

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: -

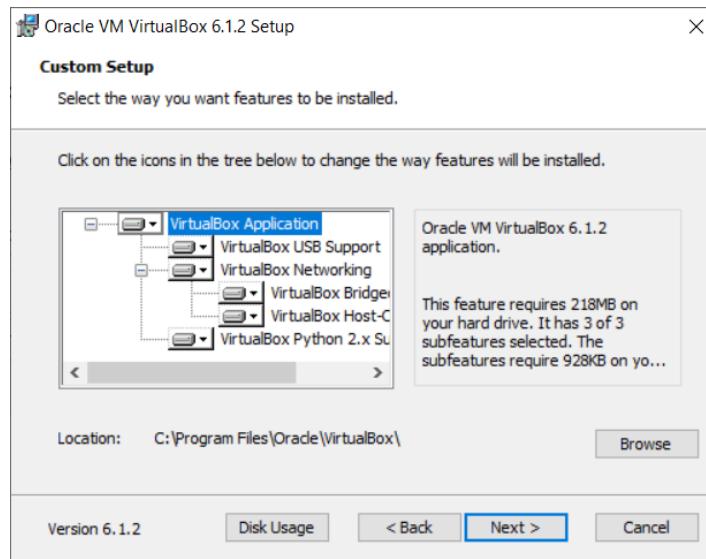
Step 1: To download VirtualBox, go to the official site [virtualbox.org](https://www.virtualbox.org) and download the latest version for windows.

The screenshot shows the official VirtualBox website's download page. On the left, there's a sidebar with links: About, Screenshots, Downloads, Documentation, End-user docs, Technical docs, Contribute, and Community. The main content area has a heading 'VirtualBox binaries'. It says, 'By downloading, you agree to the terms and conditions of the respective license.' Below that, it says, 'If you're looking for the latest VirtualBox 6.0 packages, see [VirtualBox 6.0 builds](#). Please also use version 6.0 if you need to run VMs with software virtualization, as this has been discontinued in 6.1. Version 6.0 will remain supported until July 2020.' Another section below says, 'If you're looking for the latest VirtualBox 5.2 packages, see [VirtualBox 5.2 builds](#). Please also use version 5.2 if you still need support for 32-bit hosts, as this has been discontinued in 6.0. Version 5.2 will remain supported until July 2020.' At the bottom of this section is a heading 'VirtualBox 6.1.2 platform packages' with a bulleted list: 'Windows hosts', 'OS X hosts', 'Linux distributions', and 'Solaris hosts'.

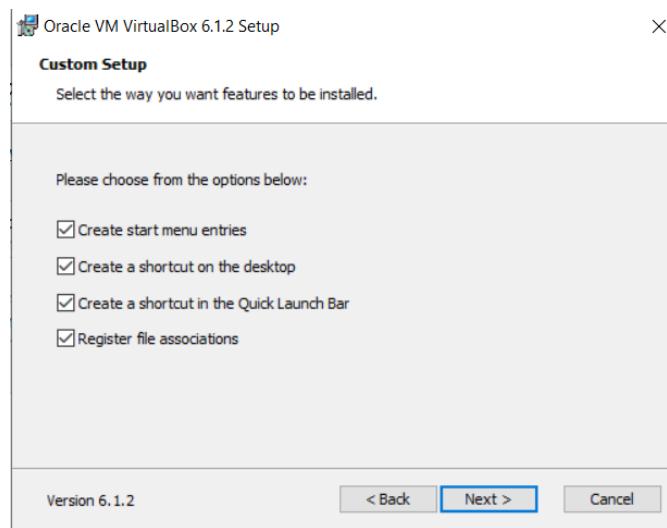
Step 2: Getting Started



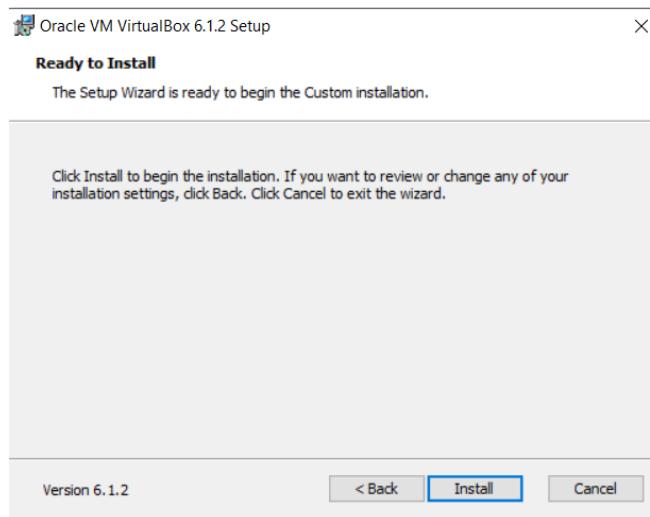
Step 3: Select Installation Location

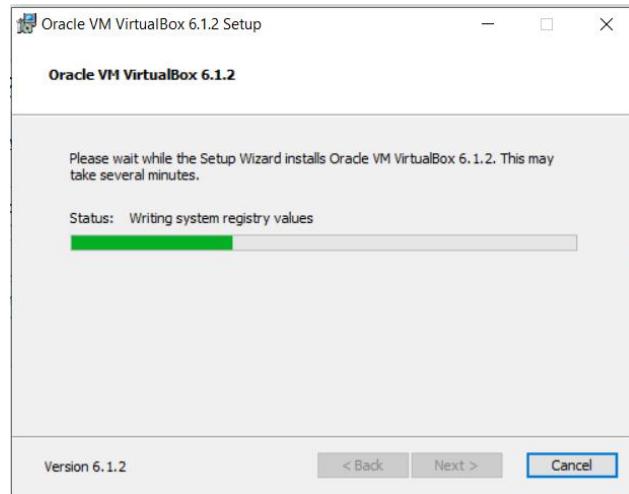
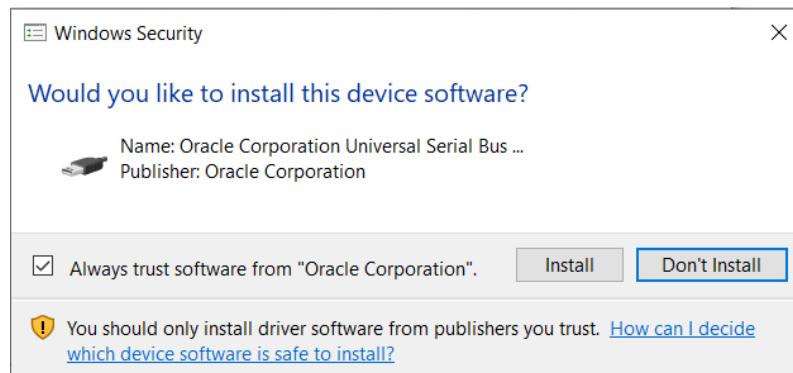


Step 4: - Creating Entries and Shortcuts



Step 5: Ready to Install



Step 6: Installing Files and packages**Step 7:** Installing Certificates

Step 8: Finished Installation



Introduction to Kali Linux: -

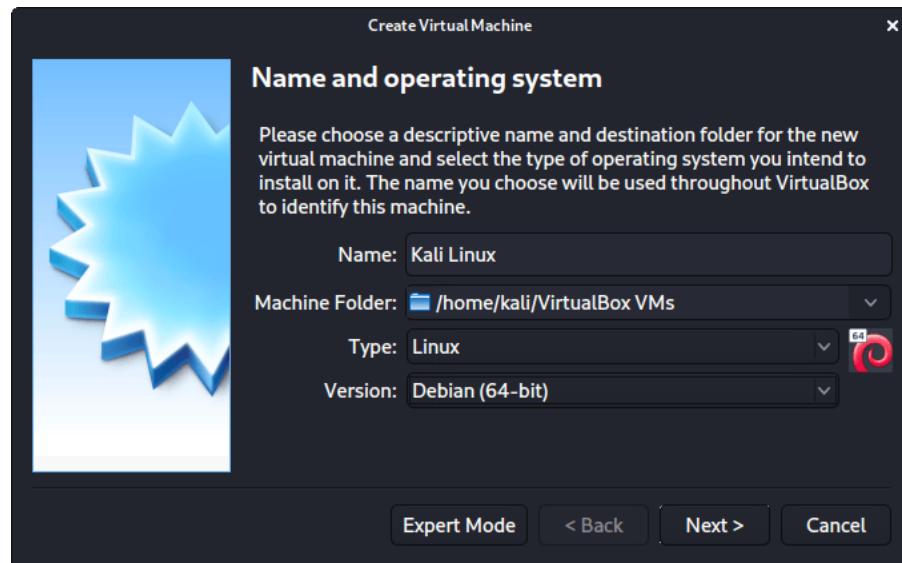
Kali Linux is a Debian-based Linux distribution featuring over six hundred preinstalled penetration testing programs. Installing Kali as a virtual machine on a type 2 hypervisor such as VirtualBox provides the isolation and rollback capabilities necessary for advanced security testing.

Steps of Installation: -

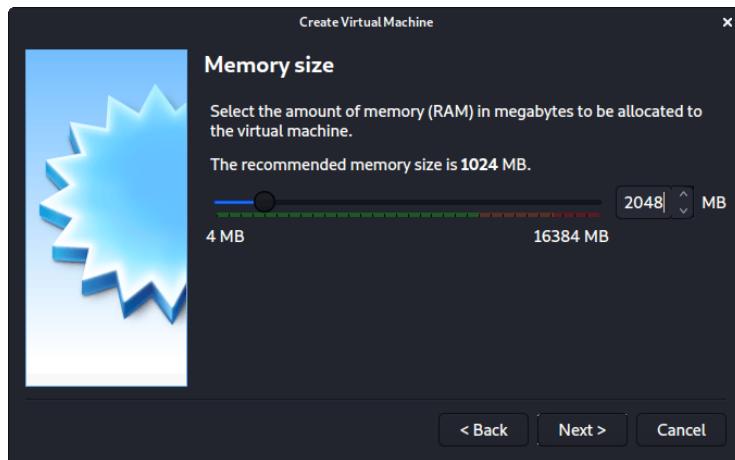
Step 1: Upon starting up VirtualBox, select “New” (Machine -> New).



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



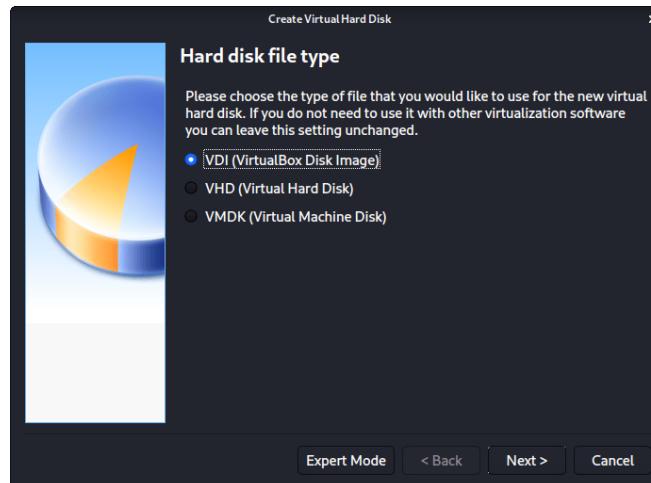
Step 3: Allot the size of RAM memory you want to allocate to the Virtual Machine of Kali Linux. (2048 MB is recommended for normal usage).



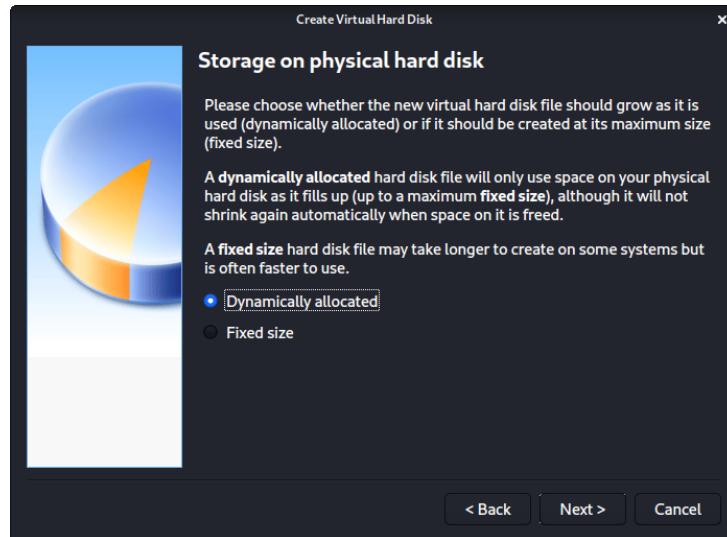
Step 4: Select the option to create a virtual hard disk now and then click on the create button.



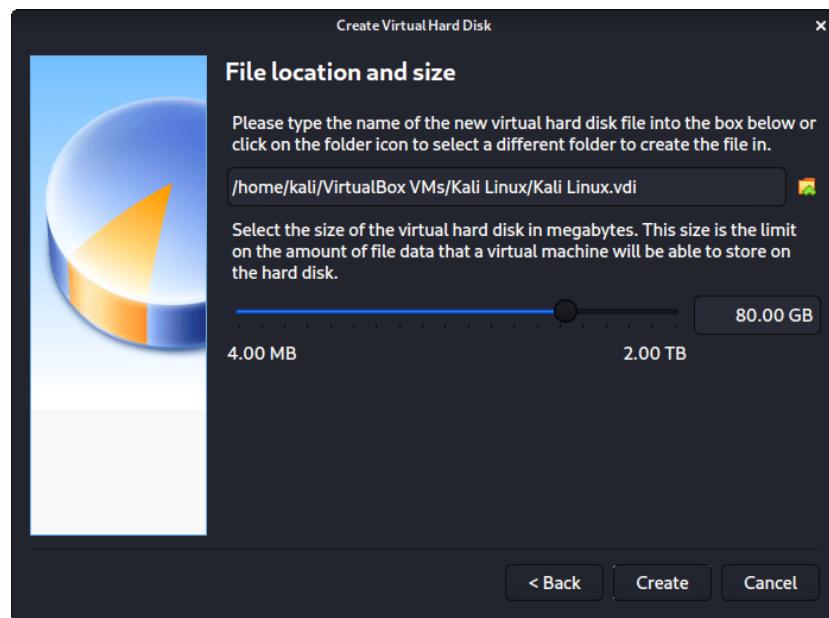
Step 5: Now, Select the Hard Disk File type for Virtual Hard Disk. (VDI is recommended for daily purposes)



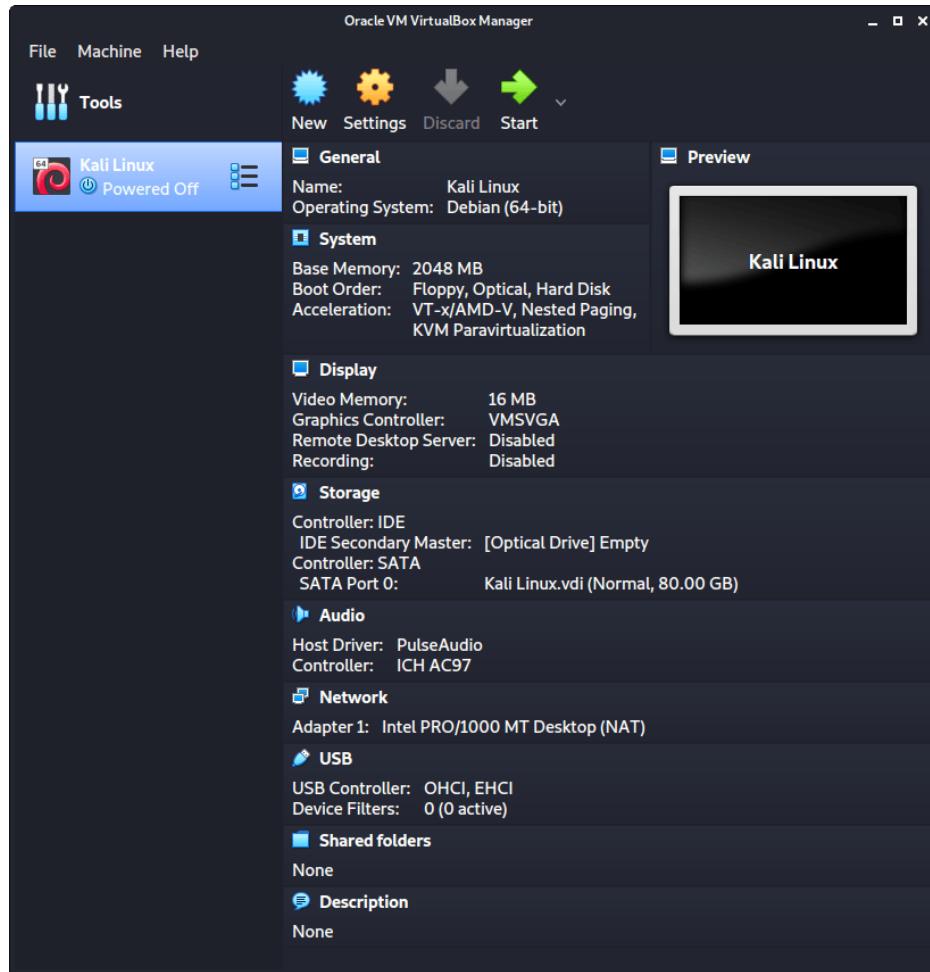
Step 6: Now select the type of Physical Hard Disk Storage. (Dynamically Allocated is recommended for general purposes)



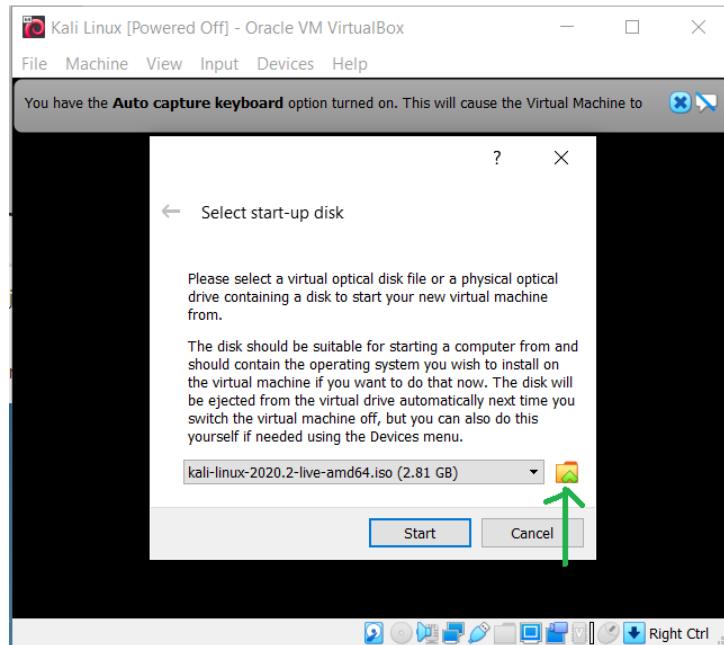
Step 7: Select the size of your virtual hard disk and also the location where you want to save your machine and its files.



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.

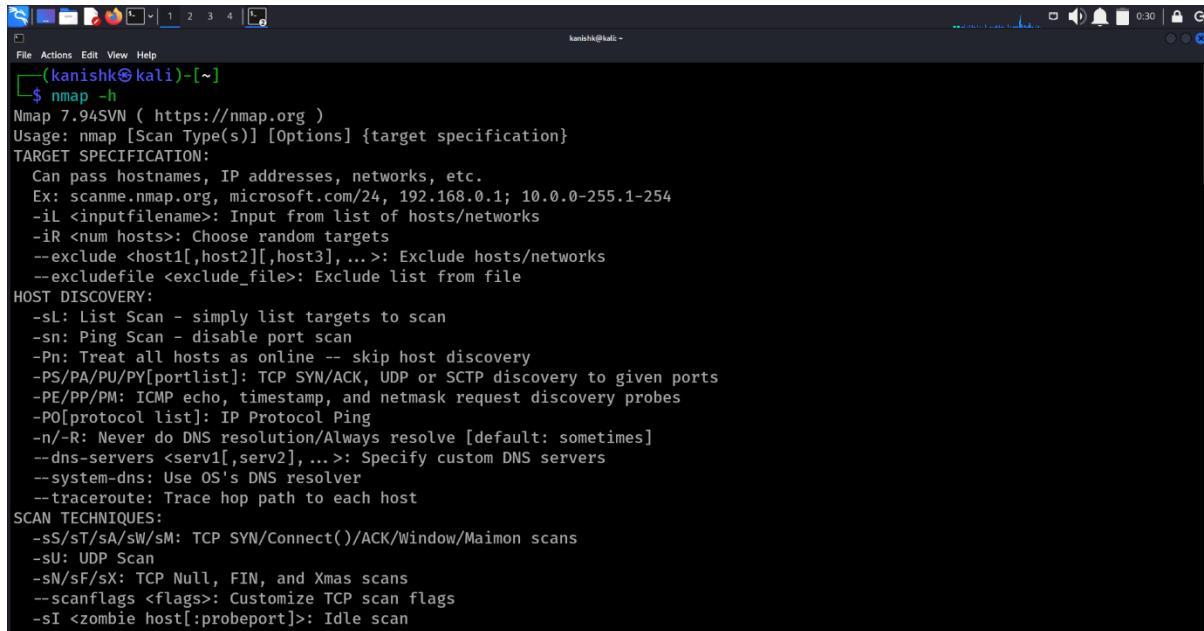
John the Ripper - Password cracking.

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.



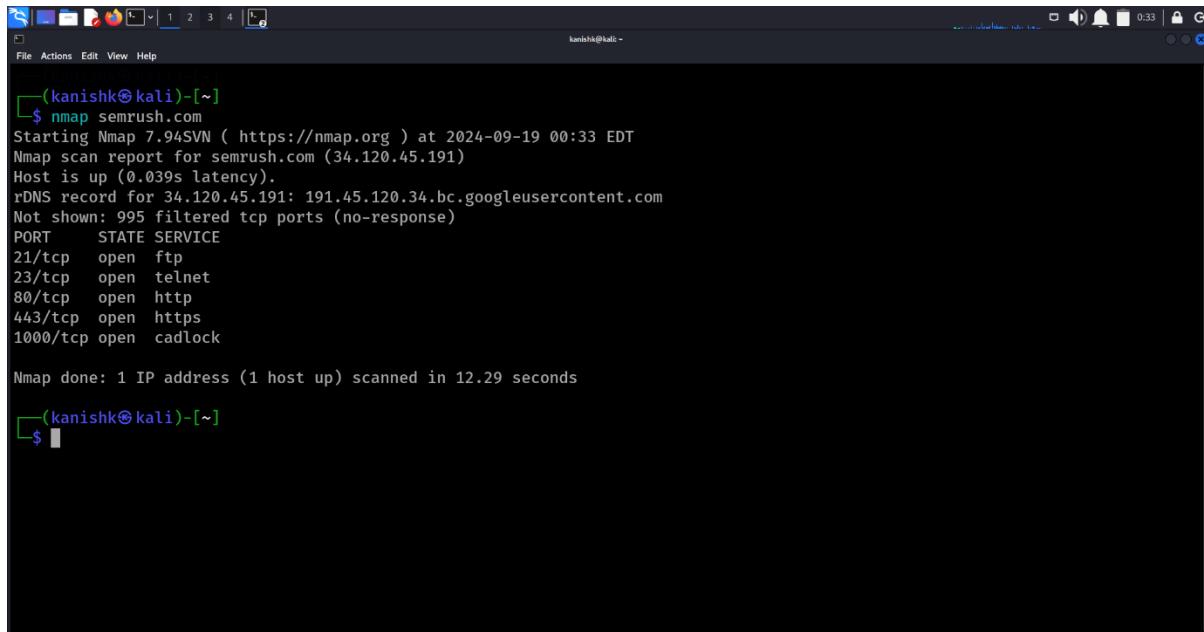
```

File Actions Edit View Help
(kanishk㉿kali)-[~]
$ nmap -h
Nmap 7.94SVN ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan

```

2. Now run **nmap semrush.com**

This will scan for open ports on the domain.

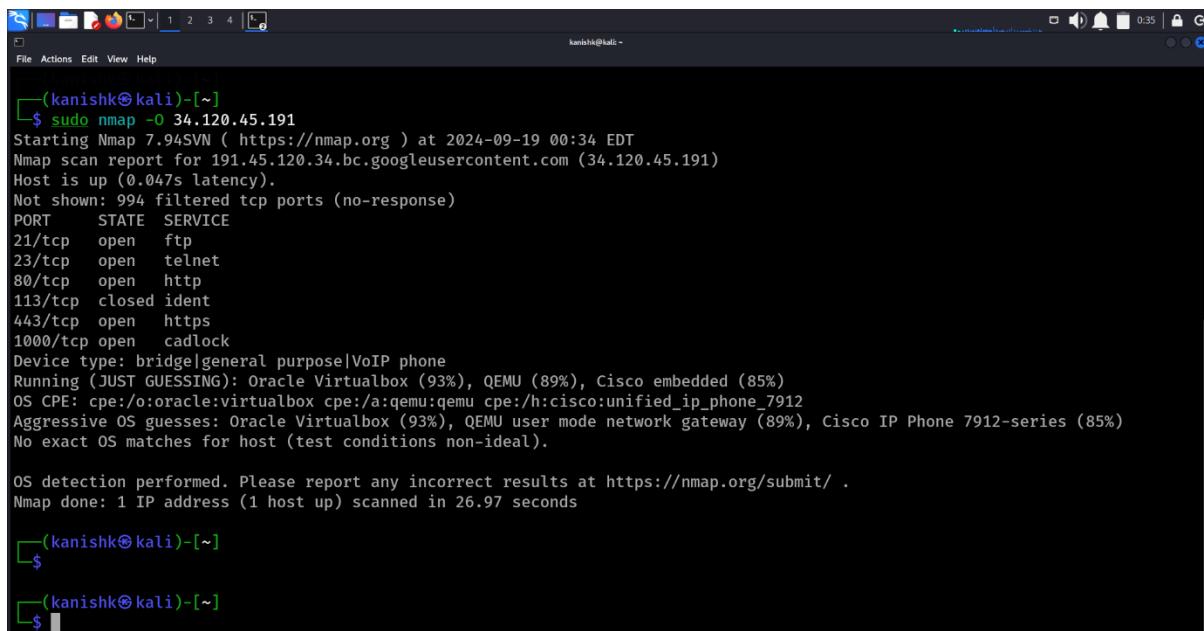


```
(kanishk㉿kali)-[~]
$ nmap semrush.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-19 00:33 EDT
Nmap scan report for semrush.com (34.120.45.191)
Host is up (0.039s latency).
rDNS record for 34.120.45.191: 191.45.120.34.bc.googleusercontent.com
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
80/tcp    open  http
443/tcp   open  https
1000/tcp  open  cadlock

Nmap done: 1 IP address (1 host up) scanned in 12.29 seconds
```

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



```
(kanishk㉿kali)-[~]
$ sudo nmap -o 34.120.45.191
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-19 00:34 EDT
Nmap scan report for 191.45.120.34.bc.googleusercontent.com (34.120.45.191)
Host is up (0.047s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
80/tcp    open  http
113/tcp   closed ident
443/tcp   open  https
1000/tcp  open  cadlock
Device type: bridge|general purpose|VoIP phone
Running (JUST GUESSING): Oracle Virtualbox (93%), QEMU (89%), Cisco embedded (85%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu cpe:/h:cisco:unified_ip_phone_7912
Aggressive OS guesses: Oracle Virtualbox (93%), QEMU user mode network gateway (89%), Cisco IP Phone 7912-series (85%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.97 seconds
```

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 7.94 ( 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    FortiGate Application filtering
| 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></body></html>

```

```

File Actions Edit View Help
kanishk@kali: ~
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=%D=9/19%Time=66EBA4E%P=x86_64-pc-linux-g
SF:nu%r(GetRequest,10E,"HTTP/1\.1\x20200\x200K\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><scri
SF:t><script language="JavaScript">window.location="https://172\.30\.0\.1:1
SF:003/fgtauth?0741300c03600055";</script></body></html>"%)(FourOhFourR
SF:request,10E,"HTTP/1\.1\x20200\x200K\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=%D=9/19%Time=66EBA32%P=x86_64-pc-linux-
SF:gnu%r(GetRequest,10E,"HTTP/1\.1\x20200\x200K\r\nContent-Length:\x20132\r
SF:r\Connection:\x20close\r\nCache-Control:\x20no-cache\r\nContent-Type:\x
SF:x20text/html\r\nX-Frame-Options:\x20SAMEORIGIN\r\n\r\n<html><body><scri
SF:pt><script language="JavaScript">window.location="https://172\.30\.0\.1:
SF:1003/fgtauth\?0547380509317dad";</script></body></html>"%)(FourOhFourR
SF:Request,10E,"HTTP/1\.1\x20200\x200K\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=%D=9/19%Time=66EBA49%P=x86_64-pc-linux

```

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

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

```

File Actions Edit View Help
[~] $ nmap -p 80 -Pn 34.120.45.191
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-19 00:46 EDT
Nmap scan report for 191.45.120.34.bc.googleusercontent.com (34.120.45.191)
Host is up (0.090s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
[~] $ 
[~] $ 
[~] $ 
[~] $ 
[~] $ 
[~] $ 
[~] $ 
[~] $ nmap -p 80 -Pn 34.120.45.191

```

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

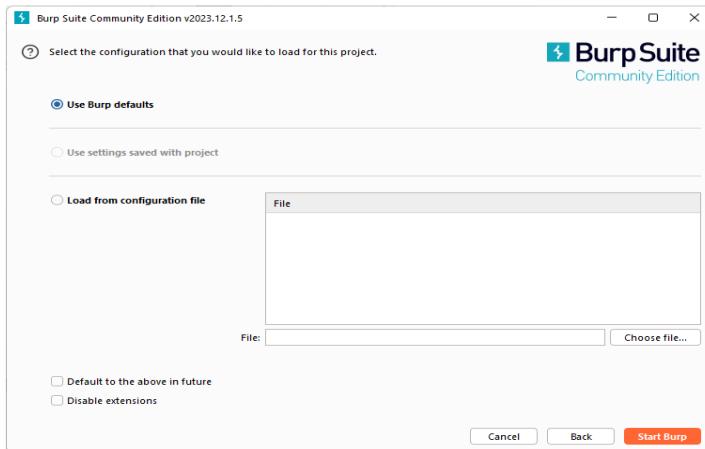
```

File Actions Edit View Help
[~] $ sudo nmap -oG - 34.120.45.191
# Nmap 7.94SVN scan initiated Thu Sep 19 00:40:11 2024 as: nmap -oG - 34.120.45.191
Host: 34.120.45.191 (191.45.120.34.bc.googleusercontent.com)      Status: Up
Host: 34.120.45.191 (191.45.120.34.bc.googleusercontent.com)      Ports: 25/open/tcp//smtp///, 80/open/tcp//http///, 110/open/tcp//pop3///, 113/closed/tcp//ident///, 143/open/tcp//imap///, 443/open/tcp//https///, 8010/open/tcp//xmpp///      Ignored State: filtered (993)
# Nmap done at Thu Sep 19 00:40:22 2024 -- 1 IP address (1 host up) scanned in 10.46 seconds
[~] $ nmap -sV 34.120.45.191
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-19 00:40 EDT
Nmap scan report for 191.45.120.34.bc.googleusercontent.com (34.120.45.191)
Host is up (0.088s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
25/tcp    open  smtp?
80/tcp    open  http
110/tcp   open  pop3?
143/tcp   open  imap?
443/tcp   open  ssl/https
8010/tcp  open  ssl/xmpp?
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%I=7%D=9/19%Time=66EBAB73%P=x86_64-pc-linux-gnu%r(G
SF:etRequest,C2,"HTTP/1.1\x20301\x20Moved\x20Permanently\r\nCache-Control
```

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.



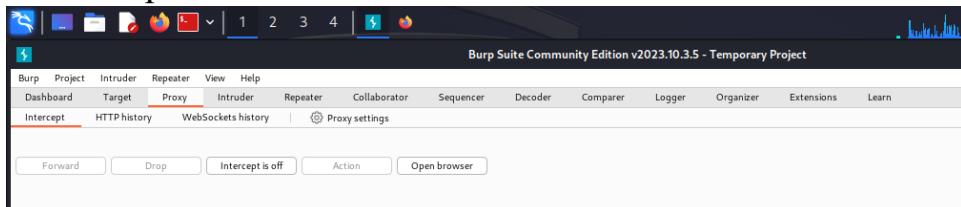
2. Now click the start button, you were redirected here on Burp suite dashboard.

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

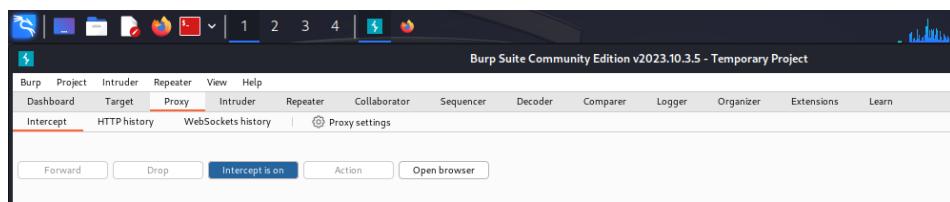
Burp Proxy lets you intercept HTTP requests and responses sent between Burp's browser and the target server. This enables you to study how the website behaves when you perform different actions.

3. Launch Burp's browser

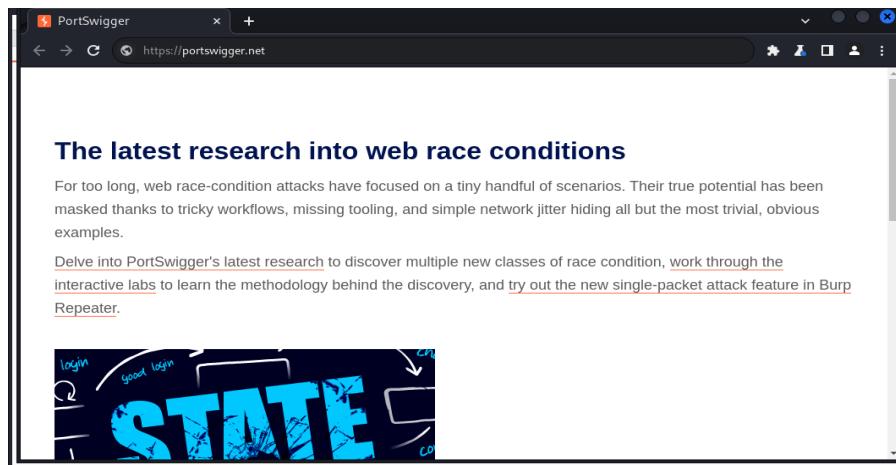
Go to the Proxy > Intercept tab. Set the intercept toggle to Intercept on.
Before intercept tab On.



After intercept tab On.

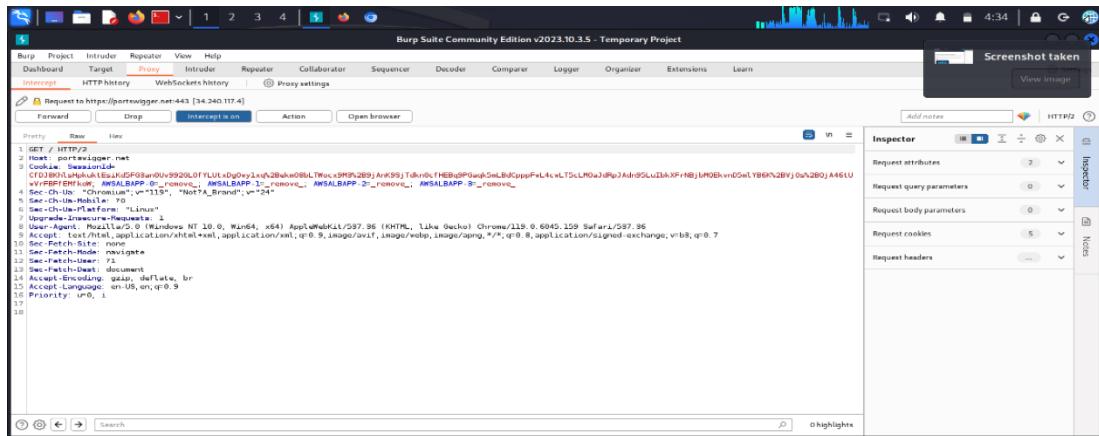


Click **Open Browser**. This launches Burp's browser, which is preconfigured to work with Burp right out of the box.



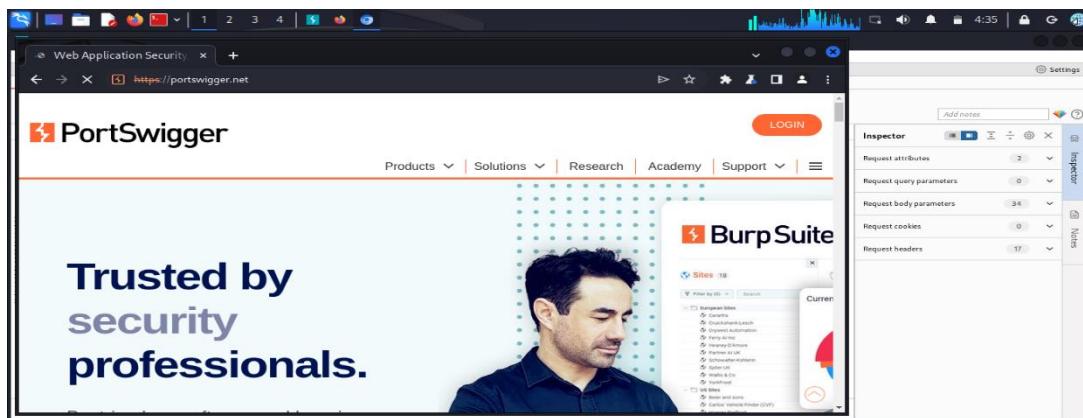
4. Intercept a request

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

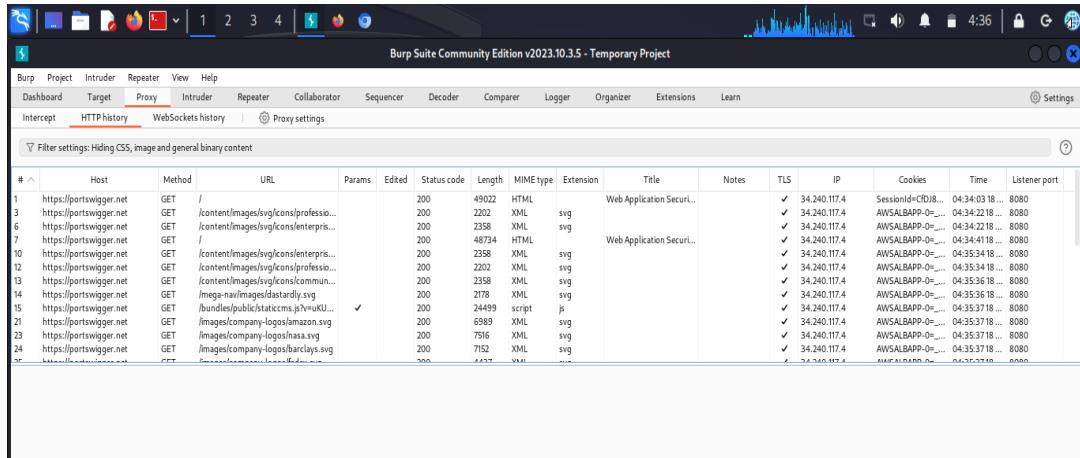
Click the **Forward** button to send the intercepted request. Click **Forward** again to send any subsequent requests that are intercepted, until the page loads in Burp's browser. The **Forward** button sends all the selected requests.



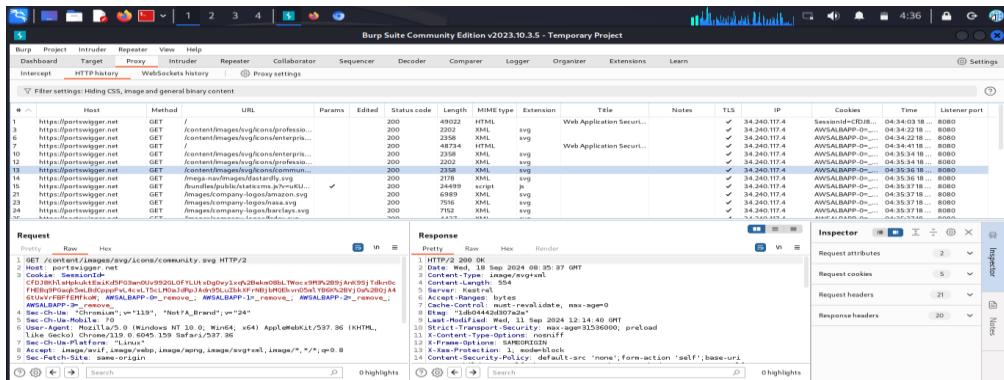
6. Switch off interception

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.

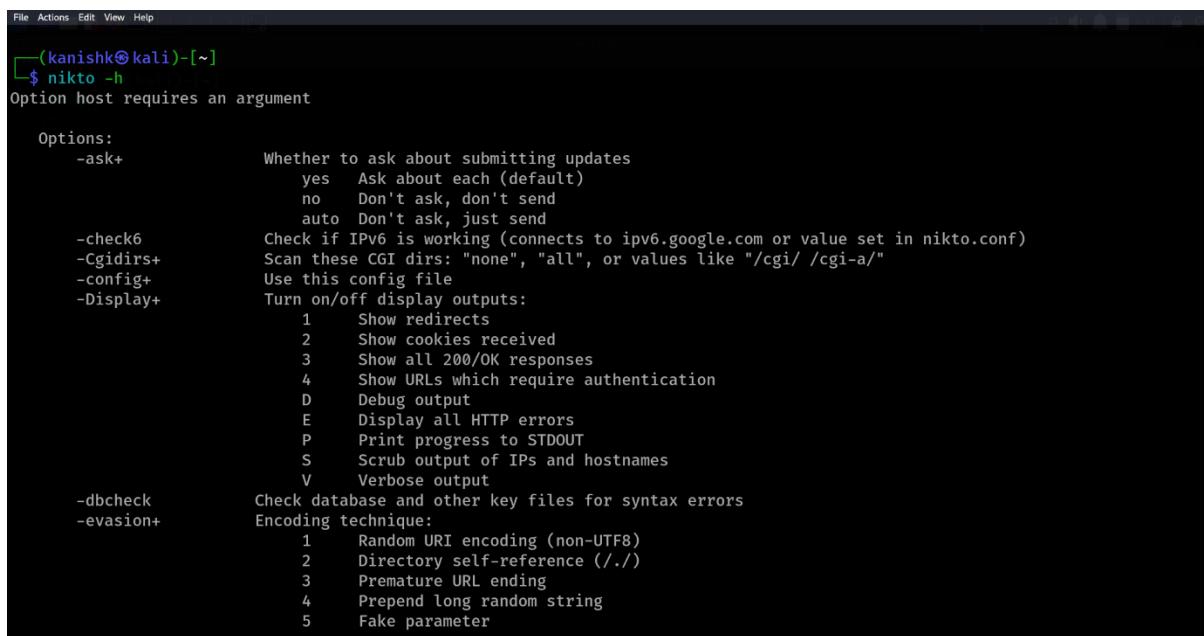


3) WEB SERVER VULNERABILITY SCAN WITH NIKTO

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.



```

File Actions Edit View Help
(kanishk㉿kali)-[~]
$ nikto -h
Option host requires an argument

Options:
  -ask+           Whether to ask about submitting updates
                  yes   Ask about each (default)
                  no    Don't ask, don't send
                  auto  Don't ask, just send
  -check6         Check if IPv6 is working (connects to ipv6.google.com or value set in nikto.conf)
  -Cgidirs+       Scan these CGI dirs: "none", "all", or values like "/cgi /cgi-a/"
  -config+        Use this config file
  -Display+       Turn on/off display outputs:
                  1    Show redirects
                  2    Show cookies received
                  3    Show all 200/OK responses
                  4    Show URLs which require authentication
                  D    Debug output
                  E    Display all HTTP errors
                  P    Print progress to STDOUT
                  S    Scrub output of IPs and hostnames
                  V    Verbose output
  -dbcheck        Check database and other key files for syntax errors
  -evasion+       Encoding technique:
                  1    Random URI encoding (non-UTF8)
                  2    Directory self-reference (./.)
                  3    Premature URL ending
                  4    Prepend long random string
                  5    Fake parameter

```

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

[~] $ 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 Server 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 site 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

[~] $ 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

[~] $ 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 Server 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_httpproxy
  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 Server 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)

```
File Actions Edit View Help
[~] (kanishk㉿kali)-[~]
$ nikto -h https://semrush.com -evasion 2
- 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 Server CA
+ Using Encoding: Directory self-reference (./)
+ Start Time:    2024-09-19 01:22:11 (GMT-4)

+ Server: No banner retrieved
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: 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
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: https://semrush.com/
```

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

```
[File Actions Edit View Help

└─(kanishk㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe09:cb4a prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:09:cb:4a txqueuelen 1000 (Ethernet)
            RX packets 27426 bytes 15587576 (14.8 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 36535 bytes 3991255 (3.8 MiB)
            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 8 bytes 480 (480.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 8 bytes 480 (480.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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

```
[File Actions Edit View Help
└── (kanishk㉿kali)-[~]
$ msfconsole

Metasploit tip: The use command supports fuzzy searching to try and
select the intended module, e.g. use kerberos/get_ticket or use
kerberos forge silver ticket

          .$$$$$L,,=aaccaacc%$b.      d8,    d8P
d8P     #####$$$$$b.    BP d888888p
d888888P '7$$$\\"""^` .7$$$|D*`?88'
          .os#$|8`` d8P    ?8b  88P
88P`?P`?P d8b_,dP 88P d8P' ?88   .oaSH#%$*` d8P d8888b $whi?88b 88b
d88  d8 ?8 88b  88b 88b .os$$$$$*` ?88,.d88b, d88 d8P' ?88 88P`?8b
d88' d88b 8b ?8888P`?8b ?88P'.as$$$$$Q*` ?88' ?88 ?88 88b d88 d88
          .a$$$$$``       88b d8P 88b`?8888P'
          ,s$$$$$``       888888P' 88n  .,,.,ass:
          .a$$$$$``       d8P' .,.ass%$$$$$$$$$$'
          .a$##$$P`       .,-aqsc#SS$$$$$$$$$$$$$$$`$$$$$$'
          ,a$##$$P`       .,-as$#S$$$$$$$$$$$$$$$`$$$$$$`$#SSSS'
          .a$$$$$$$$$$$`$#=-"``/$$$$$'
          ,$$$$$`         lll66666'
          ..;lll6666'
          ...;;lllll6'
          .....;;;lll;;;.....
          .....;;;... .
```

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.

```

msf6 > search
Usage: search [<options>] [<keywords>:<value>]

Prepending a value with '-' will exclude any matching results.
If no options or keywords are provided, cached results are displayed.

OPTIONS:

-h, --help           Help banner
-I, --ignore         Ignore the command if the only match has the same name as the search
-o, --output <filename> Send output to a file in csv format
-r, --sort-descending <column> Reverse the order of search results to descending order
-S, --filter <filter> Regex pattern used to filter search results
-s, --sort-ascending <column> Sort search results by the specified column in ascending order
-u, --use             Use module if there is one result

Keywords:
  adapter      : Modules with a matching adater reference name
  aka          : Modules with a matching AKA (also-known-as) name
  author       : Modules written by this author
  arch         : Modules affecting this architecture
  bid          : Modules with a matching Bugtraq ID
  cve          : Modules with a matching CVE ID
  edb          : Modules with a matching Exploit-DB ID
  check        : Modules that support the 'check' method
  date         : Modules with a matching disclosure date
  description   : Modules with a matching description

```

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

Name	Current Setting	Required	Description
CONCURRENCY	10	yes	The number of concurrent ports to check per host
DELAY	0	yes	The delay between connections, per thread, in milliseconds
JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS	1	yes	The number of concurrent threads (max one per host)
TIMEOUT	1000	yes	The socket connect timeout in milliseconds

View the full module info with the `info`, or `info -d` command.

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.

```
View the full module info with the info, or info -d command.

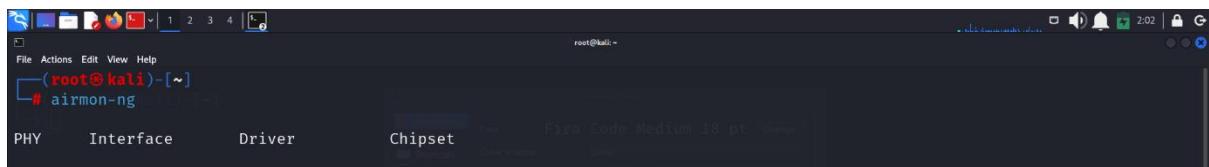
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.71.209
RHOSTS => 192.168.71.209
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 22,80,21,404,9929,31332,110,25
PORTS => 22,80,21,404,9929,31332,110,25
msf6 auxiliary(scanner/portscan/tcp) > set THREADS 5
THREADS => 5
msf6 auxiliary(scanner/portscan/tcp) >
msf6 auxiliary(scanner/portscan/tcp) > █
```

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.



```
root@kali: ~# airmon-ng
```

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



```
root@kali: ~# airmon-ng -h
usage: airmon-ng <start|stop|check> <interface> [channel or frequency]
```

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

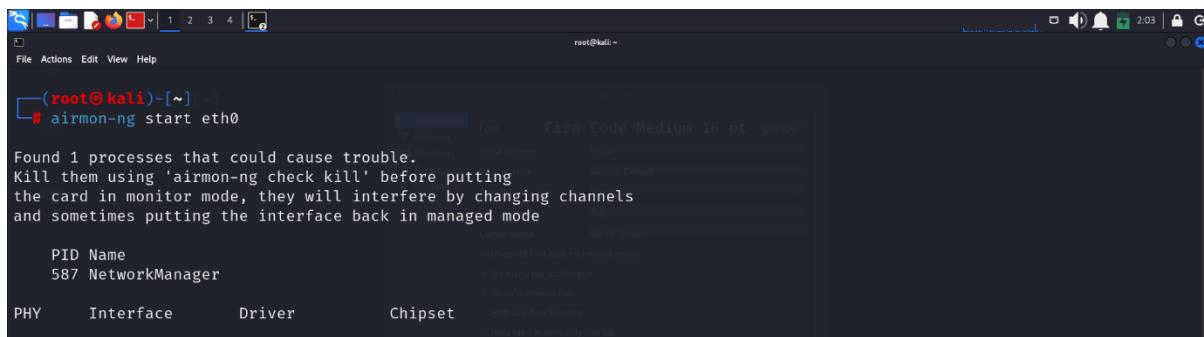


```
root@kali: ~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe09:cb4a prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:09:cb:4a txqueuelen 1000 (Ethernet)
                RX packets 5092 bytes 355978 (347.6 KiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 20808 bytes 1538584 (1.4 MiB)
                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 8 bytes 480 (480.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 8 bytes 480 (480.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

We get eth0 and lo interface.

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

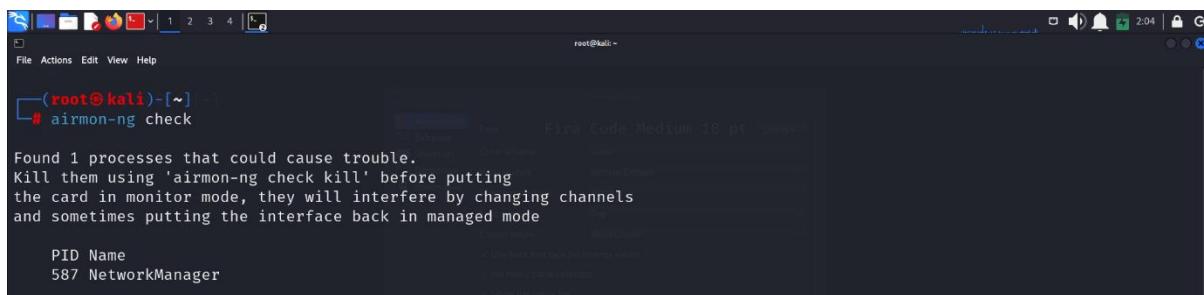


```
root@kali:~# airmon-ng start eth0
Found 1 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
587 NetworkManager
```

Here we get some trouble.

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



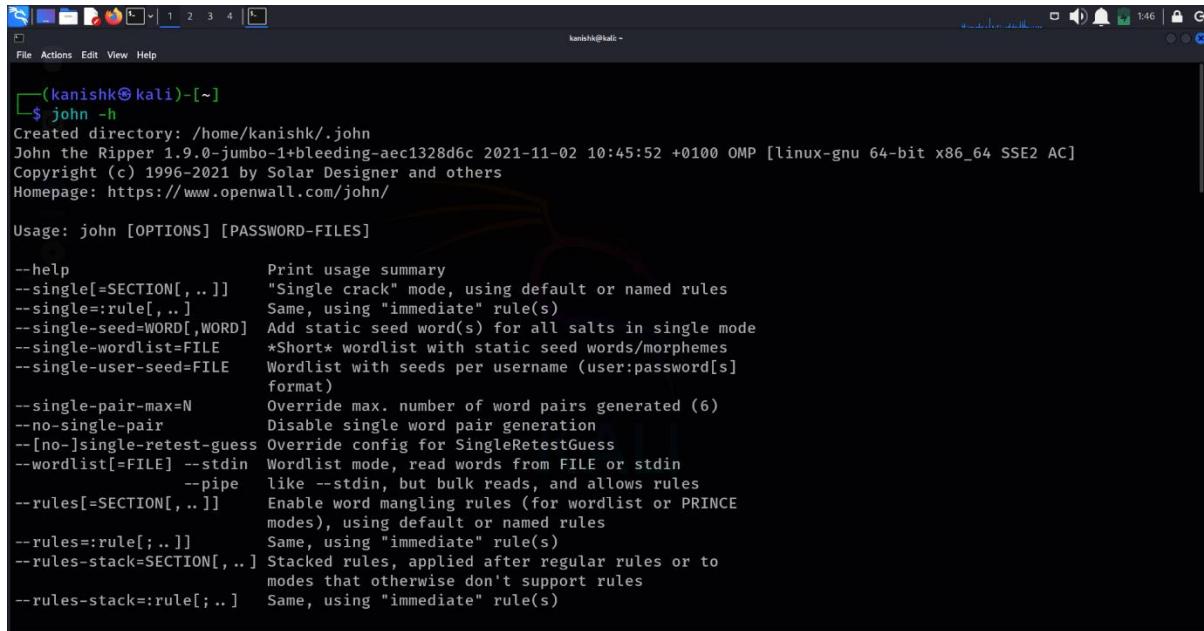
```
root@kali:~# airmon-ng check
Found 1 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
587 NetworkManager
```

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.



```
kanishk@kali:~$ john -h
Created directory: /home/kanishk/.john
John the Ripper 1.9.0-jumbo-1+bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 OMP [linux-gnu 64-bit x86_64 SSE2 AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/

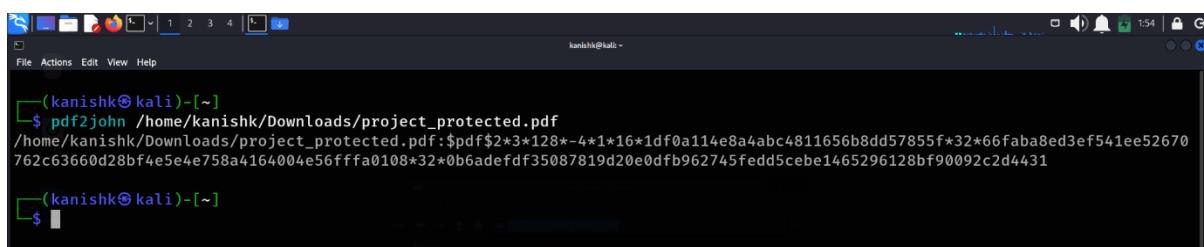
Usage: john [OPTIONS] [PASSWORD-FILES]

--help          Print usage summary
--single[=SECTION[,..]] "Single crack" mode, using default or named rules
--single=:rule[,..] Same, using "immediate" rule(s)
--single-seed=WORD[,WORD] Add static seed word(s) for all salts in single mode
--single-wordlist=FILE *Short* wordlist with static seed words/morphemes
--single-user-seed=FILE Wordlist with seeds per username (user:password[s]
format)
--single-pair-max=N Override max. number of word pairs generated (6)
--no-single-pair Disable single word pair generation
--[no-]single-retest-guess Override config for SingleRetestGuess
--wordlist[=FILE] --stdin Wordlist mode, read words from FILE or stdin
      --pipe like --stdin, but bulk reads, and allows rules
--rules[=SECTION[,..]] Enable word mangling rules (for wordlist or PRINCE
modes), using default or named rules
--rules=:rule[,..] Same, using "immediate" rule(s)
--rules-stack=SECTION[,..] Stacked rules, applied after regular rules or to
modes that otherwise don't support rules
--rules-stack=:rule[,..] Same, using "immediate" rule(s)
```

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

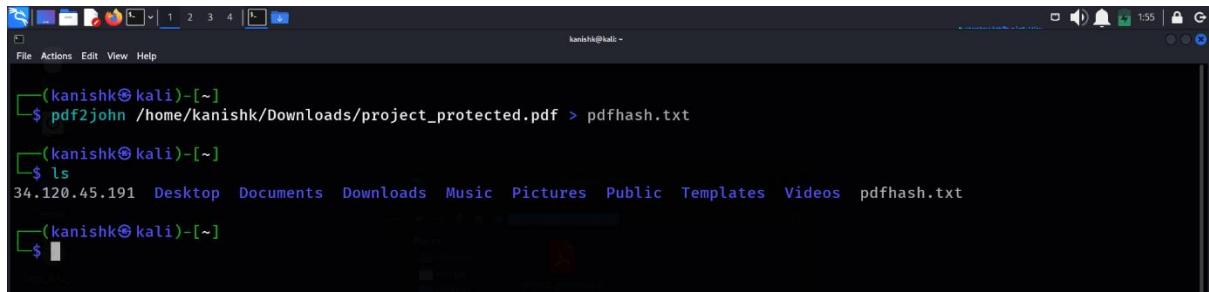
The pdf2john tool is used to extract the hash from a password-protected PDF file, which John the Ripper can then attempt to crack.

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



```
kanishk@kali:~$ pdf2john /home/kanishk/Downloads/project_protected.pdf
/home/kanishk/Downloads/project_protected.pdf:$pdf$2*3*128*-4*1*16*1df0a114e8a4abc4811656b8dd57855f*32*66fab8ed3ef541ee52670
762c63660d28bf4e5e4e758a4164004e56ffffa0108*32*0b6adefdf35087819d20e0dfb962745fedd5cebe1465296128bf90092c2d4431
kanishk@kali:~$
```

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



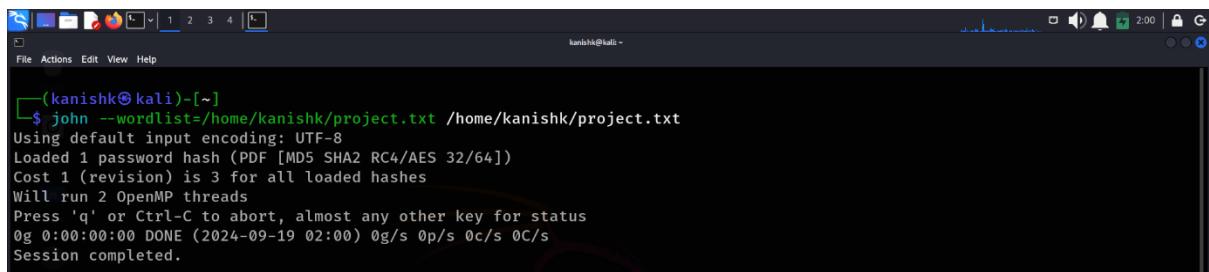
```
(kanishk㉿kali)-[~]
$ pdf2john /home/kanishk/Downloads/project_protected.pdf > pdfhash.txt
(kanishk㉿kali)-[~]
$ ls
34.120.45.191 Desktop Documents Downloads Music Pictures Public Templates Videos pdfhash.txt
(kanishk㉿kali)-[~]
$
```

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.

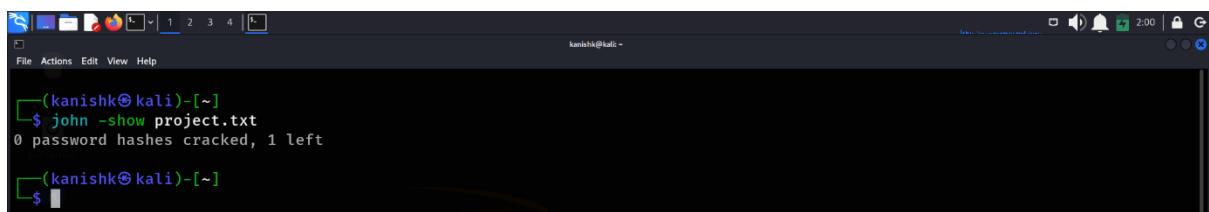


```
(kanishk㉿kali)-[~]
$ john --wordlist=/home/kanishk/project.txt /home/kanishk/project.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
Cost 1 (revision) is 3 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 DONE (2024-09-19 02:00) 0g/s 0p/s 0c/s 0C/s
Session completed.
```

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.



```
(kanishk㉿kali)-[~]
$ john -show project.txt
0 password hashes cracked, 1 left
(kanishk㉿kali)-[~]
$
```