PROJECT REPORT

WINDOWS OS EXPLOITATION

SUBMITTED BY

SARTHAK SINGH GAUR

(S7S5G4@GMAIL.COM)

ACMEGRADE

IIT BOMBAY

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Windows operating system exploitation project requires one to have at least two virtual machines, one for attacking and the other one for the target. The attacking virtual machine requires a hacking-oriented Linux, such as Kali Linux or ParrotOS Linux. The victim device needs to be any Windows operating system after Windows XP SP1. The Linux needs to be updated to the latest version. When starting the exploitation, attacker and victim need to be on same network for the simulation to be effective. Internet connection or at least DHCP connection is also required if manual IP assignment is not present for both devices.

The initial reconnaissance will be done from attacker's device only and it would require the use of Nmap. The payload generation part of the project requires in-depth knowledge of Metasploit Framework, specifically msfvenom. It provides plethora of choices from a huge collection of payloads for different architecture, vulnerabilities, encoders, auxiliary modules, etc. The initial reconnaissance will dictate what payloads to use. Accurate IP addresses and ports are to be decided and preserved for the exploitation stage. Already used ports shouldn't be used. Instead, the dynamic ports are preferred in order to prevent any possible clash.

Furthermore, the use of correct encoders and iteration is essential as it can render the payload unusable if incompatible combination is applied. Services like VirusTotal become necessity when it comes to check the encoders and it's encapsulating strength. Payload Spoofing will also help to create a more realistic project simulation. Once the spoofing and encoding is complete, the payload is sent to the victim machine disguised as something harmless, such as a setup for a famous software or an attachment along an email.

After payload generation is finished, vulnerability exploitation stage starts. The attacker prepares his/her device for incoming connection from the victim once the payload infects their system. It again requires Metasploit Framework, specifically msfconsole. Previously preserved ID addresses and port number are used to setup a receiving server which will act as Attacker's path to control the victim's device once it is compromised. The moment Victim runs the payload (providing it does not get caught by antivirus), it will start looking for attacker's device to connect to. The Attacker who is readily available grabs the opportunity and takes over the victim's device. The victim's device gets compromised without alerting the victim.

Post Exploitation practices begin, and the attacker secures his unauthorized control over the victim's device by escalating privileges to device administrator. Next the attacker creates persistence for the payload that has infected the victim's device. It creates an autorun script which ensures the payload's activation even after reboots. Thus securing the attacker's grasp over victim's device. Afterwards the attacker is free to do as he/she wishes, be it recording victim's keyboard strokes, or capturing account password hashes, or even access personal files.

INTRODUCTION

In today's technologically evolving world, more and more services and facilities are turning online and digitalized, be it health records, school marksheets, or simply payments. But such rapid field of opportunities also attract some shady elements who want to take advantage of the system for their personal gain and sometimes for the loss of others. These people are called Hackers. They pose a major threat to everyone and every service available online. To counter such elements, ethical hackers and security specialist are needed. They are the ones to counter them and secure infrastructures from their numerous attempts to break in.

Just like hackers are getting better and better at breaking in, the frontline defense also need to prepare better to counter them. To stop someone with a string intention and tools, one needs to know their thinking pattern, their priorities, their tactics, techniques and procedures, and create defense beforehand to counter them. In that spirit, this project was created to give cyber security specialists and ethical hackers to act like hackers on a simulated situation to get to know their adversary profiles and actions more effectively. During the entire project and it's practice, it does not cross ethical boundaries.

This Project entails the vulnerability exploitation of the most popular operating system Microsoft Windows using a very popular hacking tool by the company 'rapid7', known as Metasploit Framework. It is an payload generator, encoder, packer as well as controller software. It can create payloads for almost all platforms, encode them into bypassable packages and control them if needed. It's most popular sub framework is Metasploit interpreter, or in short meterpreter.

The attack will simulate real world scenario where the payload is sent in disguise of something harmless and explore the potential it has when it is not stopped. The skills, mental power and training of cybersecurity professionals and ethical hackers will be tested through this project and it will encourage one to learn more in order to get strong enough to step these attacks. It's primary objective to make would more secure will be fulfilled.

METHODOLOGY

Fig 3.1: Hacker OS

The project stimulated the Hacker using Kali Linux as shown in figure 3.1.

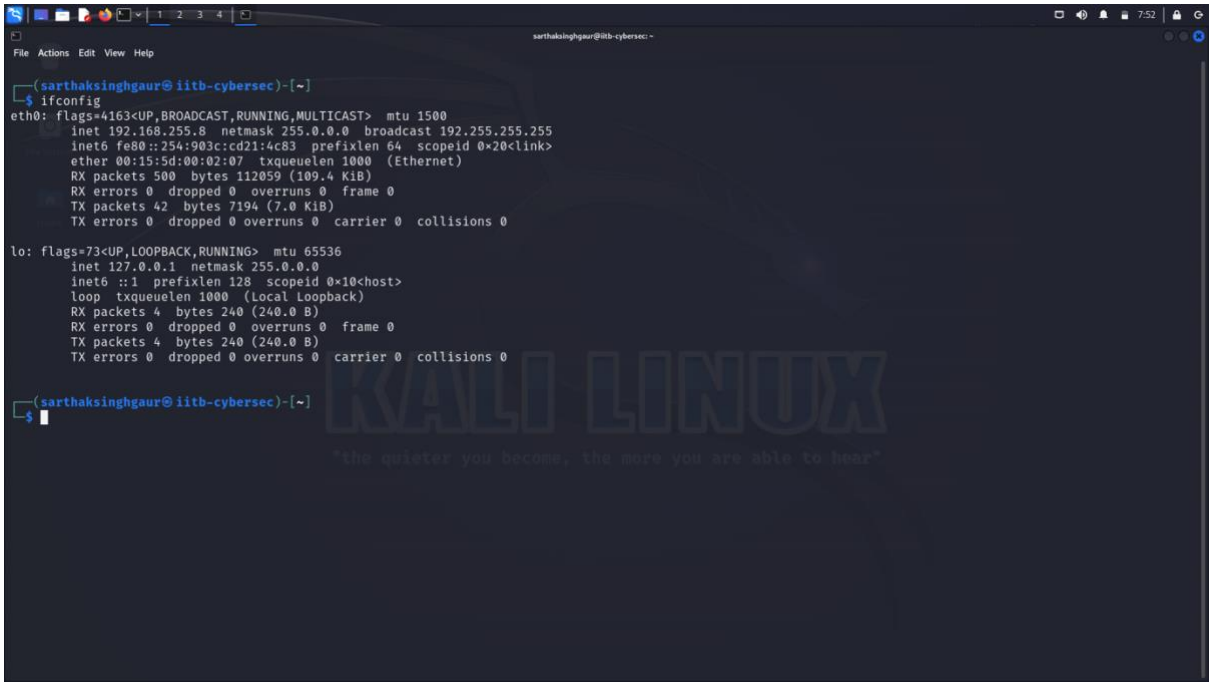

Fig 3.2: Hacker Network Status

The hacker is on a Network with IP Address 192.168.255.8 and simulated MAC address 00:15:5d:00:02:07 as depicted in figure 3.2.

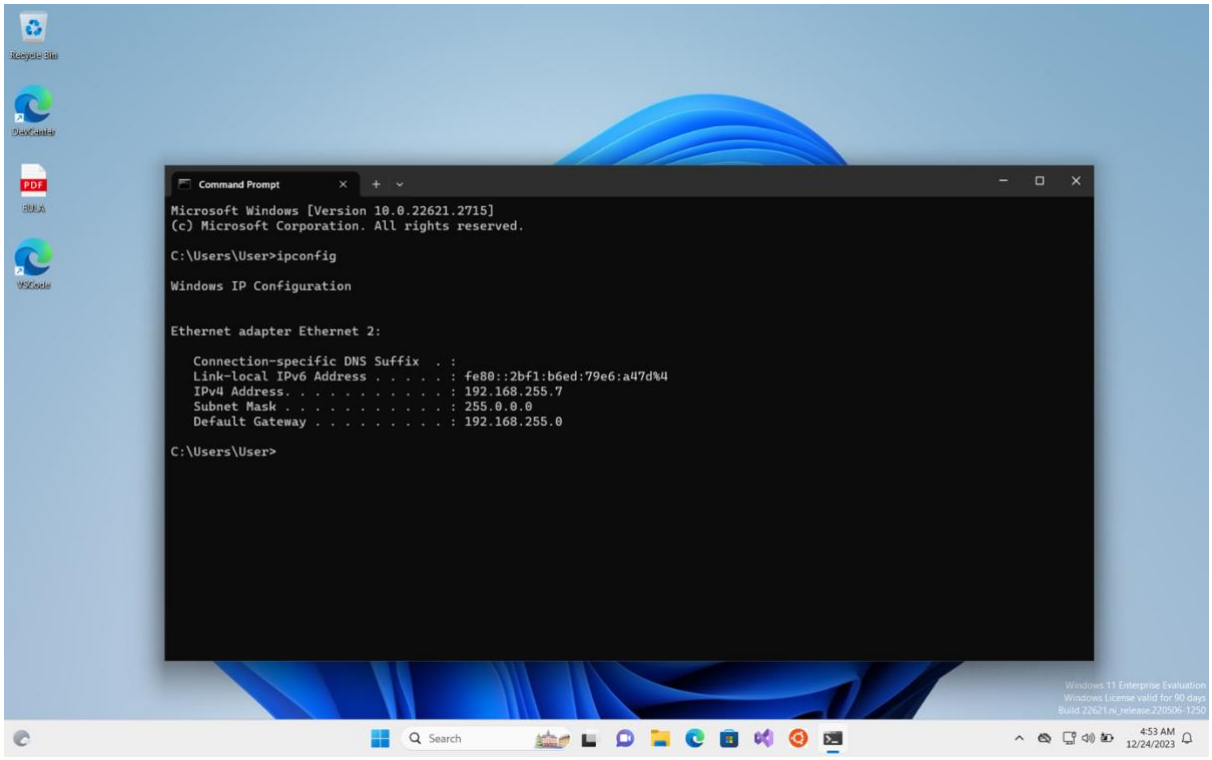

Fig 3.3: Victim OS and Network Status

The victim is situated on the same network with Microsoft Windows OS. The Victim IP Address is 192.168.255.7 as presented in figure 3.3.

METASPLOIT FRAMEWORK (MSFVENOM)

The Metasploit Project is a computer security project that provides information about security vulnerabilities and aids in penetration testing and IDS signature development. It is owned by Boston, Massachusetts-based security company Rapid7.

Its best-known sub-project is the open-source Metasploit Framework, a tool for developing and executing exploit code against a remote target machine. Other important sub-projects include the Opcode Database, shellcode archive and related research.

Metasploit currently has over 592 payloads. Some of them are:

- Command shell enables users to run collection scripts or run arbitrary commands against the host.
- Meterpreter (the Metasploit Interpreter) enables users to control the screen of a device using VNC and to browse, upload and download files.
- Dynamic payloads enable users to evade anti-virus defense by generating unique payloads.
- Static payloads enable static IP address/port forwarding for communication between the host and the client system.

Fig 3.4: Metasploit Framework in Kali Linux

Payload modules are stored in `modules/payloads/{singles,stages,stagers}/<platform>`. When the framework starts up, stages are combined with stagers to create a complete payload that you can use in exploits. Then, handlers are paired with payloads so the framework will know how to create sessions with a given communications mechanism.

Payloads are given reference names that indicate all the pieces, like so:

- Staged payloads: `<platform>/[arch]/<stage>/<stager>`
- Single payloads: `<platform>/[arch]/<single>`

This results in payloads like `windows/x64/meterpreter/reverse_tcp`. Breaking that down, the platform is `windows`, the architecture is `x64`, the final stage we're delivering is `meterpreter`, and the stager delivering it is `reverse_tcp`.

Singles: Single payloads are fire-and-forget. They can create a communications mechanism with Metasploit, but they don't have to. An example of a scenario where you might want a single is when the target has no network access – a file format exploit delivered via USB key is still possible.

Stagers: Stagers are a small stub designed to create some form of communication and then pass execution to the next stage. Using a stager solves two problems. First, it allows us to use a small payload initially to load up a larger payload with more functionality. Second, it makes it possible to separate the communications mechanism from the final stage so one payload can be used with multiple transports without duplicating code.

Stages: Since the stager will have taken care of dealing with any size restrictions by allocating a big chunk of memory for us to run in, stages can be arbitrarily large. One advantage of that is the ability to write final-stage payloads in a higher-level language like C.

Msfvenom is the combination of payload generation and encoding. It's a replacement for msfpayload and msfencode. It supports the following options:

Options:

-p, --payload	<payload>	Payload to use. Specify a '-' or stdin to use custom payloads
-l, --list	[module_type]	List a module type example: payloads, encoders, nops, all
-n, --nopsled	<length>	Prepend a nopsled of [length] size on to the payload
-f, --format	<format>	Output format (use --help-formats for a list)
-e, --encoder	[encoder]	The encoder to use

To generate a payload, we use the -p flag. As an example, we use a payload and specify the required details such as LHOST, LPORT and encoder.

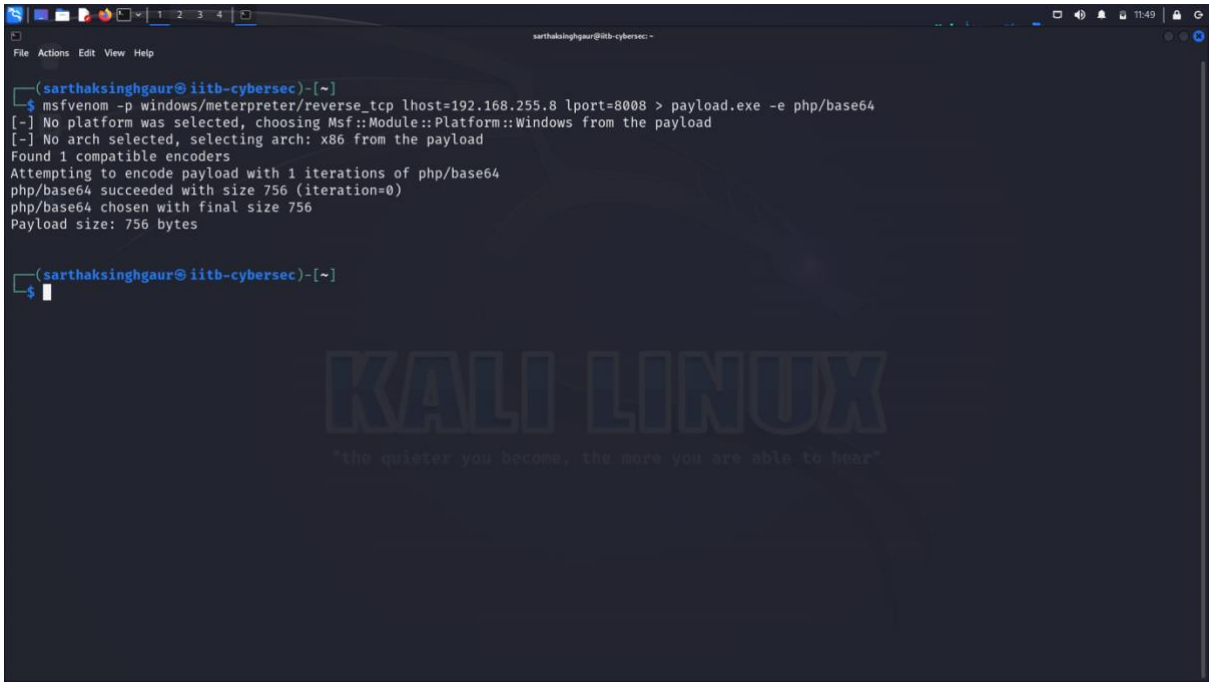

Fig 3.5: Msfvenom Payload Generation Example

When we generate a reverse shell with either msfpayload or msfvenom, we configure the following:

LHOST - This is the IP address we want our target machine to connect to. If we're in a local area network, it is unlikely our target machine can reach us unless we both are on the same network. In that case, we will have to find out your public-facing IP address, and then configure your network to port-forward that connection to our box. LHOST should not be "localhost", or "0.0.0.0", or "127.0.0.1", because if we do, we're telling the target machine to connect to itself (or it may not work at all).

LPORT - This the port we want our target machine to connect to.

METASPLOIT FRAMEWORK (MSFCONSOLE)

MSFconsole provides a command line interface to access and work with the Metasploit Framework. The MSFconsole is the most commonly used interface to work with the Metasploit Framework. The console lets you do things like scan targets, exploit vulnerabilities, and collect data.

Fig 4.1: Msfconsole under Metasploit framework

To run MSFconsole on kali, we open a terminal, cd into the framework directory and type: `msfconsole` There are two popular types of shells: bind and reverse.

- Bind shell - Opens up a new service on the target machine and requires the attacker to connect to it to get a session.
- Reverse shell - A reverse shell is also known as a connect-back. It requires the attacker to set up a listener first on his box, the target machine acts as a client connecting to that listener, and then finally, the attacker receives the shell.

In Windows, the most commonly used reverse shell is `windows/meterpreter/reverse` . We can also use `windows/meterpreter/reverse_http` or `windows/meterpreter/reverse_https` because their network traffic appears a little bit less abnormal.

Fig 4.2: Msfconsole Exploitation Example

When we set up a listener for the reverse shell, we also at least need to configure LHOST and LPORT, but slightly different meanings (different perspective):

- LHOST - This is the IP address we want our listener to bind to.
- LPORT - This is the port we want our listener to bind to.

Once the victim unknowingly opens the payload, it will try to connect to the attackers ID and it would look as shown in figure 4.3.

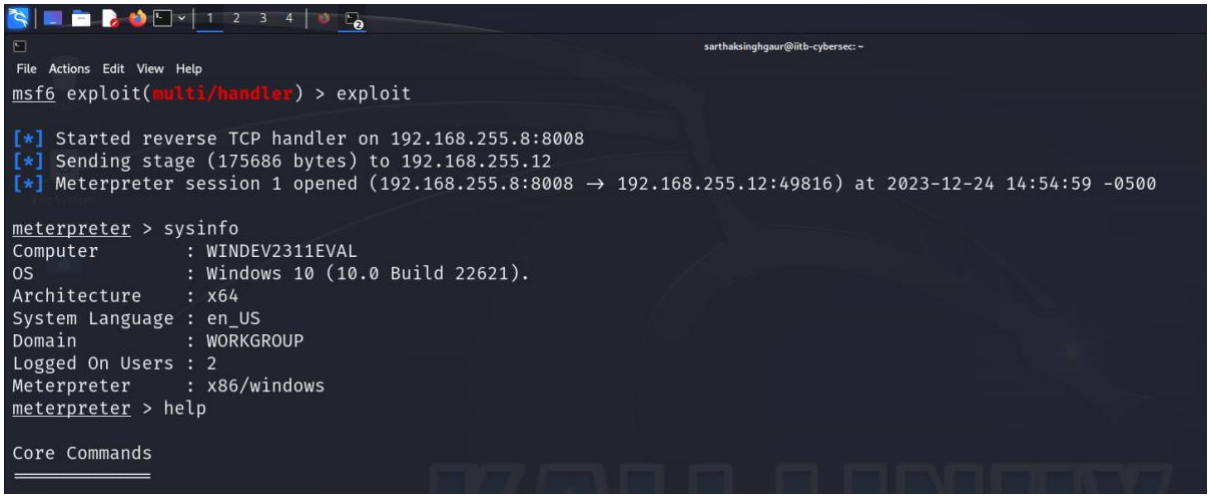

Fig 4.3: Msfconsole Exploitation Post Command Example

FINDINGS

KALI LINUX AS HACKER OS

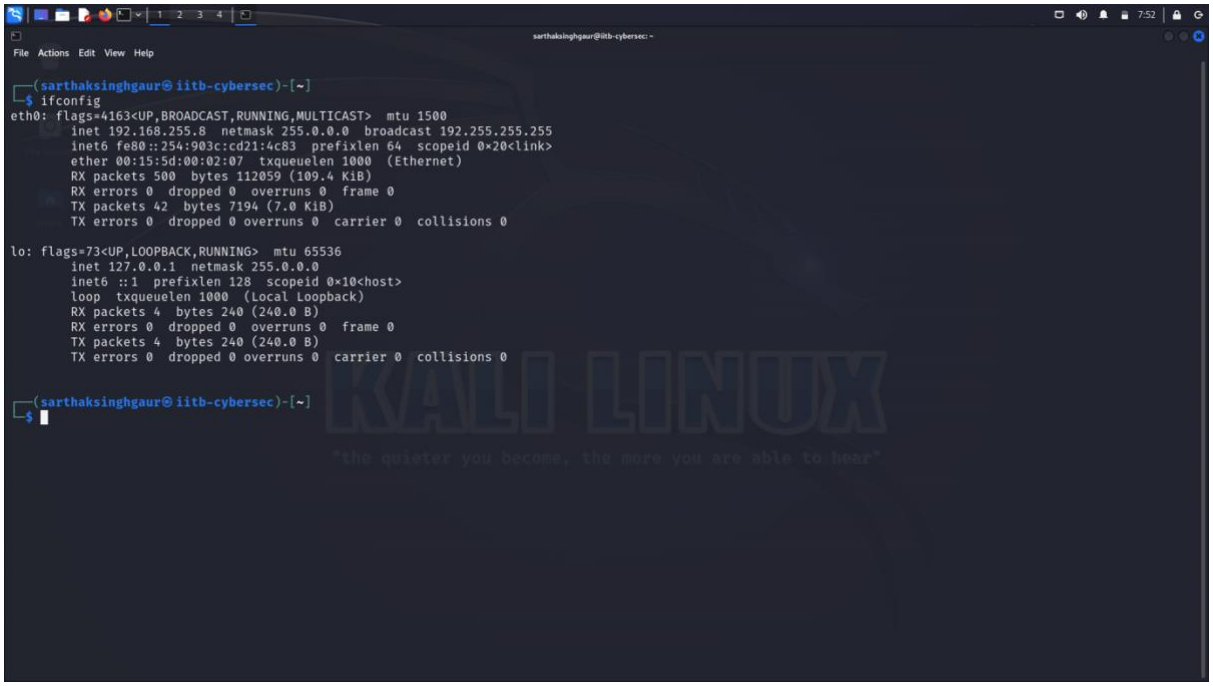

Fig 5.1: Attacker Kali Linux Initial Stage

MICROSOFT WINDOWS AS VICTIM OS

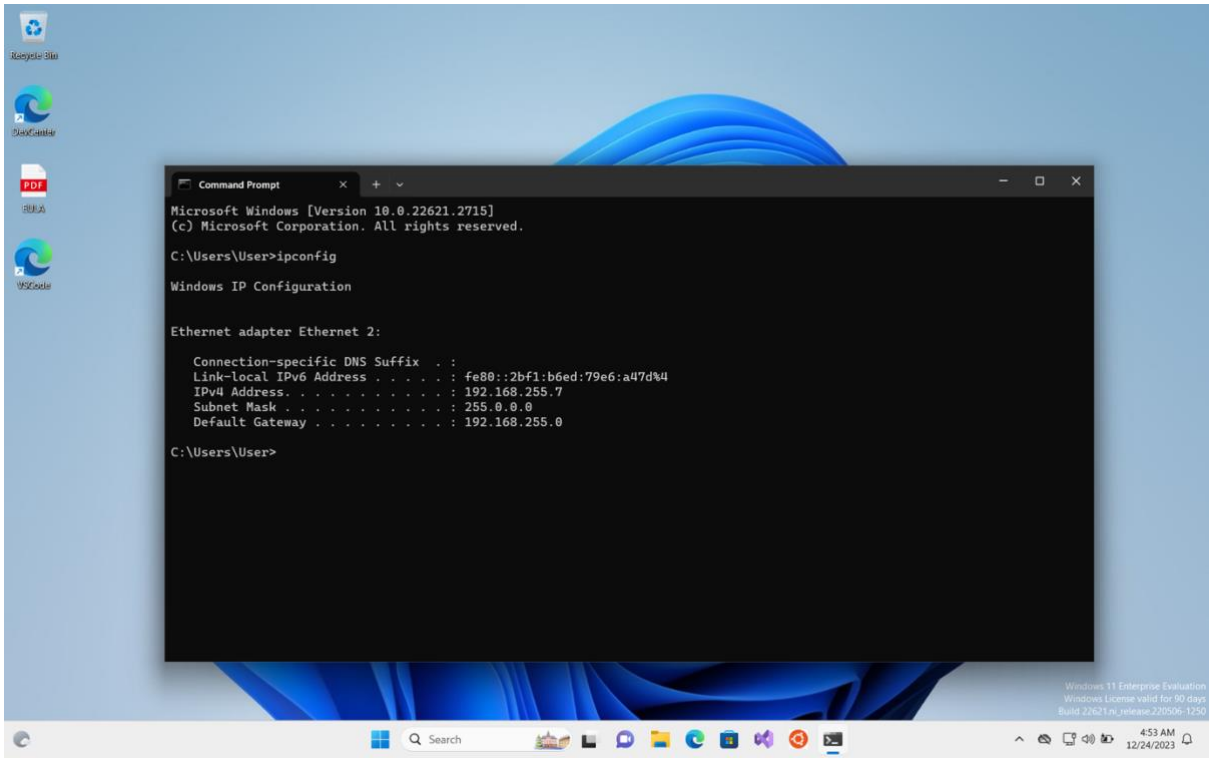

Fig 5.2: Victim Windows Initial Stage

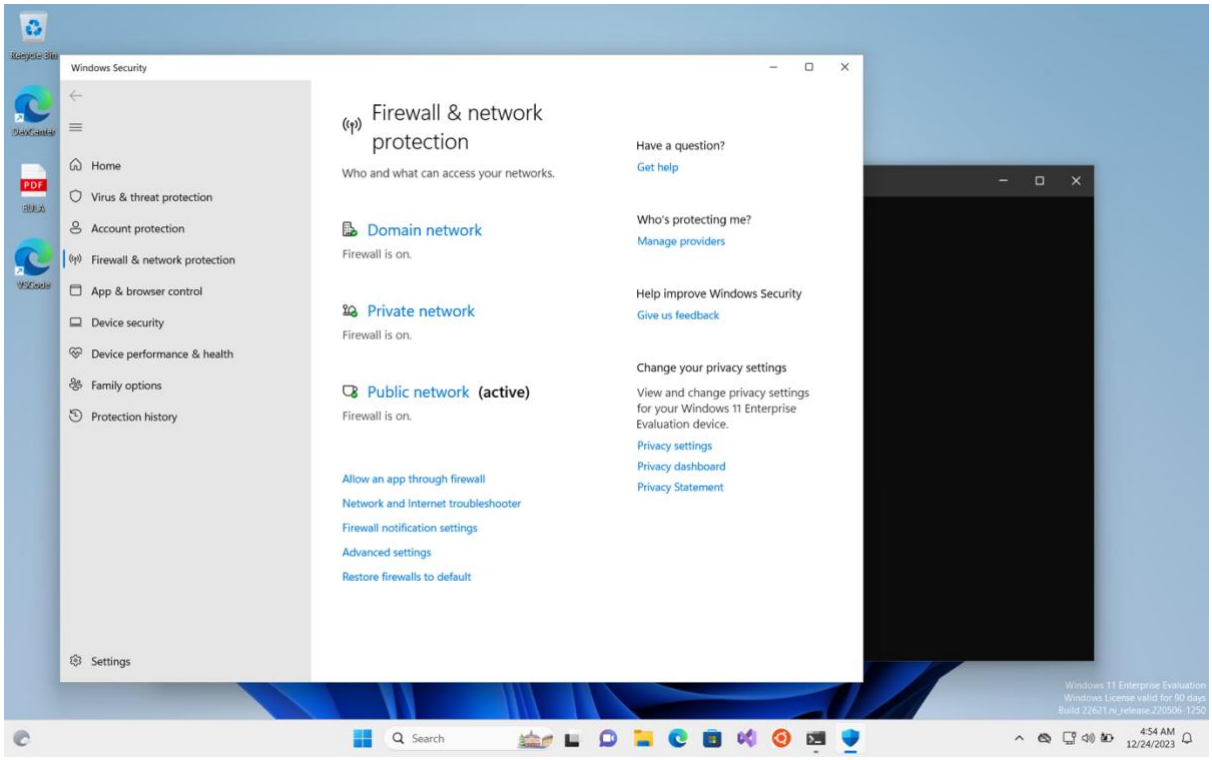

Fig 5.3: Victim Windows Initial Stage Pt.2

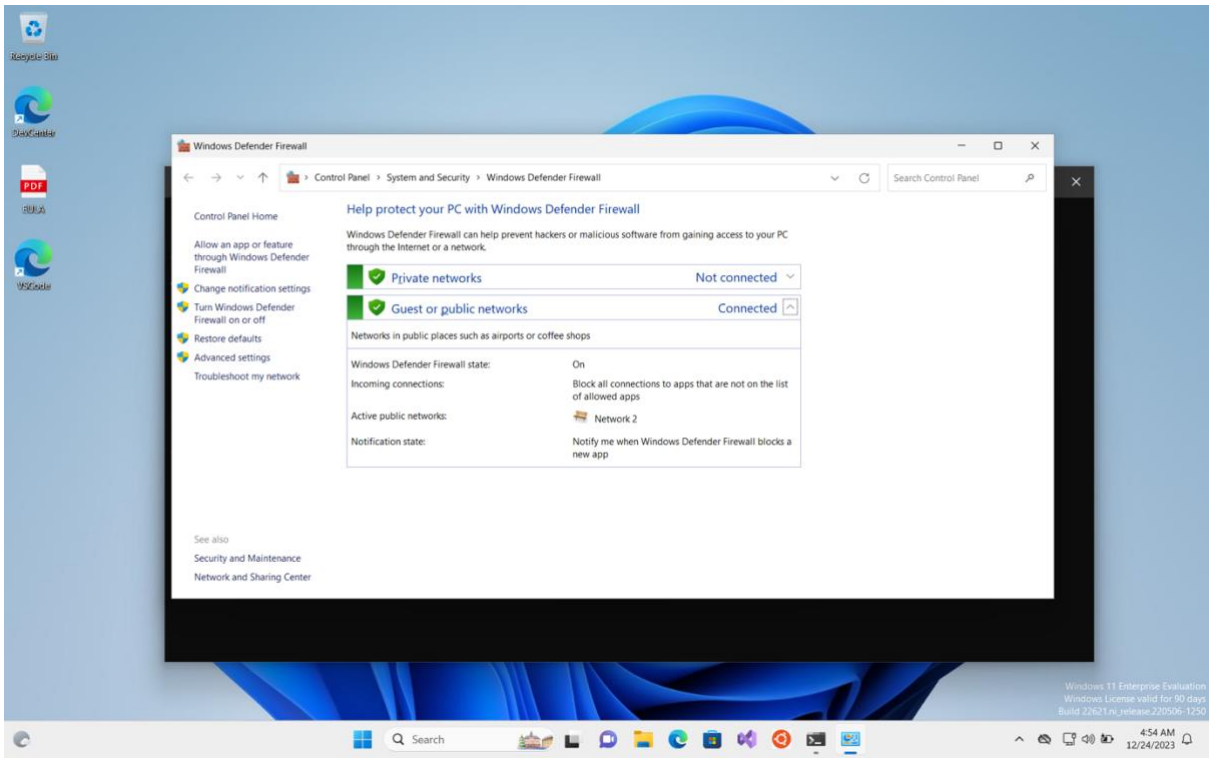

Fig 5.4: Victim Windows Initial Stage Pt.3

As we can see in Fig 5.1 though 5.4, this is the initial stages for both Attacker and Victim when the Hacking starts.

RECONNAISSANCE

We begin by finding open ports on our Victim OS from the Attacker OS using NMAP.

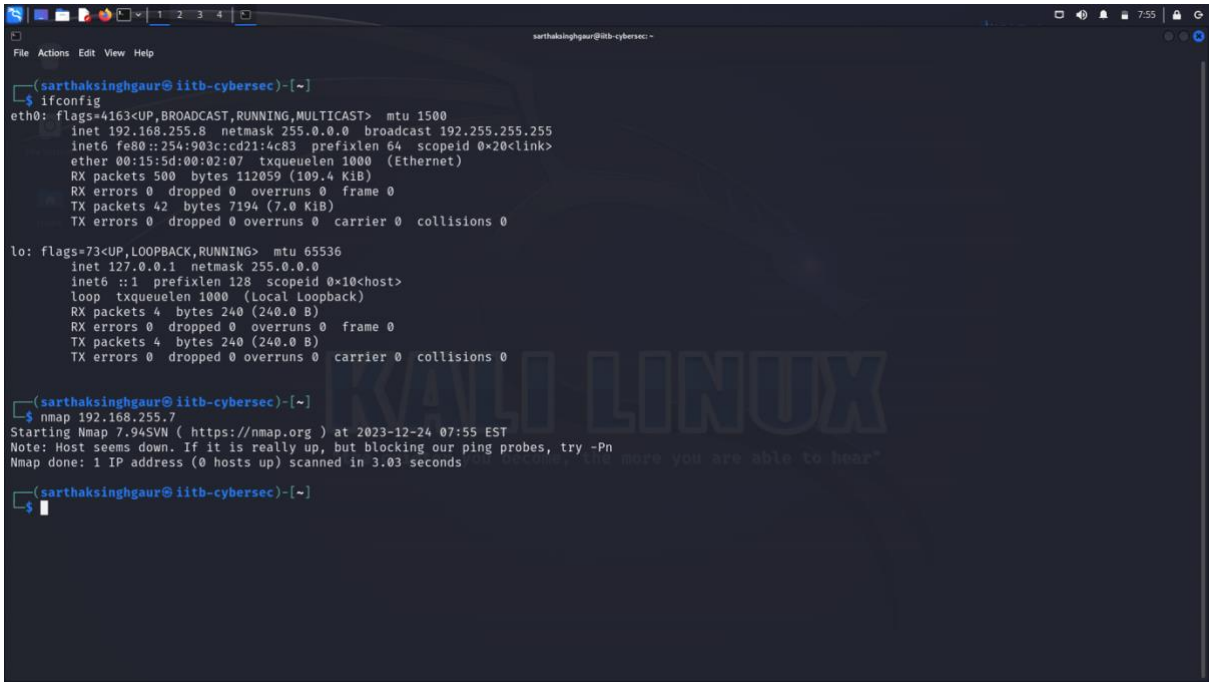

Fig 6.1 : Reconnaissance Pt.1

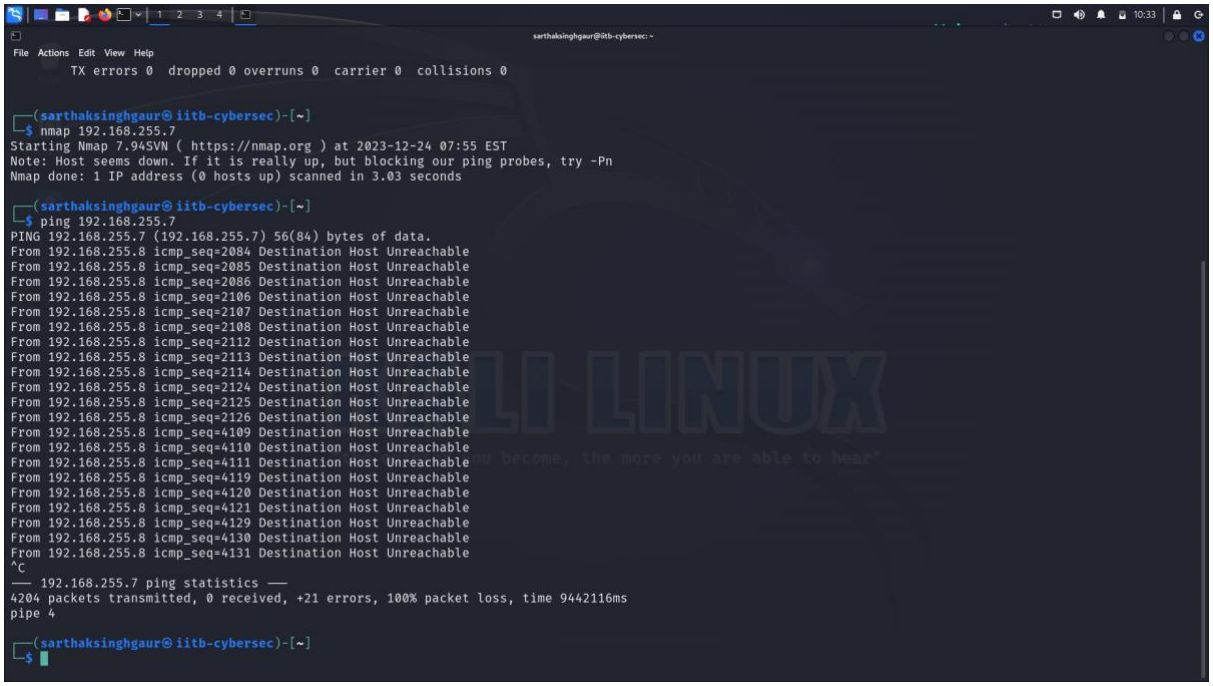

Fig 6.2 : Reconnaissance Pt.2

Since we find the ports blocked, we decide to create a Meterpreter Payload that won't require open ports, rather which will be sent as an Email Attachment and then establish a connection with the Attacker once it is executed.

METERPRETER PAYLOAD GENERATION

Based on the situation, we decide to create Meterpreter Reverse TCP payload on windows architecture (staged). We specify LHOST and LPORT of Attacker's IP inside the payload. We use SHIKATA_GA_NAI encoder and iterate it ten times to ensure smooth bypassing of security solutions. Then we specify the resulting payload's location as shown in Fig 7.1 below.

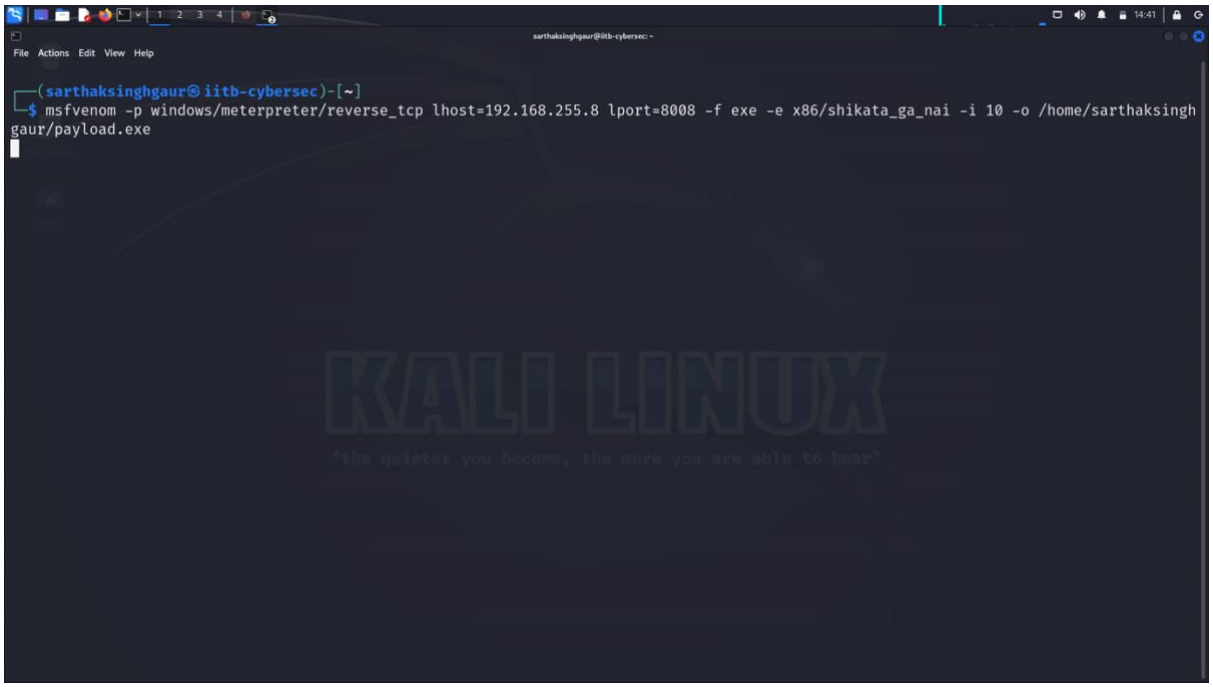

Fig 7.1: Using Msfvenom for payload generation

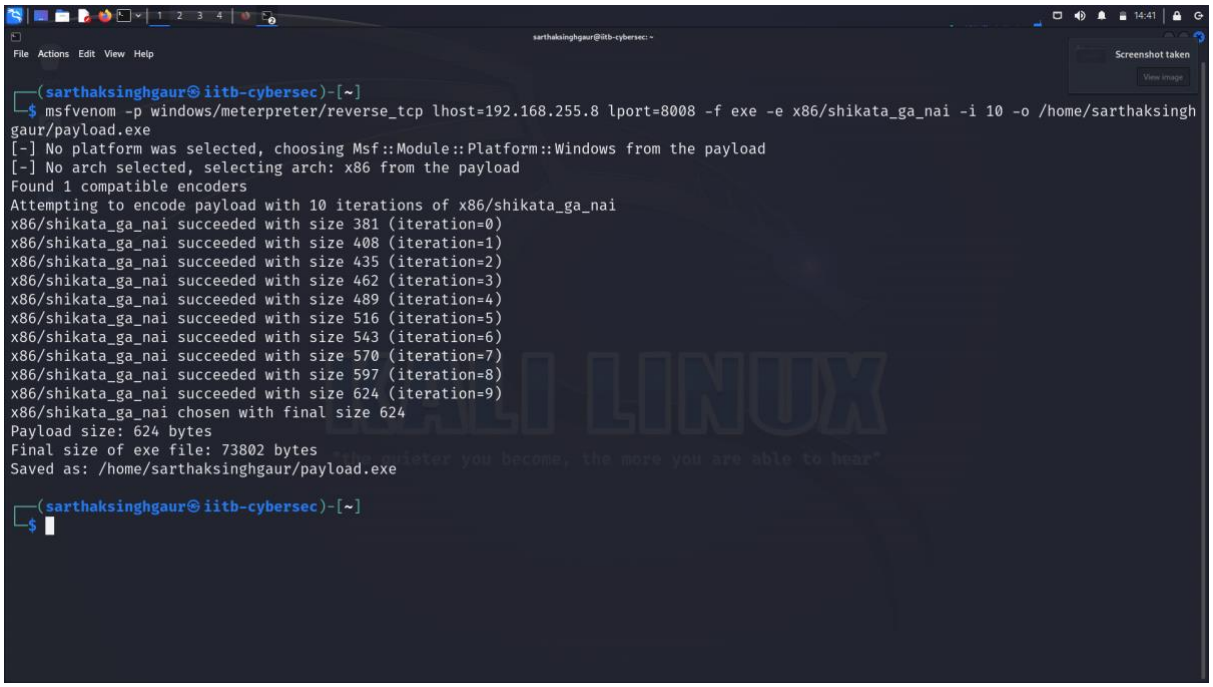

Fig 7.2: Payload generation completed

We get the payload of size 624 bytes, but with ending we end up with 73802 bytes payload.

We then use an online service called VirusTotal, which is an antivirus scanning service.

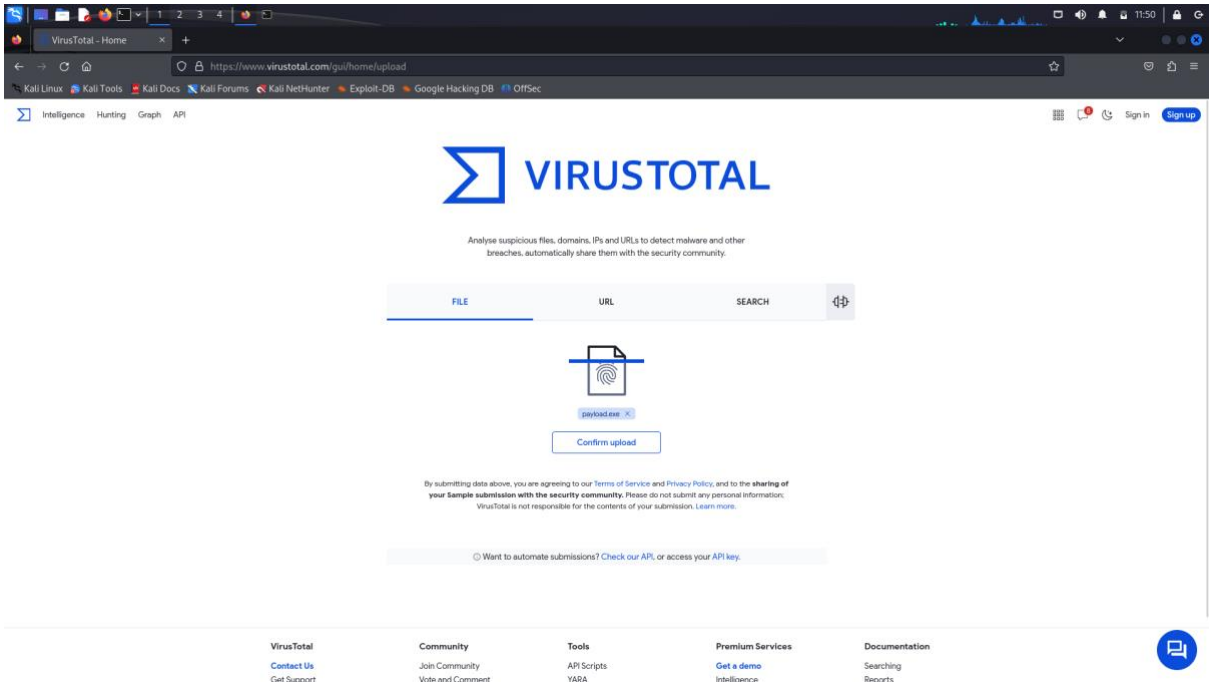

Fig 7.3: VirusTotal payload test

We use it to check if our payload is properly encrypted and bypasses encryption or not. It inspects items with over 70 antivirus scanners and URL/domain blocklisting services, in addition to a myriad of tools to extract signals from the studied content.

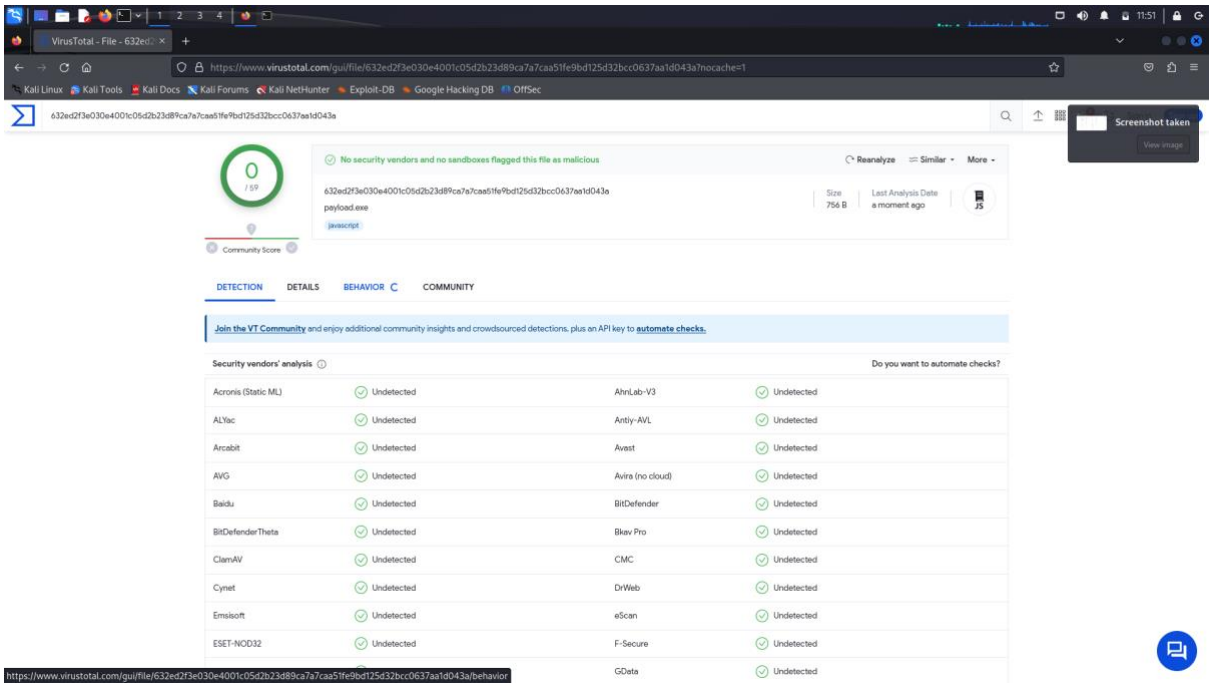

Fig 7.4: Payload Scanning Results

As we can see in figure 7.4, the msfvenom successfully encrypted the payload and it bypasses all 59 security solutions on VirusTotal.

PAYLOAD SPOOFING

In order to deliver the payload to the victim without alerting them, we spoof the '.exe' extension to 'jpg'extension. To remove the .exe from extension, we use a right-to-left-override character to make the file name read from right to left after the right-to-left-override is placed. Right-to-left of .jpg would be gpj.

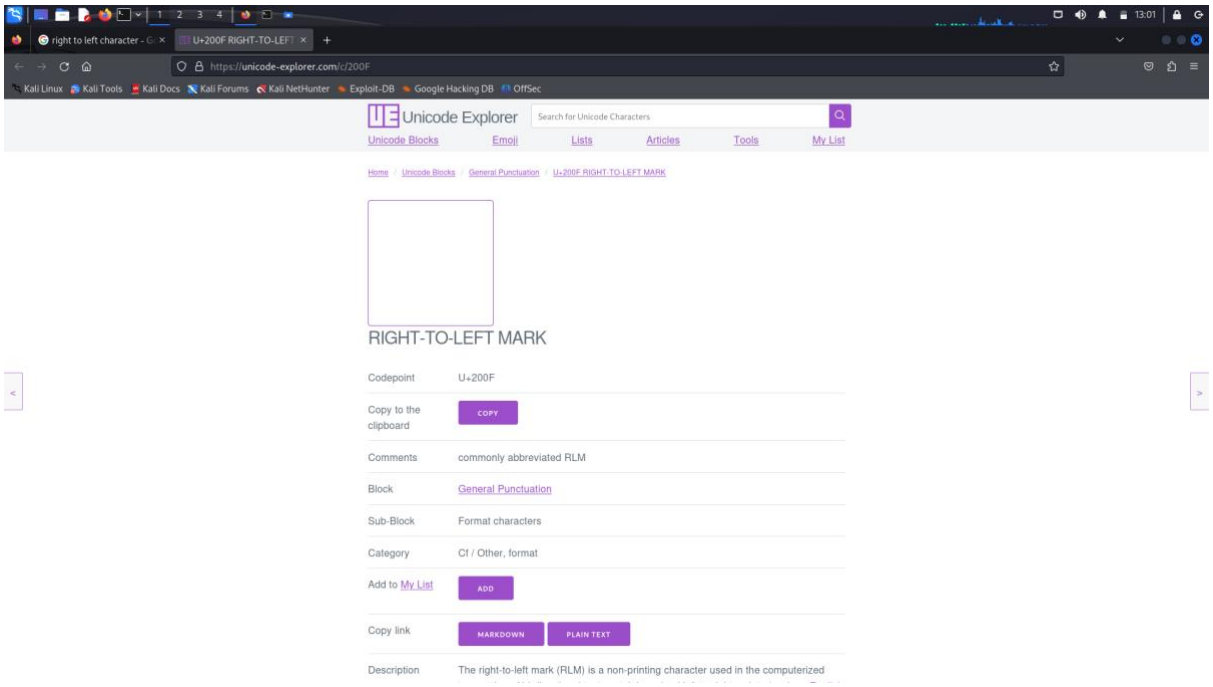

Fig 7.5: Right-To-Left Override Unicode Character

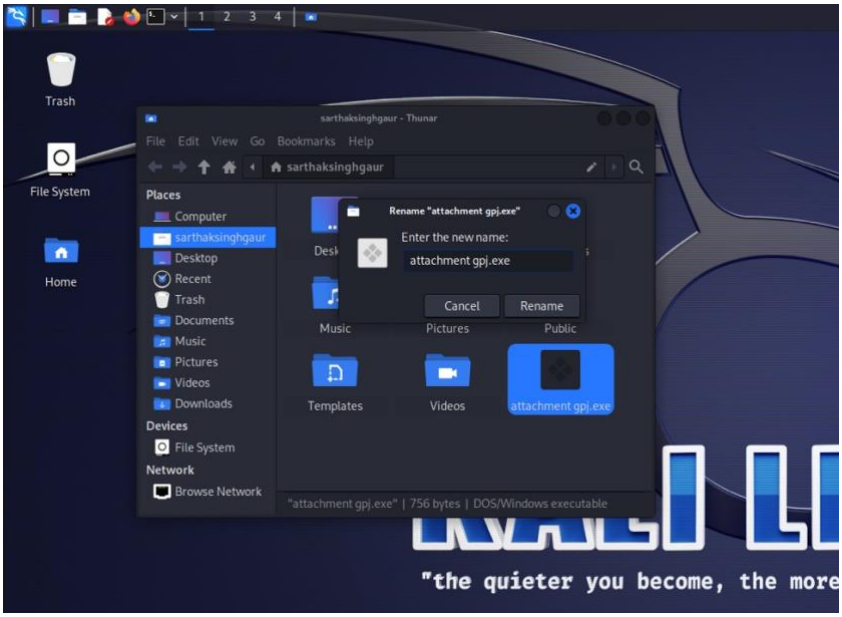

Fig 7.6: Spoofing Payload as Attachment.exe.jpg

In this case we take the filename to be attachment.exe. We rename attachment.exe to attachmentgpj.exe. Paste the right-to-left-override character at the 10th position after attachment. All the characters after the right-to-left-override character will be flipped i.e. read right to left. The filename looks like attachmentexe.jpg. Now we send it to the victim via email.

VULNERABILITY EXPLOITATION

Msfconsole is instantiated on the Kali Linux on attacker's side.

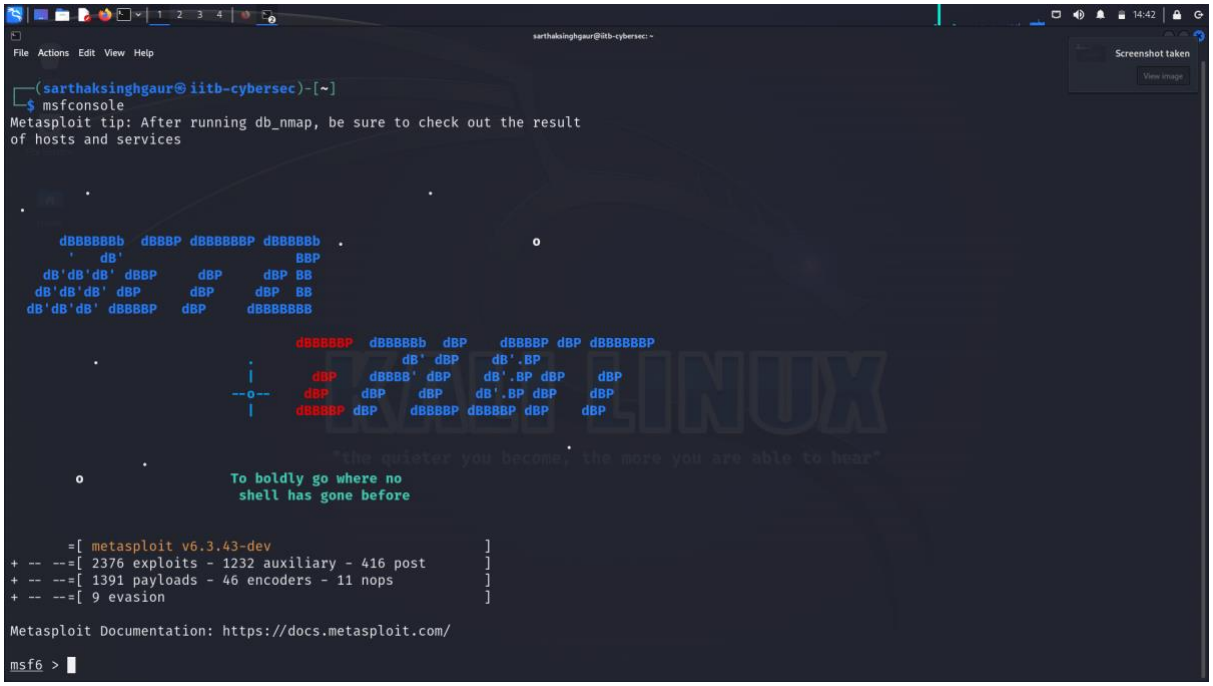

Fig 8.1: Msfconsole execution

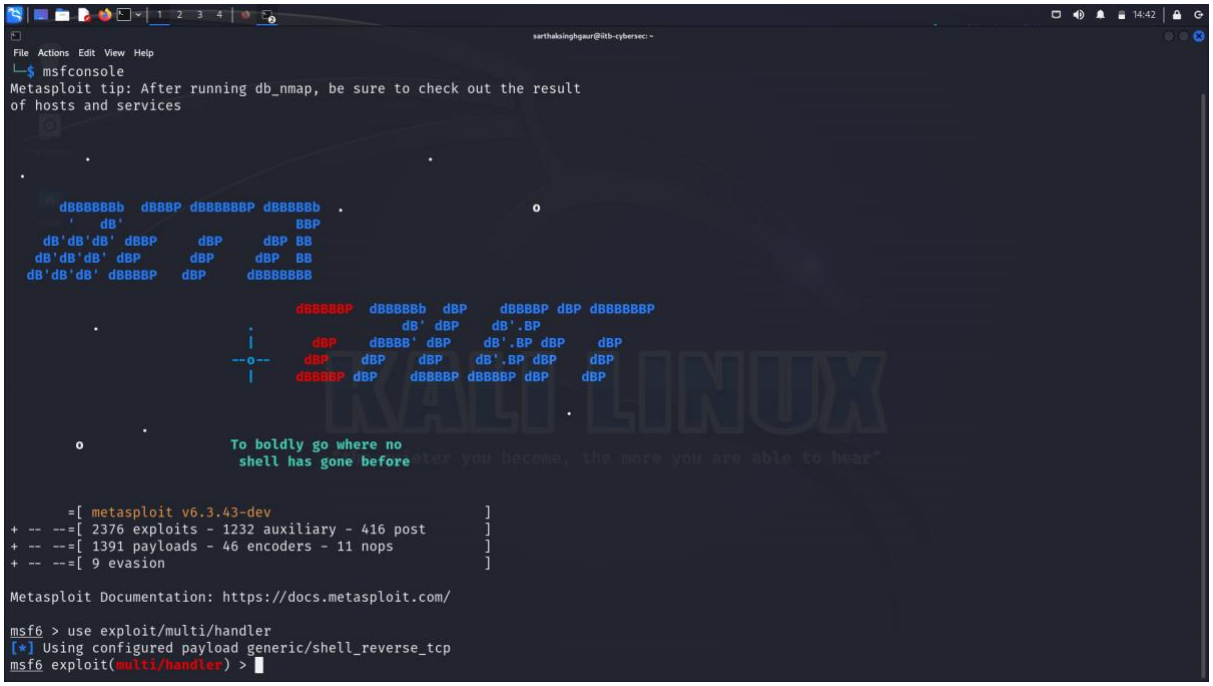

Fig 8.2: Msfconsole handler assigning

Handler for multiple architecture for the purpose of exploitation is used under msfconsole. The cursor changes to exploit(multi/handler) > as shown in Fig. 8.2

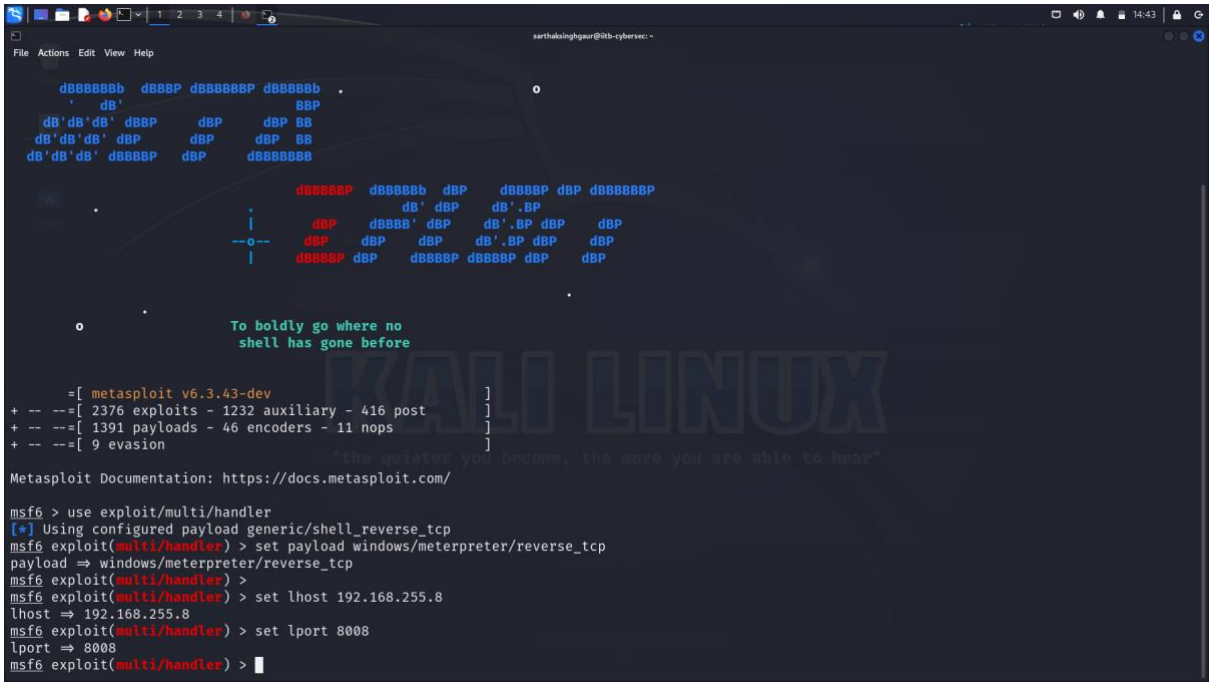

Fig 8.3: LHOST & LPORT under msfconsole

We use the same LHOST and LPORT we used in creating the payload configuration.

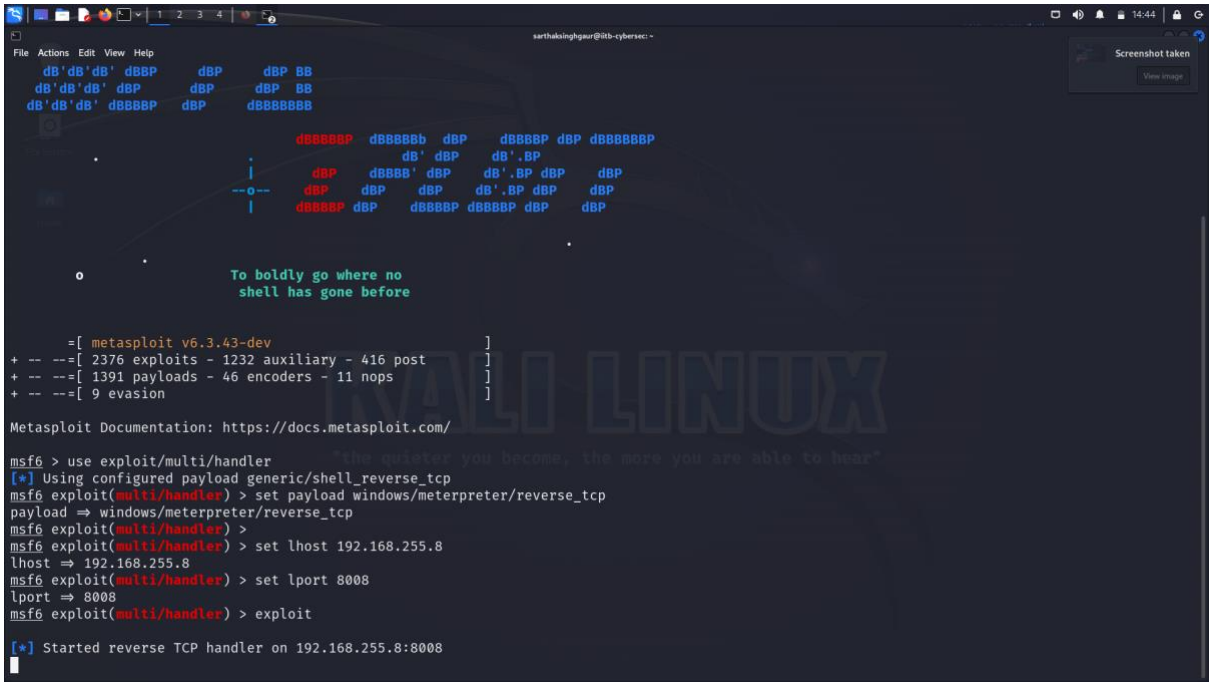

Fig 8.4: Exploitation and Listening under msfcosole

We start the listening process on Metasploit framework for any incoming connection any payload may be trying to establish.

Meanwhile the Victim opens the payload sent to them as attachment in a email.

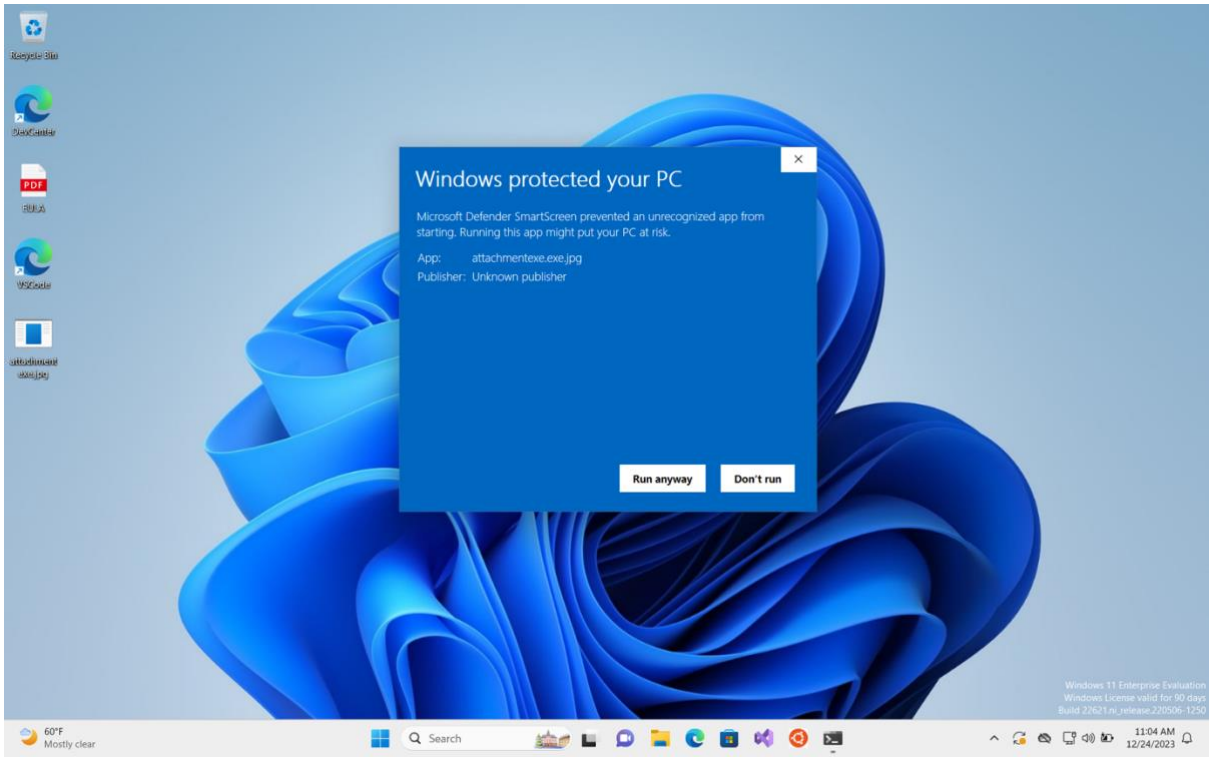

Fig 8.5: Non-Virus Warning on Victim's OS

The victim is greeted with a Non-Virus warning from Windows OS. Since there's no mention of any malware/virus, the victim opens the file in order to look at the attachment sent to them.

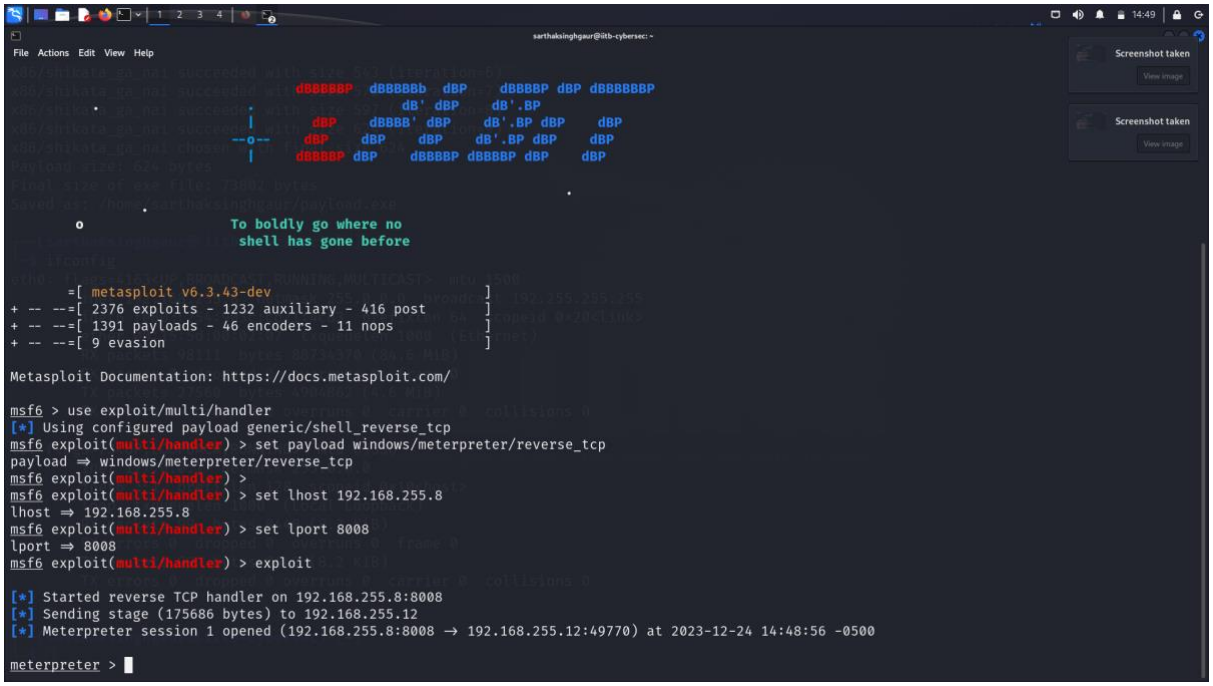

Fig 8.6: Msfconsole capturing incoming connection

As shown in Fig 8.6, Msfconsole captured the incoming connection once the victim opens the payload.

POST EXPLOITATION

Once the exploit has successfully established connection with Attacker, the hacker starts retrieving information and data from the victim.

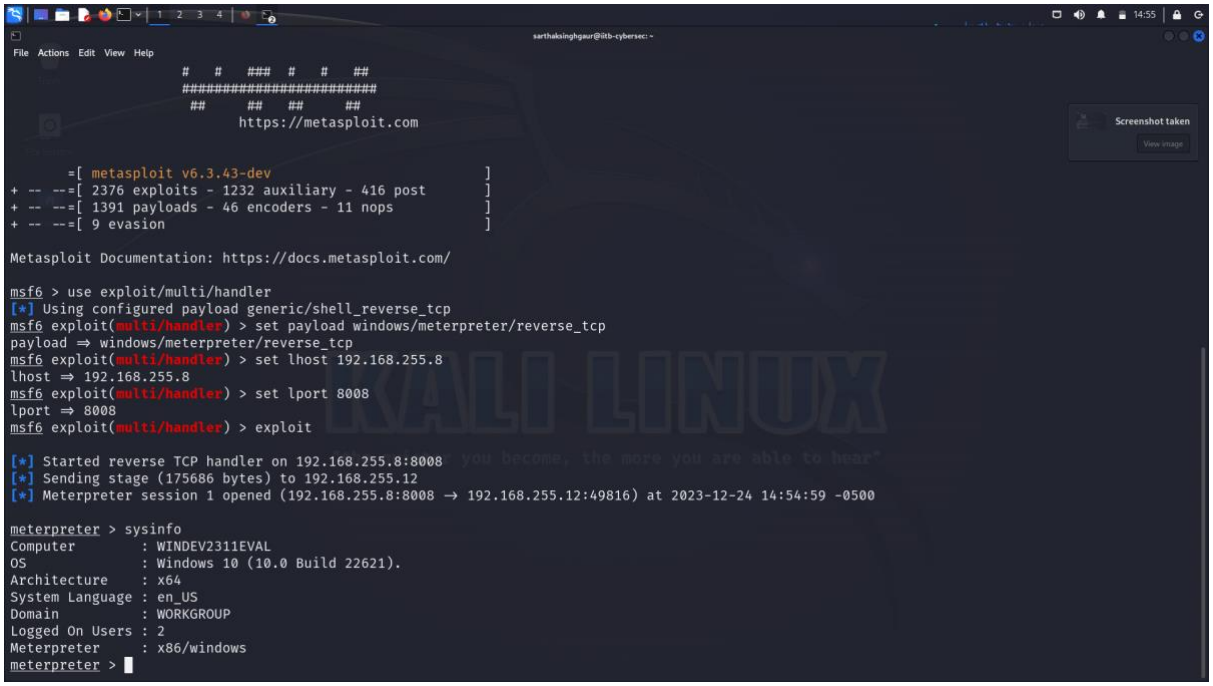

Fig 9.1: System Info retrieved by payload

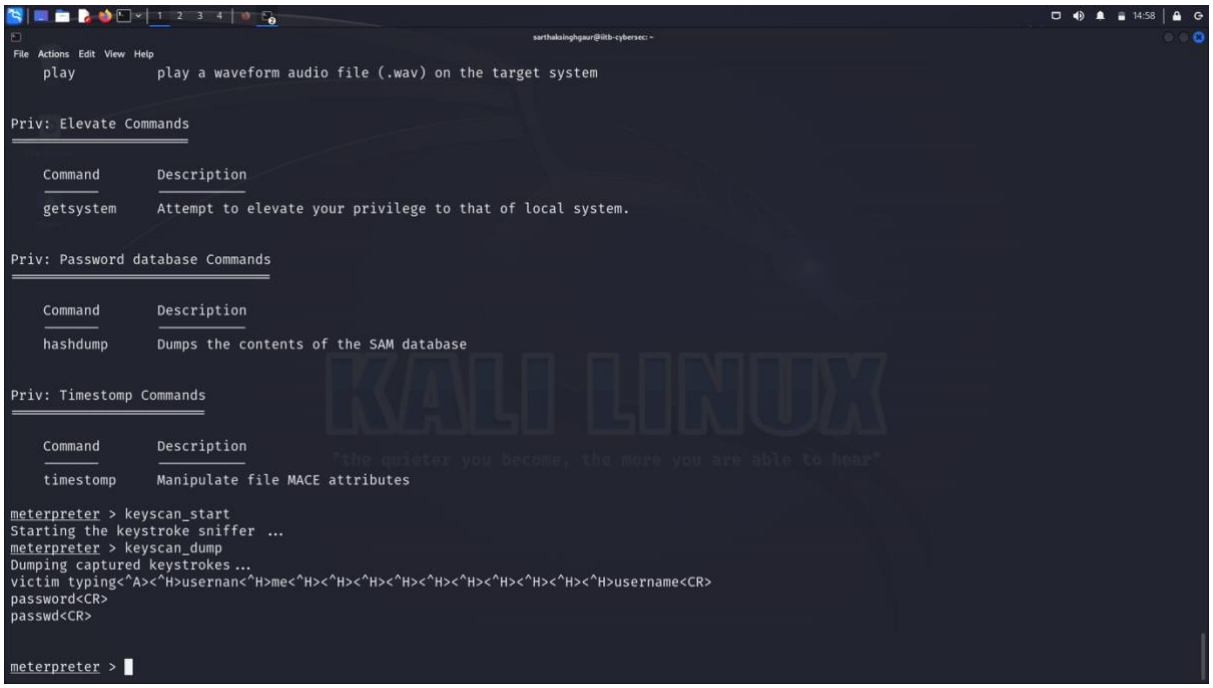

Fig 9.2: KeyScan to record username and password

The Hacker records the keystrokes the victim is making on his/her OS and eventually captures the login username and password for some account using KeyScan as shown in Fig 9.2

PRIVILEGE ESCALATION

The hacker then proceeds to perform Privilege Escalation on Victim's OS. The current session is backgrounded and a new instance is created under msfconsole.

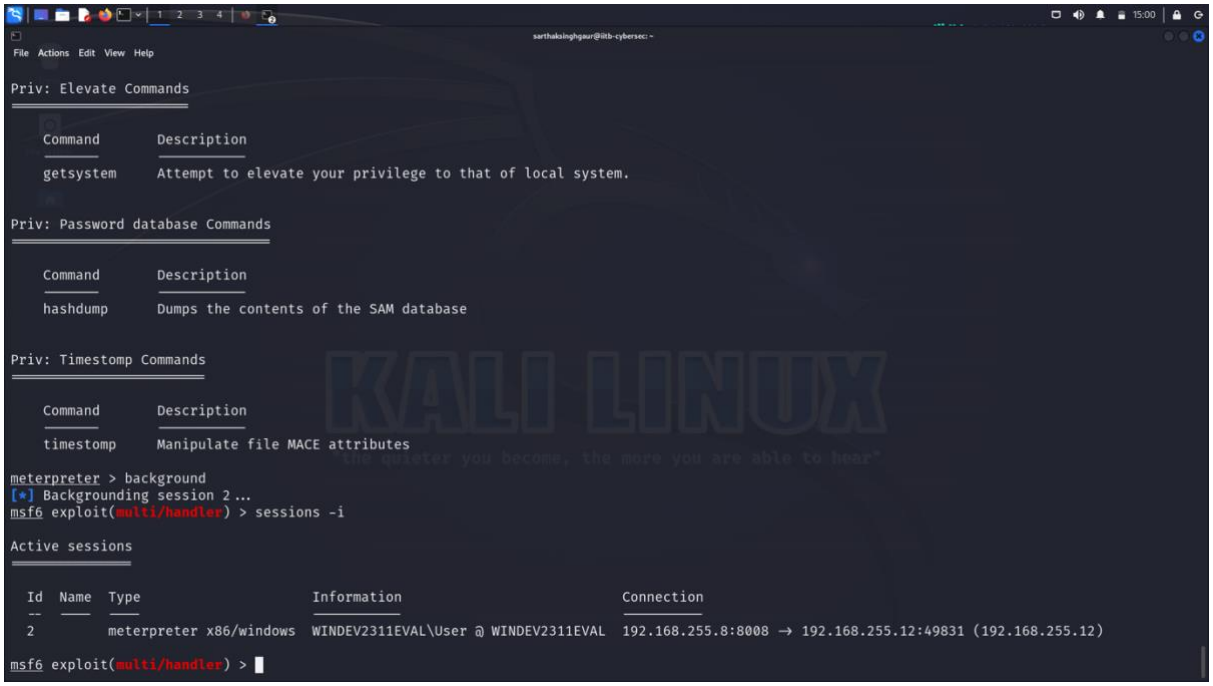

Fig 9.3: Privilege Escalation Pt.1

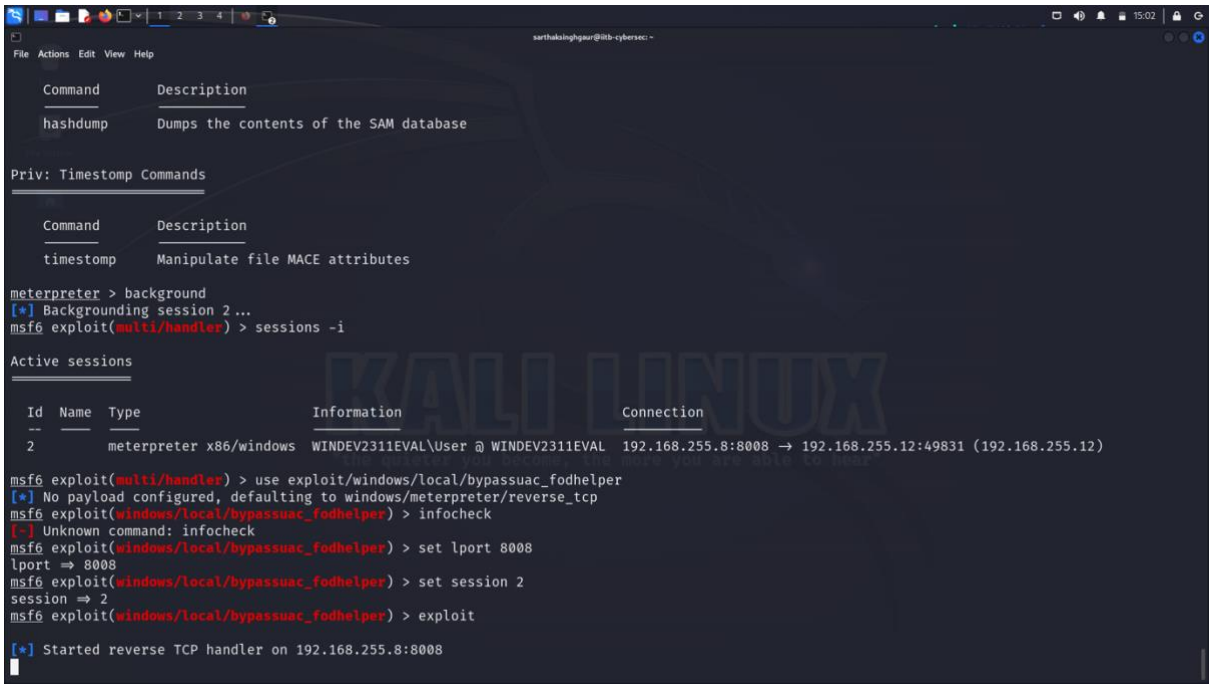

Fig 9.4: Privilege Escalation Pt.2

The hacker uses bypassuac_fodhelper module of Metasploit framework to attempt Privilege Escalation as visible in Fig 9.4

The module finds UAC is enabled and is a part of Administrators group as apparent in Fig 9.5

Fig 9.5: Privilege Escalation Pt.3

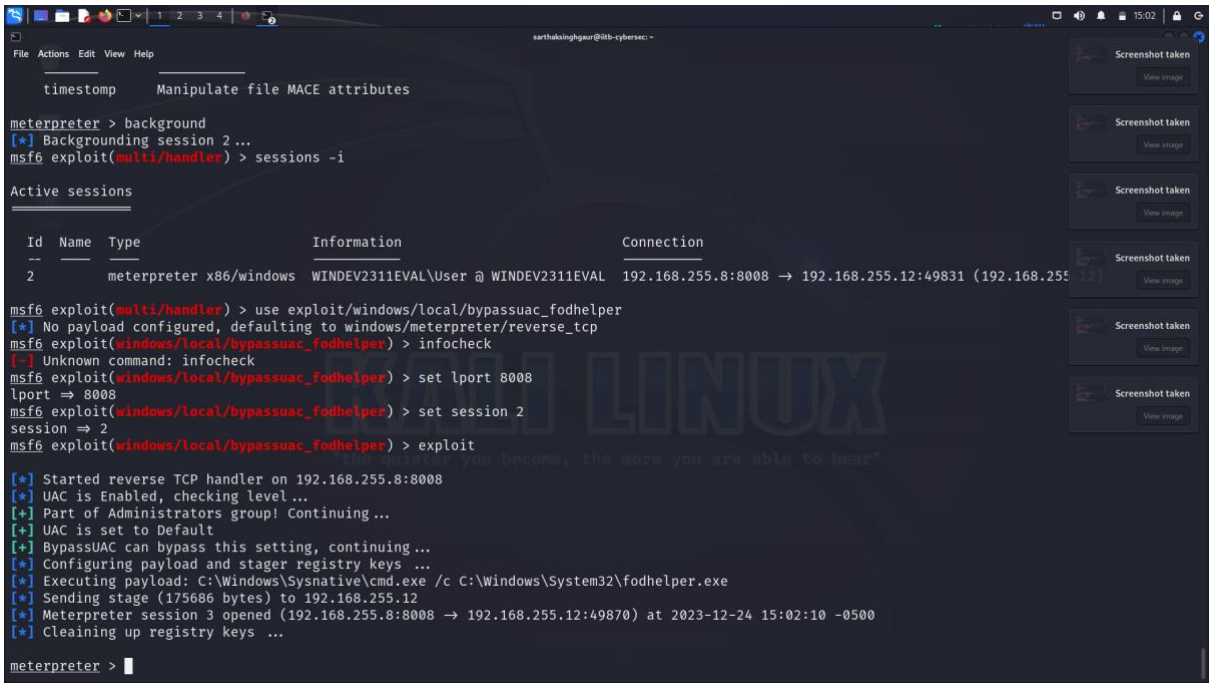

Fig 9.6: Privilege Escalation Pt.4

Msfconsole creates a new session with admin access. The module then cleans up the registry keys used to escalate the access to administrator as shown in Fig 9.6

The hacker verifies the administrator access using getsystem module which returns true.

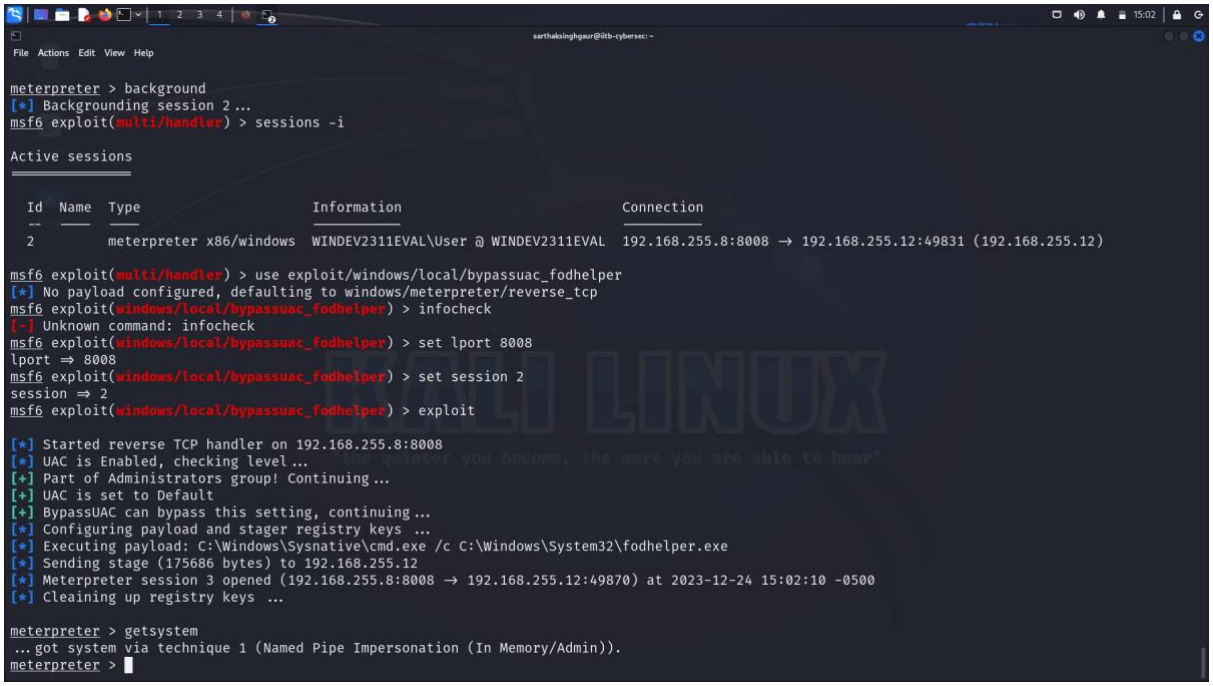

Fig 9.7: Privilege Escalation Verification Pt.1

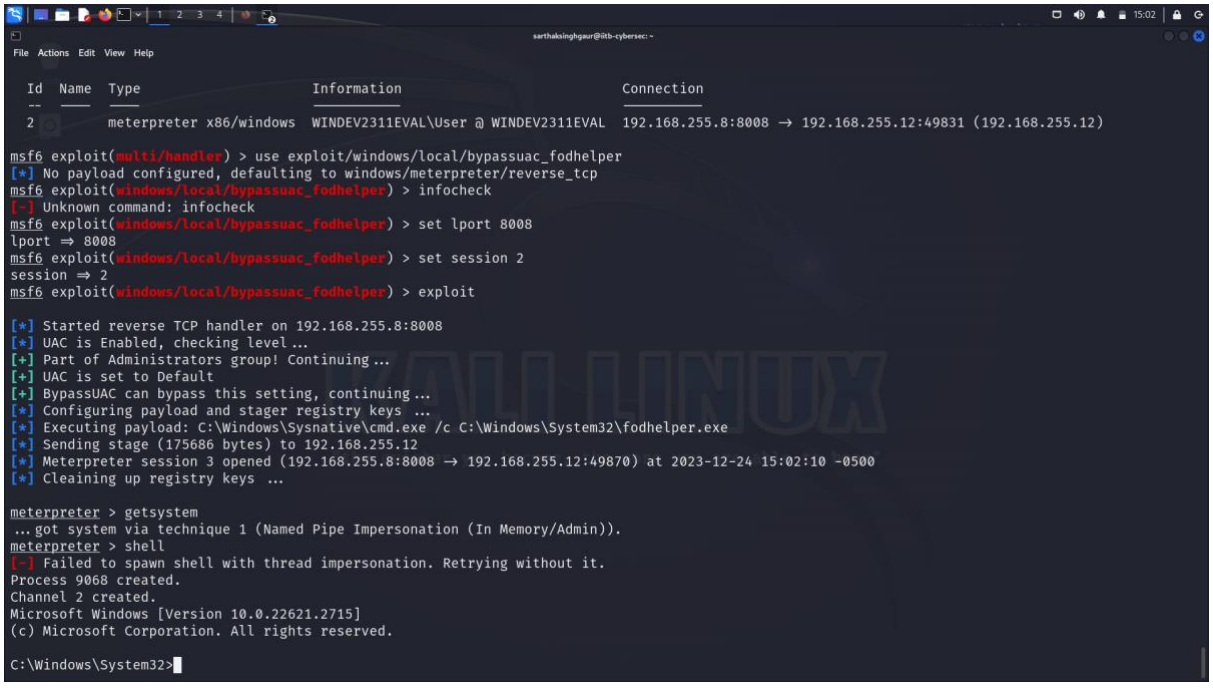

Fig 9.7: Privilege Escalation Verification Pt.2

The hacker then opens victim OS's powershell (command prompt) as administrator on his OS. The shell is created using the module 'shell' and returns with a shell in System32 folder of Victim's OS. (see Fig 9.7)

EXPLOITATION PERSISTENCE

The hacker backgrounds the admin access session and proceeds to create persistence for the payload, so the system is infected even after reboots without reinfecting.

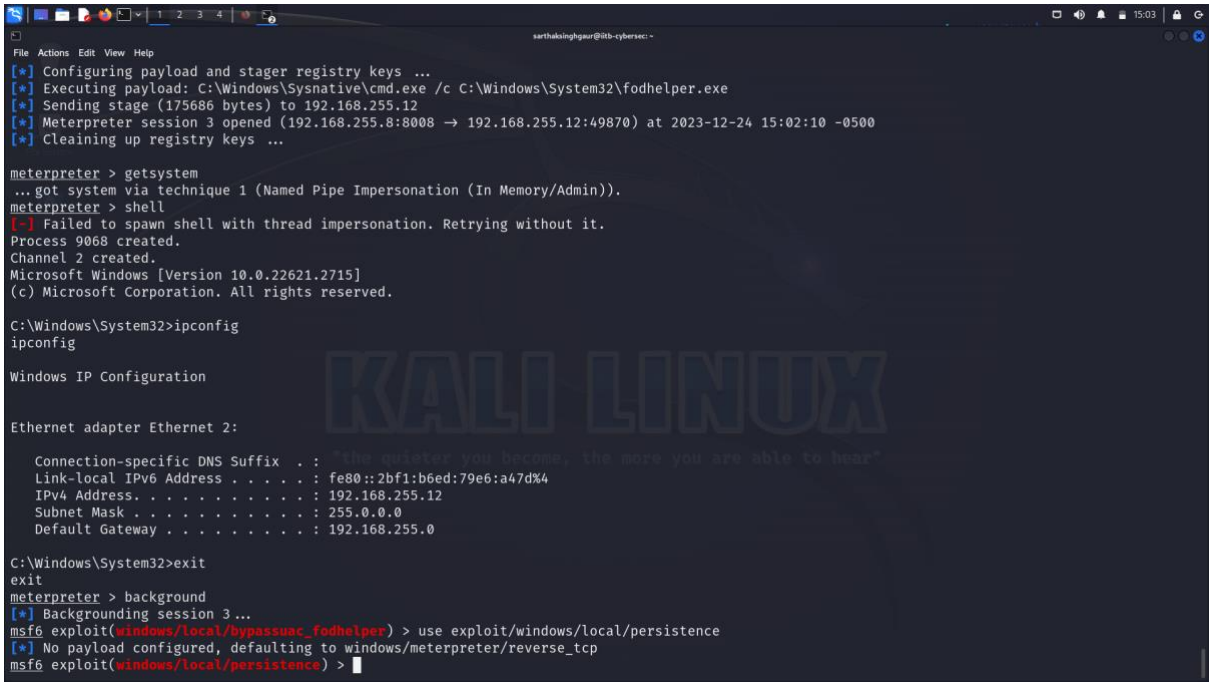

Fig 10.1: Persistence Pt.1

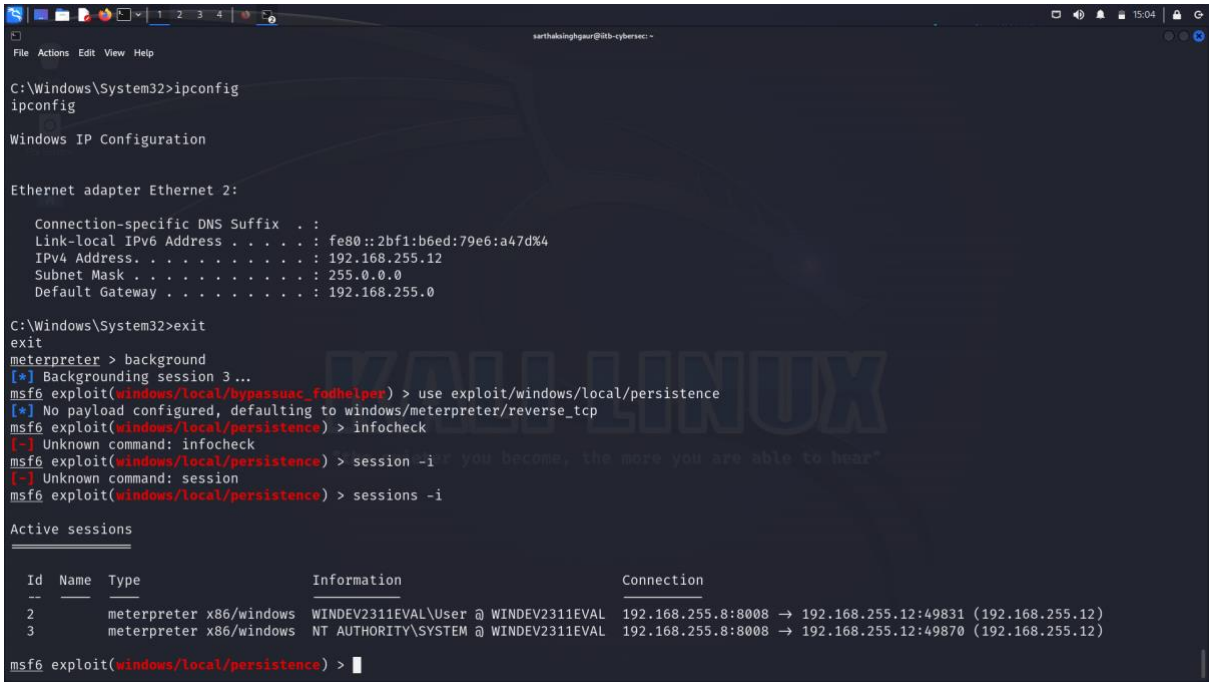

Fig 10.2: Persistence Pt.2

The hacker initialises windows/local/persistence module on msfconsole as visible in Fig 10.2

The admin access session 3 is initialised for creating persistence under msfconsole as shown below in fig 10.3

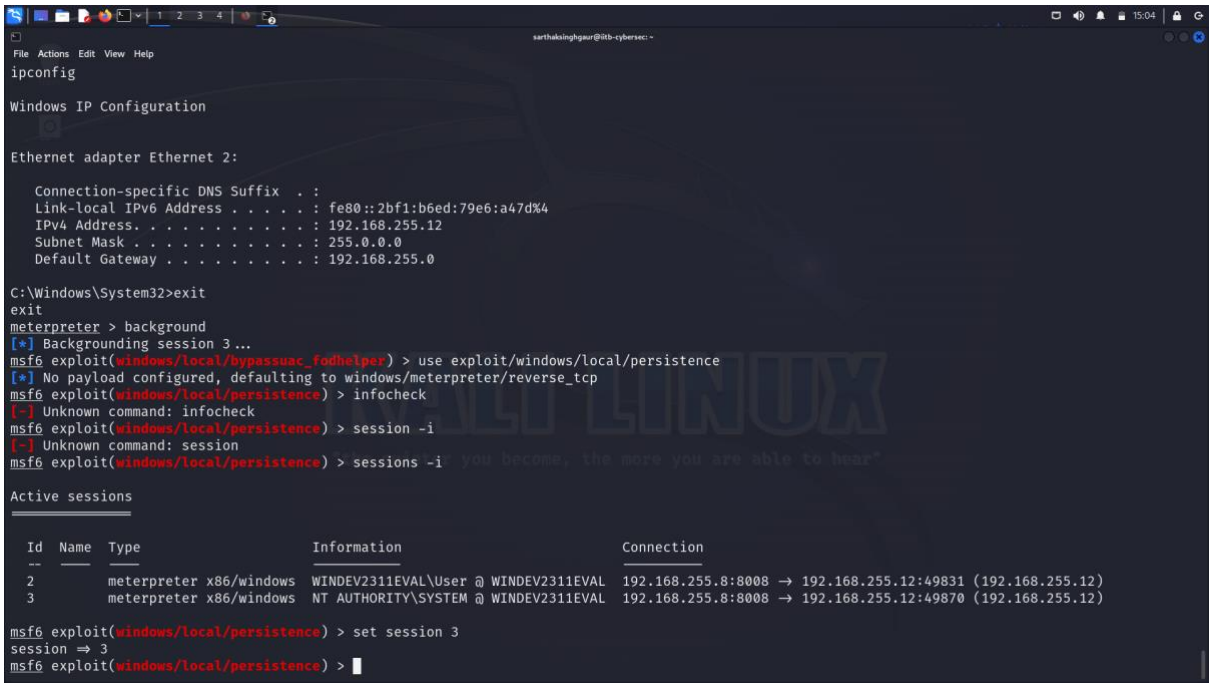

Fig 10.3: Persistence Pt.3

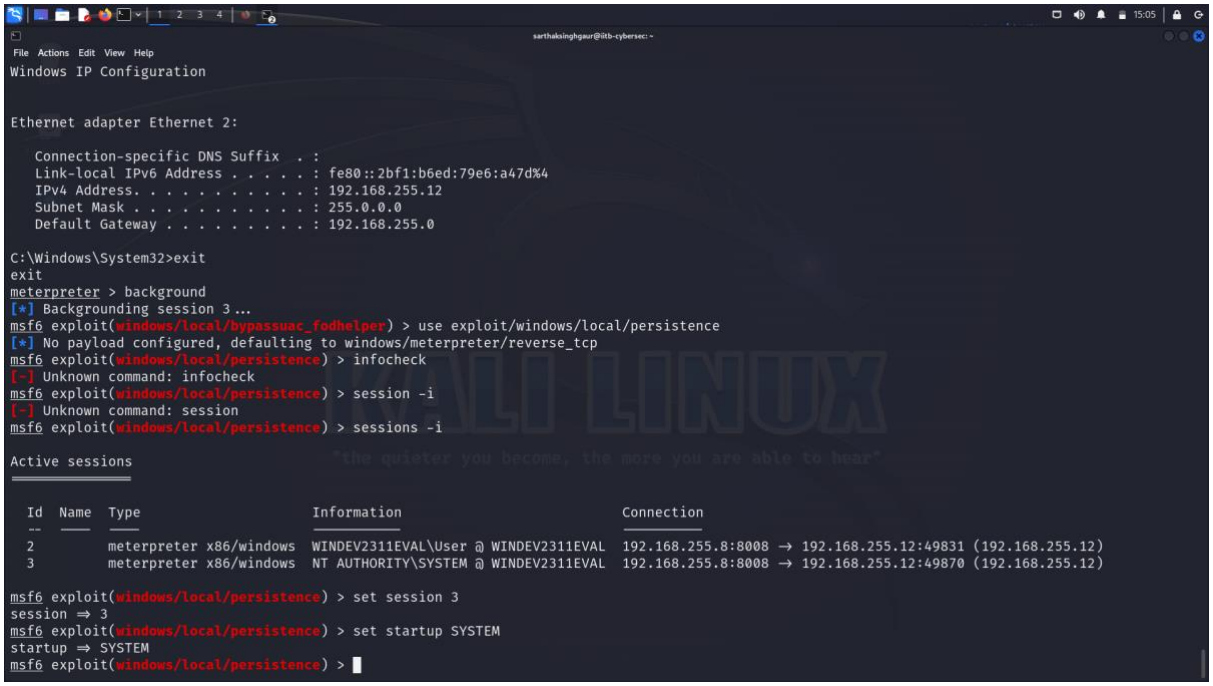

Fig 10.4: Persistence Pt.4

The startup parameter of persistence module is set to SYSTEM (see Fig 10.4)

The module is put to exploitation which then runs Persistence VBS script for the Victim's OS

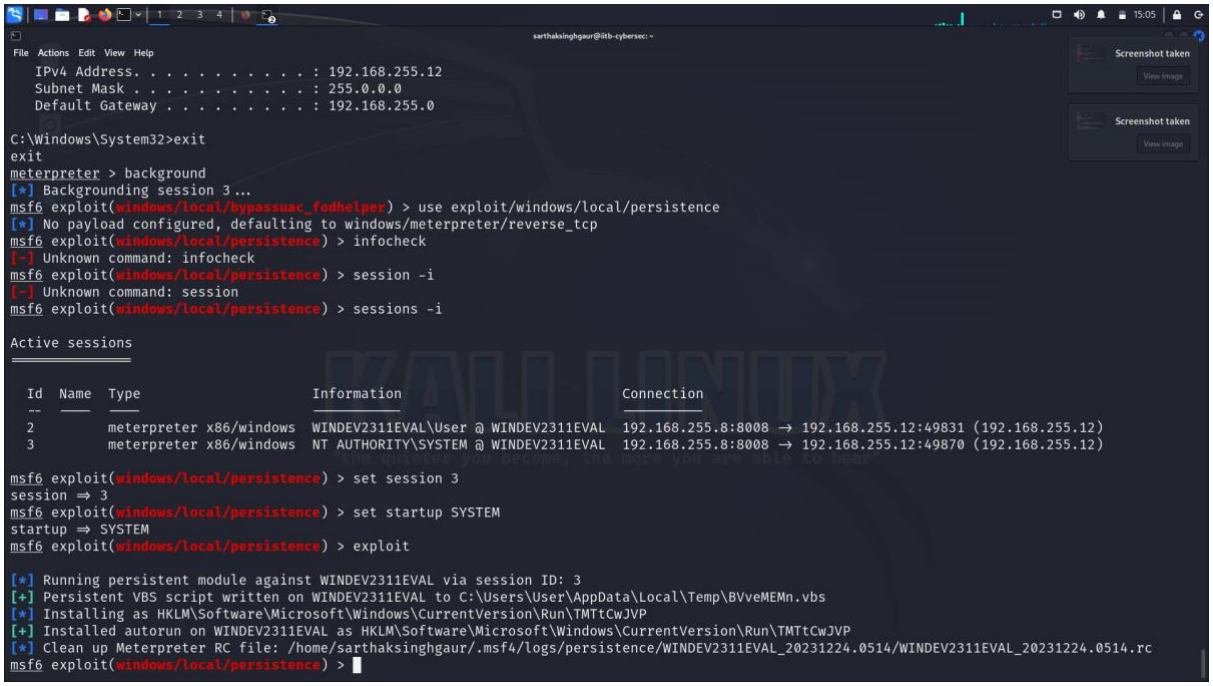

Fig 10.5: Persistence Pt.5

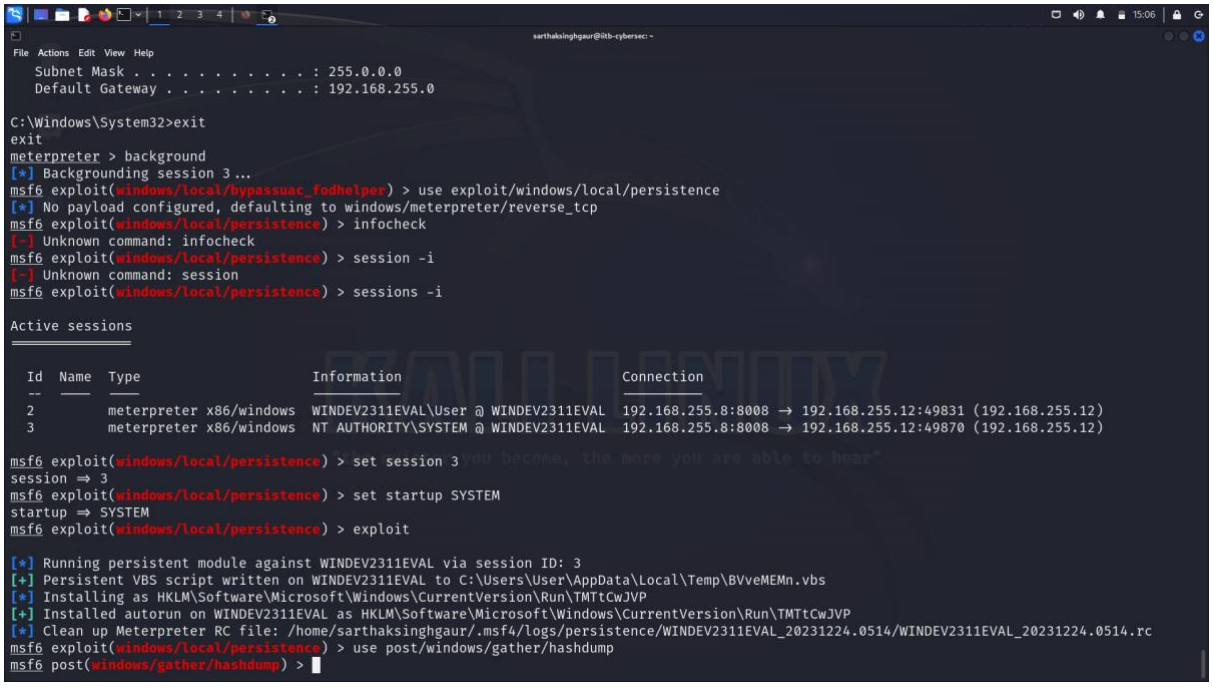

Fig 10.6: Persistence Pt.6

The VBS script installs autorun for the payload on Victim's OS and cleans up the files afterwards. Persistence is established.

HASHDUMP

The hacker initialises ‘post/windows/gather/hashdump’ module in msfconsole.

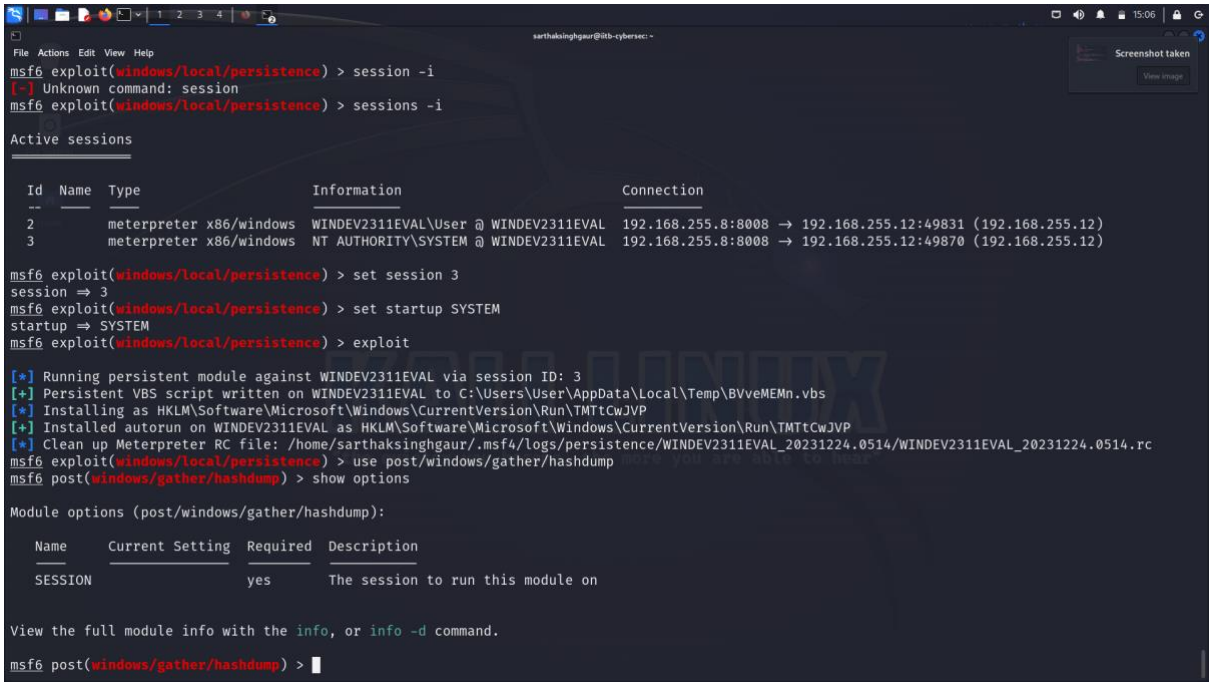

Fig 11.1: Hashdump Pt.1

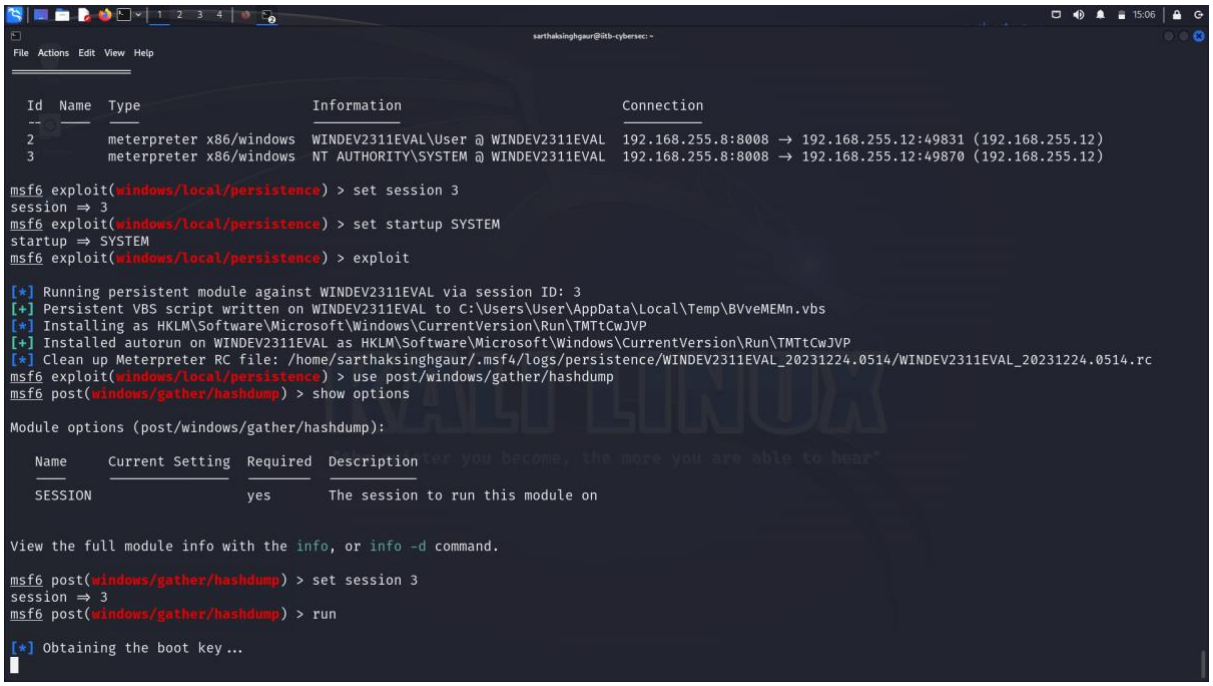

Fig 11.2: Hashdump Pt.2

The only required parameter is set to session 3 which has admin privileges (see Fig 11.2)

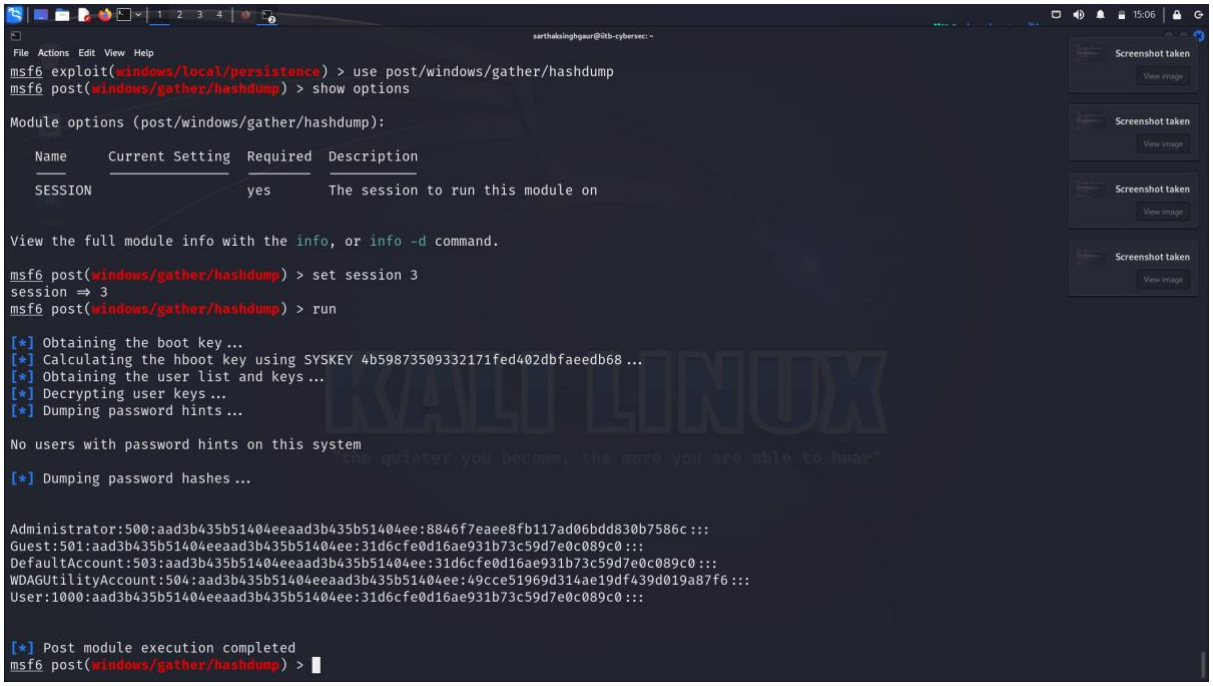

Fig 11.3: Hashdump Pt.3

The module retrieves the SYSKEY for Victim's OS and uses it decode the account username and password hashes stored in the rainbow table of Victim's OS. Eventually the hacker gets the credentials for victim's accounts as well.

RECOMMENDATIONS

This project on windows operating system exploitation revealed some critical issues and the limitless seeming potential these vulnerabilities have. It also highlighted the need for more awareness, cyber security basics and the responsibility on one's own self to keep out of harm's way. These recommendation are given below.

The Windows Operating System is very flexible, but with that flexibility comes potential for unauthorised programmes to enter disguising themselves as legitimate processes. If left at it's default configuration, the security policies of Windows leaves a lot to be taken advantage of. One must be very cautious while using it and it is recommended to change all the default setting to make it more secure. Closing useless and unnecessary ports, strengthening security policies, restricting firewall accesses to fewer applications, keeping stronger passwords that are hard to guess, not reusing your credentials on more than one website or platform, all these changes will help mitigate the inherently vulnerable nature of windows operating system.

The fact that once someone gains unauthorised access to a windows device, he/she can do virtually everything that the actual user can do on the device is a wakeup call for everyone who is oblivious to cyber security and the risks they are bearing. In the project itself, it was possible to record the keystrokes of the victim on the device and download hashes of account credentials, but it is not limited to anything less. Personal files are easily accessible, browser files can be stolen, camera and mic access can be captures, potentially spying on you 24/7 via webcam, mic, and what you do online be it banking or personal messaging. There's a saying in cyber-security, **"Trust, but verify"** One should follow this to the heart if they want to be safe in today's cyber would. Never trust blindly, be it websites, platforms, or unknows people. Always verify first, then trust them.

Such dangers call for the individual to take charge of their own cyber safety, rather than relying on operating system to get better. Cyber-conscious choices and wisdom will immensely protect one from such disasters. Not opening unknown links, not installing software from unverified sources, keeping their antivirus and operating systems updated with the latest security patches, keeping an eye open for unexplained OTPs, not logging in on shared or risky devices, logging out of such devices as soon as the task finishes, not leaving your devices unlocked all the time, not just for you but for those around you as well, all this will be the steps in the right direction.

CONCLUSION

This Windows OS Exploitation project has been more than a project, it has been an invaluable eye opening experience which highlighted the importance of cyber security and personal responsibility in today's technologically evolving world. The prime scope of this project is very admirable and gives out much needed knowledge.

Cyber Security has never been more necessary than now. We spend most of our life oblivious to the potential cybercrime we all are susceptible to. We hear news now and then about cybercrime happening throughout the world, but we never think that it might be your turn next if you didn't changed your perception. We need to become "cyber-conscious" which means to be cautious and be aware of the possibility of cyber crime around us. This project depicted how easy it is for hackers to gain unauthorised access of Windows OS and then do any kind of damage to whoever owns that device, and it doesn't stop at just financial loss, it can go up to identity theft, spying, embarrassment and leaking of private life online.

Despite it being very easy for hackers to gain access, it also shows if one looks closely, they can see the loopholes and security flaws and stop them. It doesn't take much to close these gaps in security. Windows, despite being flexible, does provide measures to prevent unauthorised access to it. All it requires is a vigilant sight, and the realization of the potential dangers these flaws contain within them. Not connecting to free, unsafe Wi-Fi, removing outdated and old software, never reusing passwords and PINs on multiple accounts, creating long passwords, keeping operating systems and antiviruses up to date, keeping your devices away from suspicious people and surroundings, being aware of latest cybersecurity practices, etc, are all small habits that will go a long way if one tries. Only then perhaps one day, we can hope to live a cybercrime free, carefree life. But until then, vigilance and awareness is the key.