perform a Full Scan with Plugins

   Nikto also includes various plugins to check for different types of vulnerabilities.

```
File  Actions  Edit  View  Help

┌──(kanishk㉿kali)-[~]
└─$ nikto -h https://semrush.com -Plugins all
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          34.120.45.191
+ Target Hostname:    semrush.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=*.semrush.com
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA Domain Validation Secure S
erver CA
+ Start Time:         2024-09-19 01:19:37 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx
+ Root page / redirects to: https://www.semrush.com/
```

5. Run **nikto -h https://semrush.com -dbcheck**

   This scan for Known Vulnerabilities

   This can detect potential vulnerabilities associated with the server software:

```
File  Actions  Edit  View  Help

┌──(kanishk㉿kali)-[~]
└─$ nikto -h https://semrush.com -dbcheck
Syntax Check: /var/lib/nikto/databases/db_drupal
        6244 entries
Syntax Check: /var/lib/nikto/databases/db_httpoptions
        12 entries
Syntax Check: /var/lib/nikto/databases/db_server_msgs
        259 entries
Syntax Check: /var/lib/nikto/databases/db_variables
        38 entries
Syntax Check: /var/lib/nikto/databases/db_404_strings
        39 entries
Syntax Check: /var/lib/nikto/databases/db_domino
        274 entries
Syntax Check: /var/lib/nikto/databases/db_realms
        170 entries
Syntax Check: /var/lib/nikto/databases/db_dictionary
        1825 entries
Syntax Check: /var/lib/nikto/databases/db_multiple_index
        35 entries
Syntax Check: /var/lib/nikto/databases/db_content_search
        20 entries
Syntax Check: /var/lib/nikto/databases/db_tests
        6954 entries
Syntax Check: /var/lib/nikto/databases/db_embedded
        16 entries
Syntax Check: /var/lib/nikto/databases/db_outdated
        1256 entries
Syntax Check: /var/lib/nikto/databases/db_parked_strings
```

6. Run **nikto -h https://semrush.com -T 3**

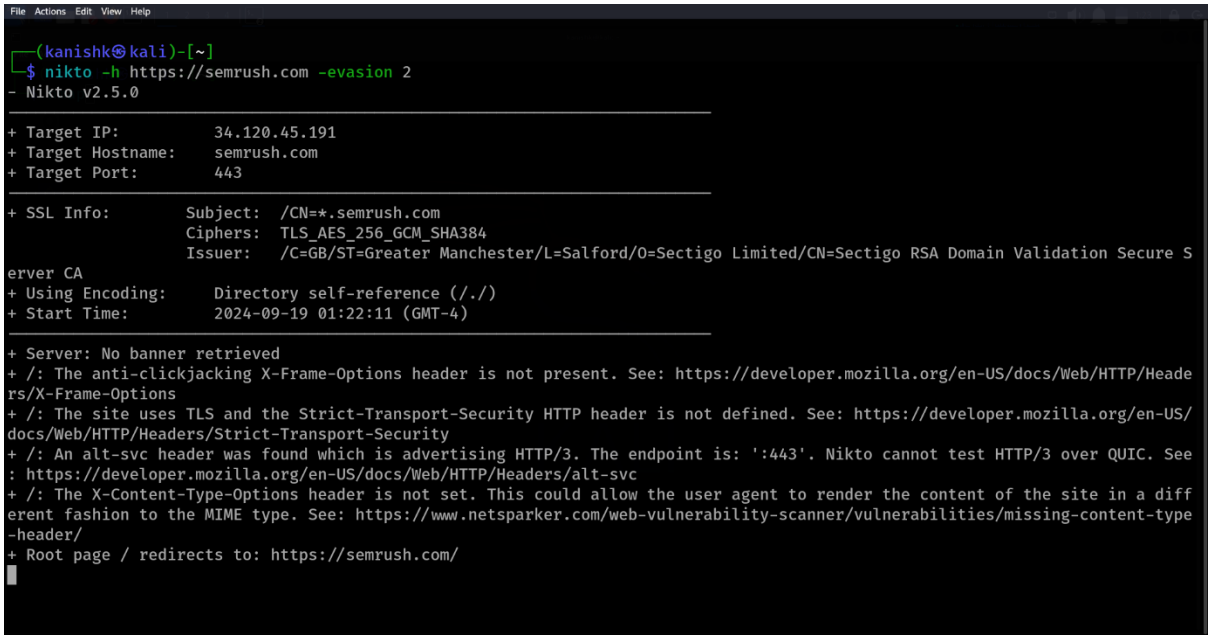
   This Evade IDS/IPS with Randomized Timing.

   The -T option specifies the timeout between requests.

```
File  Actions  Edit  View  Help

┌──(kanishk㉿kali)-[~]
└─$ nikto -h https://semrush.com -T 3
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          34.120.45.191
+ Target Hostname:    semrush.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=*.semrush.com
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA Domain Validation Secure S
erver CA
+ Start Time:         2024-09-19 01:21:35 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx
+ /: Cookie GCLB created without the secure flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Retrieved via header: 1.1 google.
+ /: Uncommon header 'sm-log-id' found, with contents: flb-73960b6aa6a2b07c86889d4c88c7d1fe.
+ /: An alt-svc header was found which is advertising HTTP/3. The endpoint is: ':443'. Nikto cannot test HTTP/3 over QUIC. See
: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/alt-svc
+ Root page / redirects to: https://www.semrush.com/
```

7. Run **nikto -h https://semrush.com -evasion 2**
   Run a Scan in Stealth Mode (IDS/IPS Evasion)

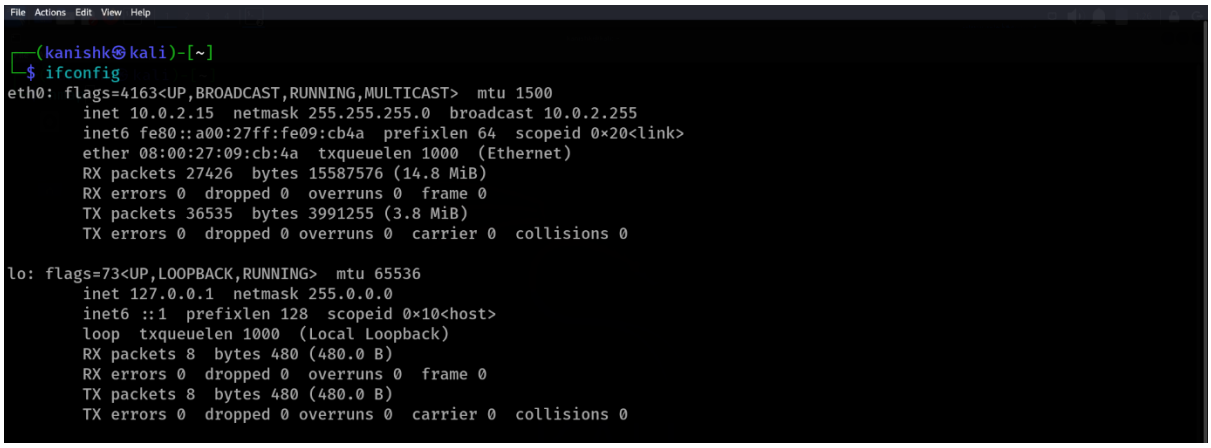


4) Exploitation Using Metasploit Framework

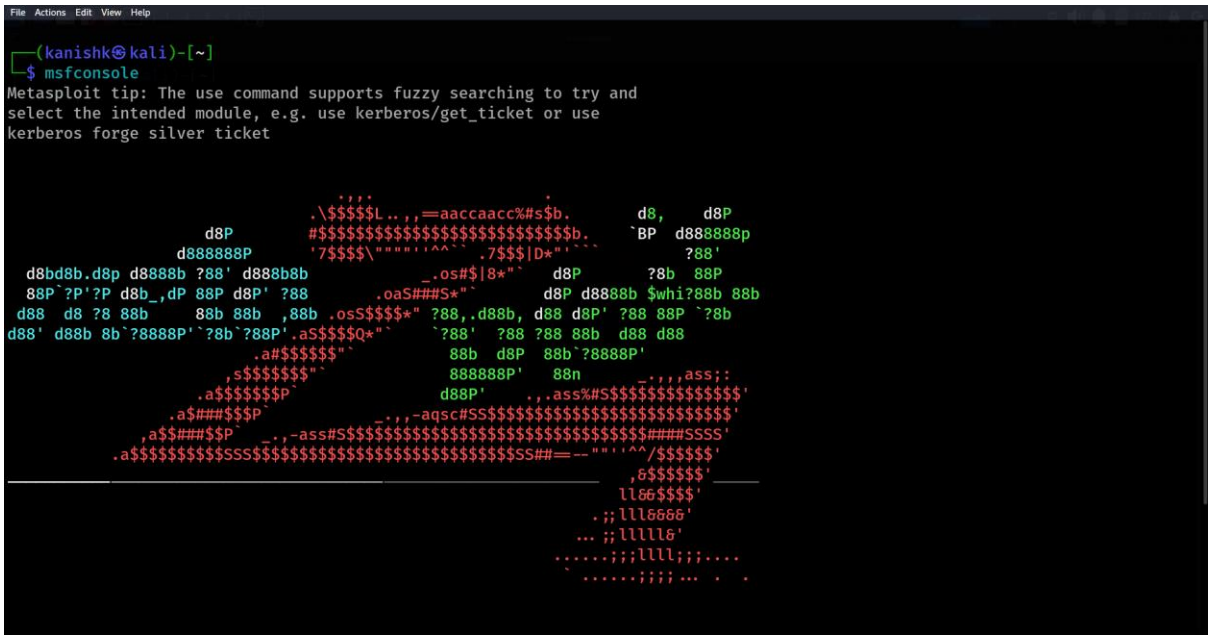
Objective: Exploit vulnerabilities found in the system and gain control or gather further information.

Target: Kali machine

1. Open Kali Linux terminal and find the IP address of machine using **ifconfig** command



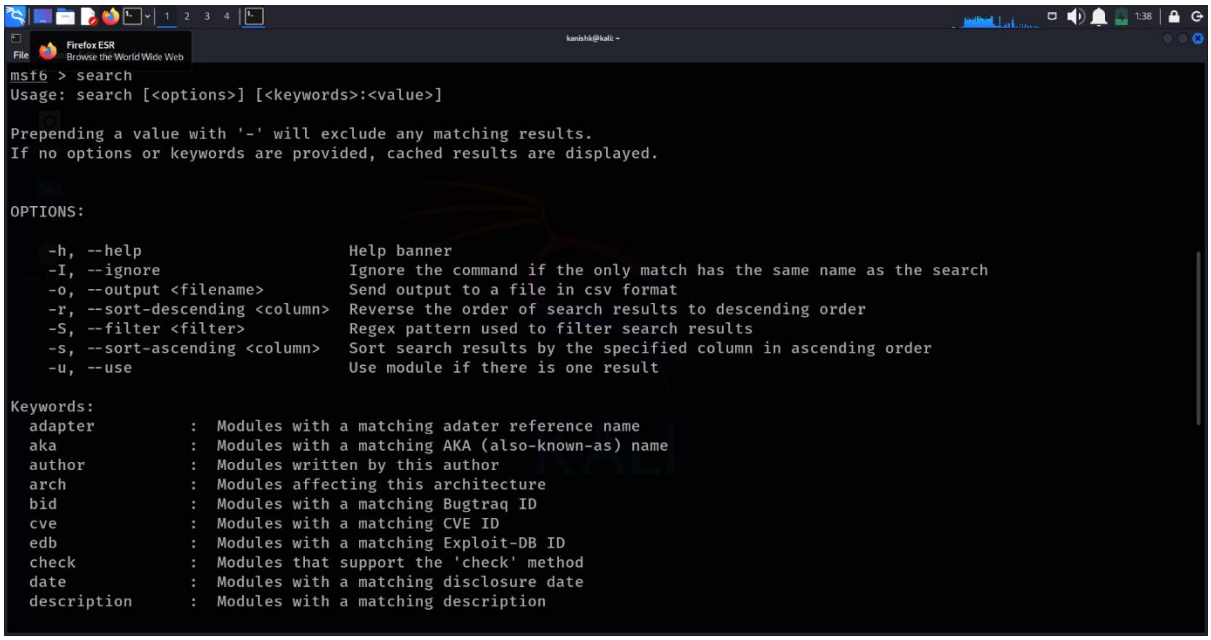
2. Now open the root terminal and run the **msfconsole** command to launch Metasploit.



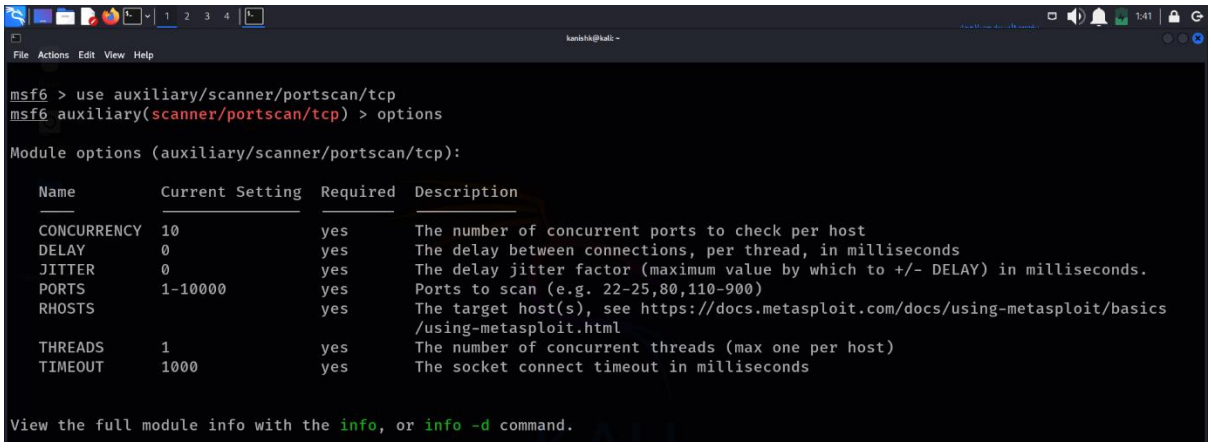
Here, we get separate terminal **msf6** of Metasploit.

3. Now run **search** command

   It Searches for modules based on a keyword. Here we not assign any keyword, hence it search for all possible modules.



4. Now run **use auxiliary/scanner/portscan/tcp** command for tcp port

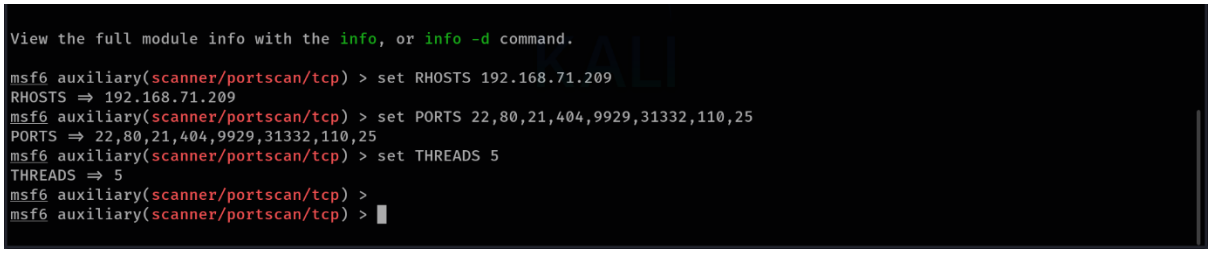


5. Set the RHOSTS, PORTS and THREADS for auxiliary tcp scan by using commands:

   set RHOSTS 192.168.71.209
   set PORTS 22,80,21,404,9929,31332,110,25
   set THREADS 5

   The RHOSTS address is our Kali machine address.

The ports assign for set port are random, we can assign any ports.
Last now, run the all set values on machine by using **run** command.



5) Aircrack: -

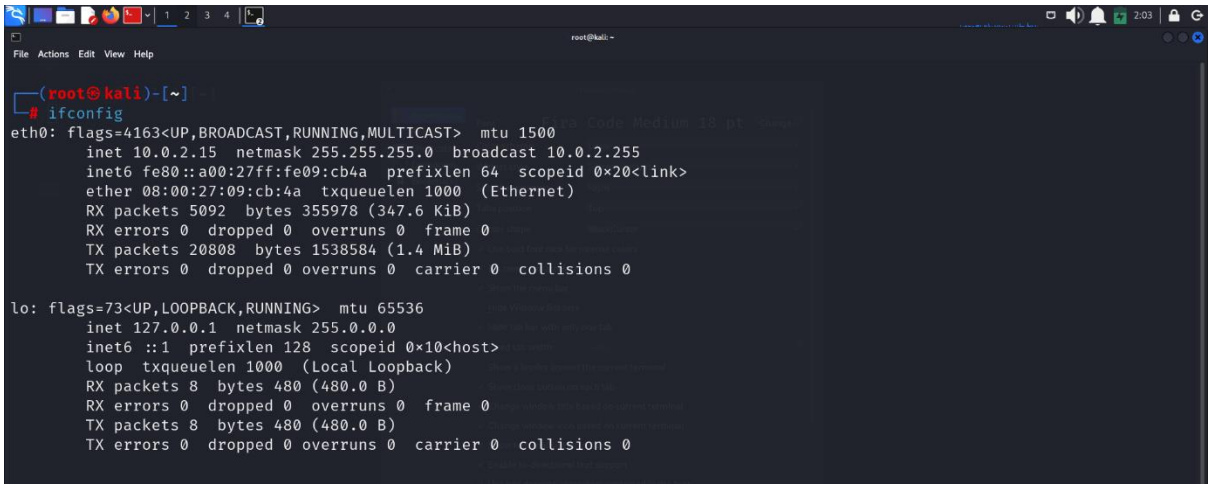
Aircrack-ng is a suite of tools for wireless network security assessment. It focuses on cracking WEP and WPA/WPA2 encryption keys by analyzing and exploiting vulnerabilities in wireless networks.

Objective: Test the security of the wireless network by attempting to crack the Wi-Fi password using packet capture.

1. Open root terminal and enter the airmon-ng command.

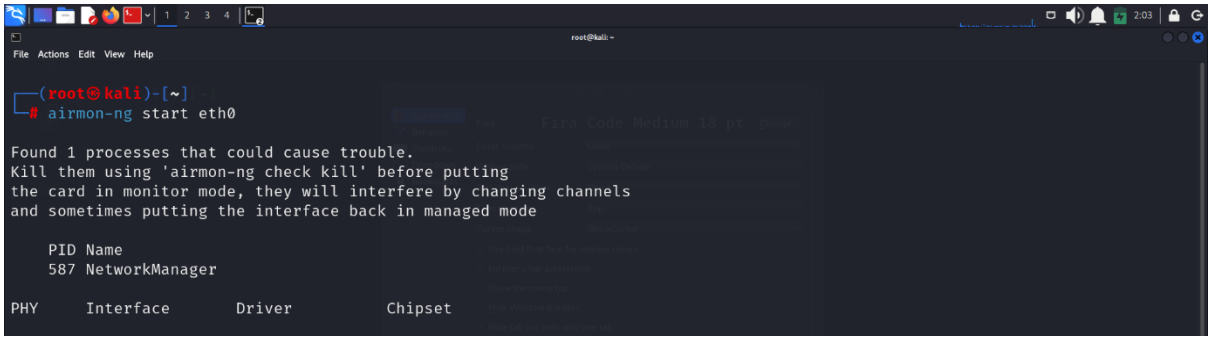
2. Now run the airmon-ng -h to see the help prompt of aircrack -ng.

3. As per help prompt usage, we need interface for executing airmon-ng. Run ifconfig command for finding interface.



We get eth0 and lo interface.

4. Now run the airmon-ng start eth0 command.



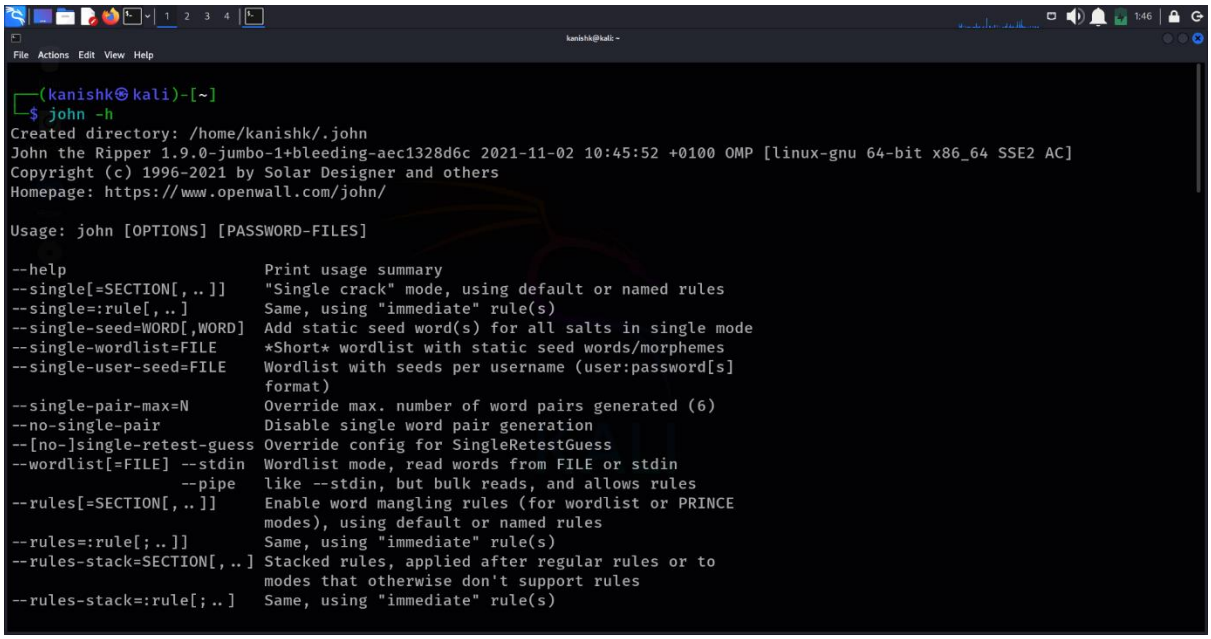
Here we get some trouble.

5. Run airmon-ng check command for finding the trouble.

6) PASSWORD CRACKING WITH JOHN THE RIPPER

Objective: Crack weak passwords from a list of password hashes.

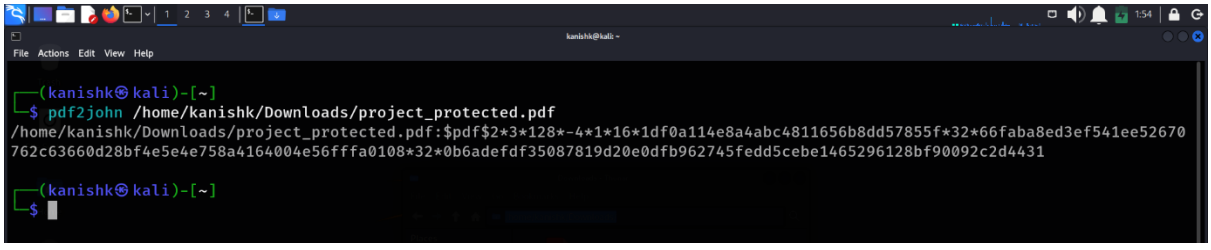
1. Run the **john -h** command to see help prompt of John the ripper.



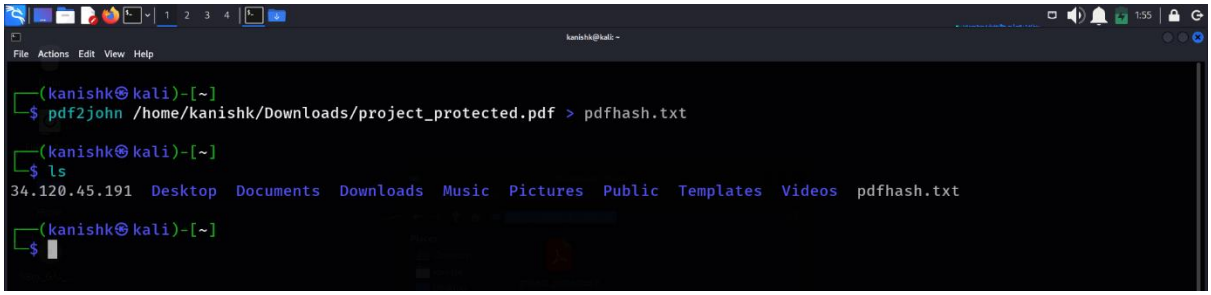
2. Use **pdf2john /(path) to Extract the PDF Hash**

   The pdf2john tool is used to extract the hash from a password-protected PDF file, which John the Ripper can then attempt to crack.

   Replace **/(path)** with the actual path to your password-protected PDF file.



Save the extract hash in pdfhash.txt file using **pdf2 /home/kanishk/Downloads/project_protected.pdf >pdfhash.txt.**

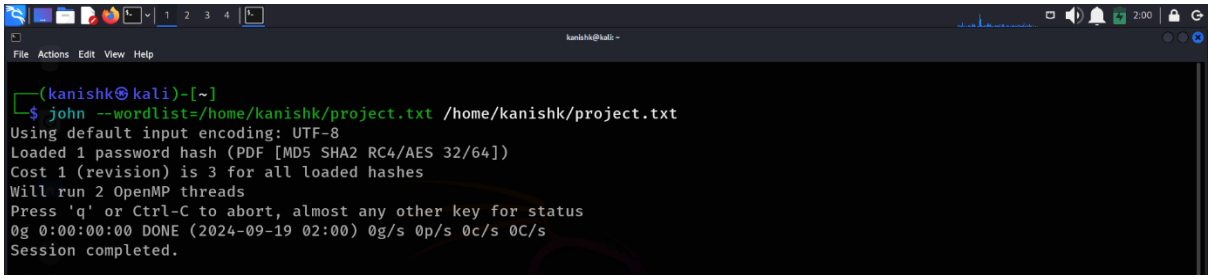


3. Crack the PDF Password with John the Ripper

   Now we have the PDF hash extracted, use John the Ripper to try and crack the password using the wordlist:

   For this use **john –wordlist=/home/kali/kanishk/project.txt /home/kanishk/project.txt** command

   The first path, =**/home/kali/kanishk/project.txt.** direct to the wordlist which contain possible passwords and the second path **/home/kanishk/project.txt** contain the hash file which we extract.



4. **View the Cracked Password**

   Once the cracking process completes (if successful), you can view the cracked password by running:

   john -show project.txt command.