ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

CDS201: Έλεγχος Εισβολών Δικτύων και Συστημάτων (Penetration Testing)

Privilege Escalation

Host Privilege Escalation Intro

- Results in elevating privileges from that of a standard user to an Administrative one (member of Local Administrators Group or the NT AUTHORITY\SYSTEM local system Account.
- A vital step during any engagement. However It’s not a necessary step in order to obtain privileged credentials and move laterally in a domain (AD) during an engagement.
- However, elevated privileges can provide a tactical advantage by allowing you to leverage some additional capabilities. For example:
  - dumping credentials (like with Mimikatz) or obtain credentials to move laterally
  - gain a foothold into the client's Active Directory environment
  - installing sneaky persistence
  - manipulating host configuration (such as the firewall).

Host Privilege Escalation Intro

Some of the methods we can use:

- Abusing Windows User/Group privileges
- Abusing Weak Service/File Perms
- Bypassing UAC
- Stealing credentials
- Unpatched kernel exploits
- ….

Host Privilege Escalation Tools

https://book.hacktricks.xyz/windows-hardening/checklist-windows-privilege-escalation

Tool	Description
Seatbelt	Seatbelt is an enumeration tool written in C#. It contains a number of enumeration checks. It does not actively hunt for privilege escalation misconfigurations, but provides related information for further investigation.
winPEAS	A very powerful tool that not only actively hunts for privilege escalation misconfigurations, but highlights them for the user in the results.
PowerUp / SharpUp	PowerShell / C# script for finding common Windows privilege escalation vectors that rely on misconfigurations.
SessionGopher	Powershell tool that extracts PuTTY, WinSCP, SuperPuTTY, FileZilla, and RDP saved session information
Watson	a .NET tool that enumerates missing KBs and suggests PrivEsc exploits for related vulns
Windows Exploit Suggester - Next Generation	
Sysinternals Suite	AccessChk, PipeList, and PsService can be used for enumeration

Host Privilege Escalation
Seatbelt

- Seatbelt is an enumeration tool. It contains a number of enumeration checks.
- It does not actively hunt for privilege escalation misconfigurations, but provides related information for further investigation.

Host Privilege Escalation Intro

- Staying focused in the "principle of least privilege" - privilege escalation steps should only be taken if it provides a means of reaching your goal, not without meaning. Because exploiting a privilege escalation vulnerability provides defenders with additional opportunities to detect your presence. It must worth the risk.
- Common methods for privilege escalation include Operating System or 3rd party software misconfigurations (and missing patches).

Host Privilege Escalation Situational Awareness

When we land on a Windows or Linux system and we want to escalate privileges next, there are several things we should always look for in order to plan our next moves:

- Are there any other hosts that we can directly access?
- Are there any protections in place (to be bypassed)?
- Which tools will not work against the system?

Host Privilege Escalation Situational Awareness

- Network Info:
  - Interface(s), IP Address(es), DNS Information
    - C:/> ipconfig /all
    - C:/> arp –a
    - C:/> route print
- Protections: AVs/EDRs are a challenge with PrivEsc especially when using public exploit PoCs or tools. Enumerating the protections in place can help us modify our tools in a lab environment and test them before using them against a system.
  - PS C:\> Get-MpComputerStatus
  - Enumerate and test AppLocker policies in place
    - PS C:\> Get-AppLockerPolicy -Effective | select -ExpandProperty RuleCollections
    - PS C:\> Get-AppLockerPolicy -Local | Test-AppLockerPolicy -path C:\Windows\System32\cmd.exe -User Everyone

Host Privilege Escalation Initial Enumeration

We can escalate our access level to:

- NT AUTHORITY\SYSTEM – highly privileged local system account used to run most Windows Services
- Local Administrator – the built-in local admin. It is seen reused across multiple systems
- Local account member of built-in Local Administrators group
- A domain user account who is a member of the Local Administrators group
- A domain Admin who is a member of the Local Administrators group

Host Privilege Escalation
Initial Enumeration

System Information: OS version, installed programs, security updates ..to hunt for missing patches of CVEs

- C:\> tasklist /svc - display running services
- C:\> set - search for PATH variables (for DLL hijacking candidates) and possible shares
- C:\> systeminfo - which patches are applied (search for HotFixes)
  - Also with C:\> wmic qfe - WMI using the WMI-Command binary with QFE (Quick Fix Engineering) or
  - PS C:\> Get-HotFix | ft –AutoSize - using the PowerShell Get-Hotfix cmdlet
- C:\> wmic product get name - installed programs
  - PS C:\htb> Get-WmiObject -Class Win32_Product | select Name, Version - using Get-WmiObject PowerShell cmdlet
- PS C:\> netstat –ano - display listening ports for services. We may find a local host only accessible vulnerable service – often overlooked, assumed secure enough under localhost

Host Privilege Escalation
Initial Enumeration

User/Group Information

- C:\> query user - what users are logged in, what they are working on…
- C:\> whoami /priv - get current user privs
- C:\> whoami /groups - is the current user member of other AD groups?
- C:\> net user - get all users
- C:\> net localgroup - enumerate all non-standard groups to determine what the host is used for or search for misconfigurations in group memberships (Domain Users in Remote Desktop Group or Local Admins group)
- C:\> net localgroup administrators - details about a group
- C:\> net account - password policy and account information

Host Privilege Escalation
Initial Enumeration

User/Group Information

- C:\> query user - what users are logged in, what they are working on…
- C:\> whoami /priv - get current user privs
- C:\> whoami /groups - is the current user member of other AD groups?
- C:\> net user - get all users
- C:\> net localgroup - enumerate all non-standard groups to determine what the host is used for or search for misconfigurations in group memberships (Domain Users in Remote Desktop Group or Local Admins group)
- C:\> net localgroup administrators - details about a group
- C:\> net account - password policy and account information

Host Privilege Escalation Initial Enumeration

Process Communication with Named Pipes

- Pipes are essentially files stored in memory that get cleared out after being read

Host Privilege Escalation Windows User Privileges

- Privileges : rights that an account can be granted to perform a variety of operations on the local system.
- User and group privileges are stored in a database and granted via an access token when a user logs on to a system.
- Each time a user attempts to perform a privileged action, the system reviews the user's access token to see if the account has the required privileges

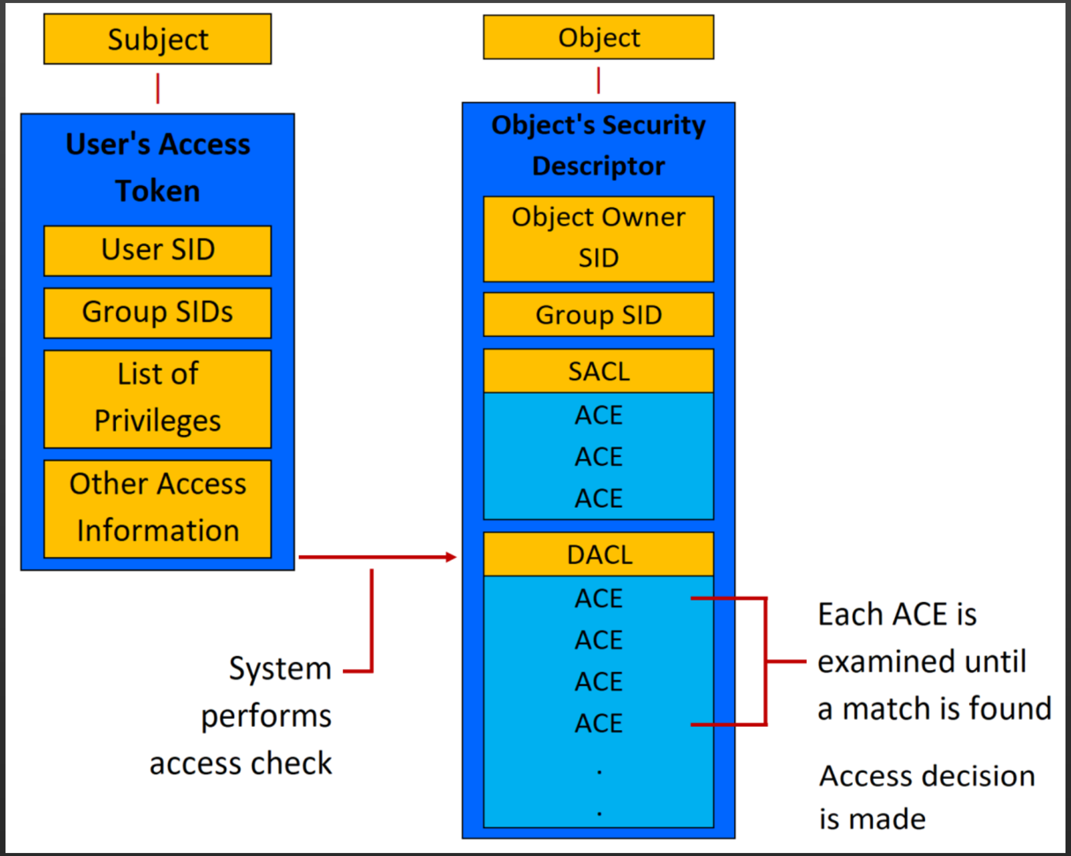
The goal of an assessment is often to gain administrative access to a system or multiple systems.

Host Privilege Escalation Windows User Privileges

Security principals: anything that can be authenticated by the Windows OS (user account, computer account, group, processes running under the security context of another user) – identified by a unique SID

Access tokens: An access token is a protected object that contains information about the identity and user rights that are associated with a user account.

During the Windows authorization process the user's access token (including their user SID, SIDs for any groups they are members of, privilege list, and other access information) is compared against Access Control Entries (ACEs) within the object's security descriptor (contains security information about a securable object such as access rights granted to users or groups)



https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-principals

Host Privilege Escalation Windows User Privileges

- Users can have various rights assigned to their account based on their group membership. These rights allow users to perform tasks on the system such as logon locally or remotely, access the host from the network, shut down the server, etc.
- Privileges can also be assigned via domain and local Group Policy.
- Some rights are only available to administrative users and can only be listed/leveraged when running an elevated cmd or PowerShell session.
- As part of enumeration and privilege escalation activities, we attempt to use and abuse access rights and to further our access towards our goal.

https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/user-rights-assignment

Host Privilege Escalation
Windows User Privileges

SeImpersonatePrivilege

- Every process has a token that has information about the account that is running it.
- SeImpersonate privilege is needed in order to utilize such tokens. It is only given to Administrative Accounts.
- Many legit programs utilize the token of another process to escalate from Administrator to SYSTEM
  - Generally by making a call to the WinLogon process to get a SYSTEM token and then self executing with that token.
- An attacker could also abuse this privilege
  - JuicyPotato, PrintSpoofer and RoguePotato (for Windows Server 2019 and Windows 10 build 1809 onwards) - tricks a process running as SYSTEM to connect to their process, which hands over the token to be used.
  - After gaining RCE under the context of a service account (svc_mssql) an attacker can check for this privilege to quickly and easily elevate privileges.

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

Host Privilege Escalation
Windows User Privileges

SeImpersonatePrivilege – JuicyPotato

- After gaining a foothold on an SQL Server using a service user with elevated privileges
-

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

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Host Privilege Escalation Windows Group Privileges

Windows Services Privilege Escalation

- A Windows "service" is a special type of application that is usually started automatically when the computer boots.
- Services are used to start and manage core Windows functionality such as Windows Defender, Windows Firewall, Windows Update and more. Third party applications may also install a Windows Service to manage how and when they're run.
- If services run with System privileges and are misconfigured exploiting them may lead to command execution with SYSTEM privileges as well.

Windows Services Privilege Escalation

- A service has several properties:
  - **Binary Path:** the path where the actual executable (.exe) for the service is located
  - **Startup Type:** when the service should start
    - Automatic - The service starts immediately on boot.
    - Automatic (Delayed Start) - The service waits a short amount of time after boot before starting (mostly a legacy option to help the desktop load faster).
    - Manual - The service will only start when specifically asked.
    - Disabled - The service is disabled and won't run.
  - **Service Status:** the current status of the service
    - Running - The service is running.
    - Stopped - The service is not running.
    - StartPending - The service has been asked to start and is executing its startup procedure.
    - StopPending - The service has been asked to stop and is executing its shutdown procedure.
  - **Log On As:** the user account (domain or local) that the service is configured to run as

Windows Services Privilege Escalation

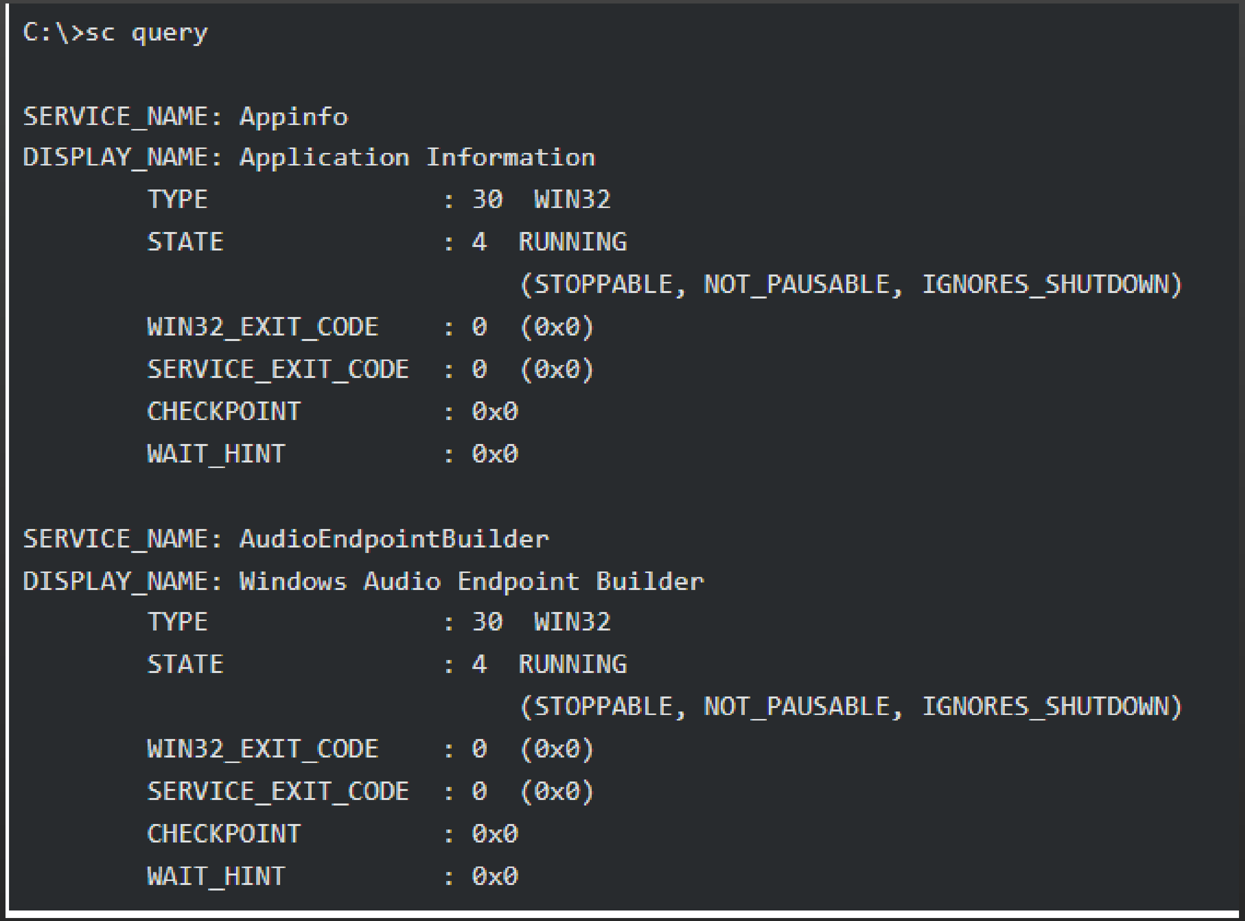
- Like files and folders - services have permissions assigned to them. This controls which users can modify, start or stop the service. Some highly sensitive services such as Windows Defender cannot be stopped, even by administrators. Other services may have much weaker permissions that allow standard users to modify them for privilege escalation.
- After a service has been manipulated to trigger a privilege escalation, it needs to be restarted (or started if it's already stopped). There will be cases where this can be done with the management tools, if you have the required permissions. Other times, you'll need to rely on a reboot.

Windows Services Privilege Escalation

Service Commands

- Query the configuration of a service
  - Sc.exe qc [service name]
- Query the current status of a service
  - Sc.exe query [service name]
- Modify a configuration option of a service
  - Sc.exe config [service name] [option]= [value]
- Start/Stop a service
  - Net start/stop [service name]

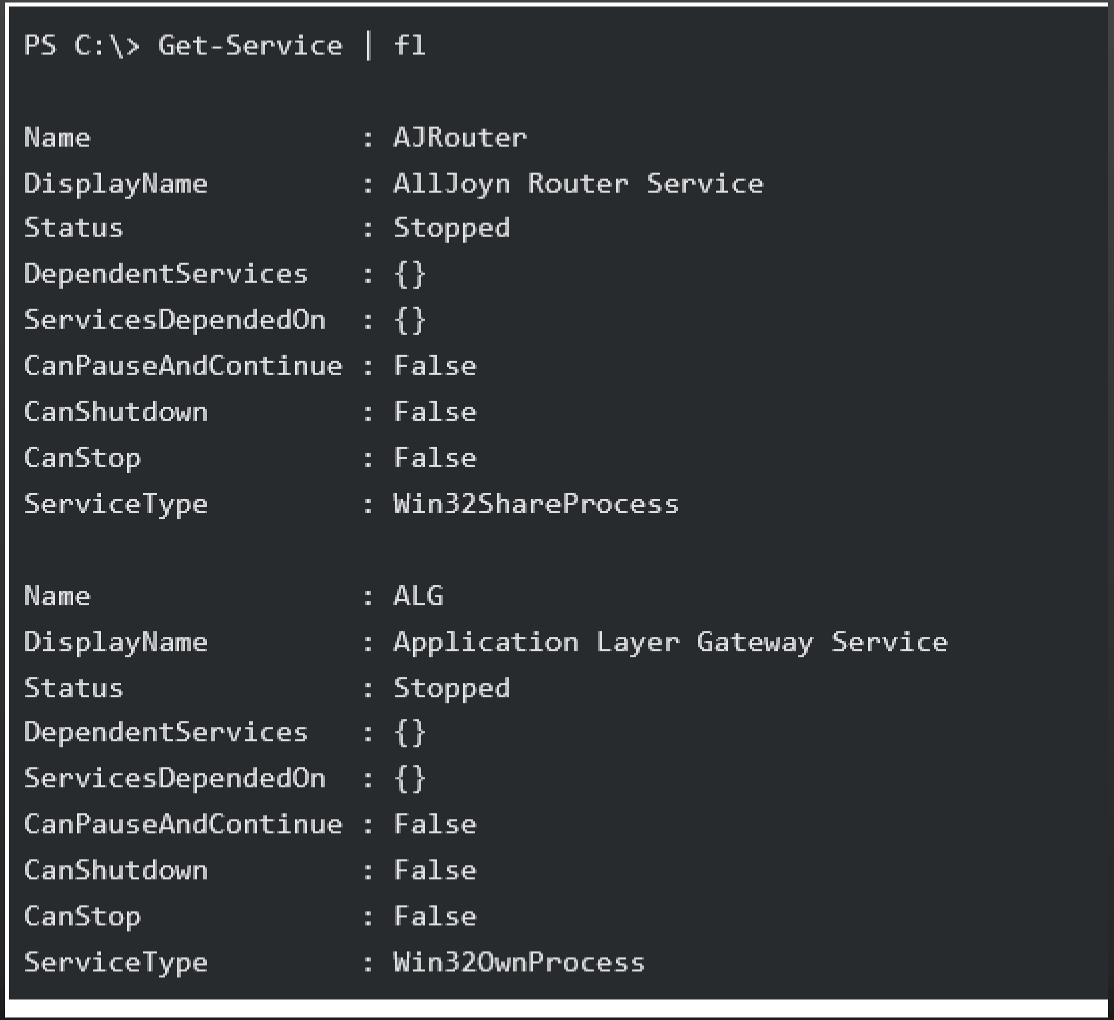
Windows Services Privilege Escalation



```
C:\>sc query

SERVICE_NAME: Appinfo
DISPLAY_NAME: Application Information
        TYPE               : 30  WIN32
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

SERVICE_NAME: AudioEndpointBuilder
DISPLAY_NAME: Windows Audio Endpoint Builder
        TYPE               : 30  WIN32
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```



```
PS C:\> Get-Service | fl

Name                : AJRouter
DisplayName         : AllJoyn Router Service
Status              : Stopped
DependentServices   : {}
ServicesDependedOn  : {}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : False
ServiceType         : Win32ShareProcess

Name                : ALG
DisplayName         : Application Layer Gateway Service
Status              : Stopped
DependentServices   : {}
ServicesDependedOn  : {}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : False
ServiceType         : Win32OwnProcess
```

Windows Services Privilege Escalation
Insecure Service Permissions

Service specific permissions

- SERVICE_QUERY_CONFIG,SERVICE_QUERY_STATUS are innocuous
- SERVICE_STOP,SERVICE START are useful
- SERVICE_CHANGE_CONFIG,SERVICE_ALL_ACCES are dangerous

Windows Services Privilege Escalation
Insecure Service Permissions

- If a user has permission to change the configuration of a service which runs with SYSTEM privileges ,we can change the executable the service users to one of our own .

- SharpUp.exe audit ModifiableServices can enumerate for modifiable services.
- To exploit:
  - Upload a malicious service executable
  - Change the binary: sc config VulnerableService binPath= C:\path\to\uploaded.svc.exe
  - Run the service: sc qc VulnerableService

  * If the service is already running we must stop and start the service.

OPSEC

Restore the service configuration once you are done. Ensure you don't interrupt business critical services, so seek permission before exploiting these types of vulnerabilities.

Windows Services Privilege Escalation Insecure Service Binary Permissions

- A variation on the previous vulnerability but instead of the insecure permissions being on the service, it's on the service binary itself

Windows Services Privilege Escalation Unquoted Service Path

- Executables in Windows can be run without using their extension.
- An unquoted service path is where the path to the service binary is not wrapped in quotes. This is not a problem by itself, but under specific conditions it can lead to an elevation of privilege.
- Some executables take arguments, separated by spaces, e.g. program.exe arg1 arg2 ..
- This behavior leads to ambiguity when using absolute paths that are unquoted and contain spaces.
- We can use WMI to pull a list of every service and the path to its exe.
  - wmic service get name, pathname

Service Privilege Escalation
Unquoted Service Path

- Consider the following unquoted path:
  - C:\Program Files\A Dir\program.exe
- To us ,this obviously runs program.exe.
- When Windows attempts to read the path to this executable, it interprets the space as a terminator
- So to Windows, C:\Program could be the executable, with two arguments: “Files\A” and “Dir\program.exe” Windows resolves this ambiguity by checking each of the possibilities in turn.
- If we have permissions to write to a location Windows checks before the actual executable, we can trick the service into executing it instead.
- SharpUp.exe audit UnquotedServicePath can enumerate this vuln.

Service Privilege Escalation
Weak Registry Permissions

- The Windows registry stores entries for each service. Since registry entries can have ACLs, if the ACL is misconfigured, it may be possible to modify a service's configuration even if we cannot modify the service directly.

Windows Services Privilege Escalation DLL Hijacking

- A more common misconfiguration that can be used to escalate privileges is if a DLL is missing form the system, and our user has write access to a directory within the PATH that Windows searches for DLLs in.
- The initial detection of vulnerable services is difficult , and often the entire process is very manual.

Registry Privilege Escalation - AutoRuns

- Windows can be configured to run commands at startup, with elevated privileges.
- These “Autoruns” are configured in the Registry.
- If you are able to write to an AutoRun executable, and are able to restart the system (or wait for it to be restarted) you may be able to escalate privileges.

Registry Privilege Escalation
AlwaysInstallElevated

- MSI files are package files used to install applications.
- These files run with permissions of the user trying to install them .
- Windows allows for these installers to be run with elevated(i.e admin) privileges.
- If this is the case, we can generate a malicious MSI file which contains a reverse shell.

UAC Bypasses

- User Account Control (UAC) is a technology that exists in Windows which forces applications to prompt for consent when requesting an administrative access token.
- For example if a local administrator on a Workstation opens a Command Prompt and attempts to add a new local user, he will get an access denied. This instance of cmd.exe is running in "medium integrity".
- Instead, he must right-click and select "Run as administrator", which will cause a UAC prompt to appear.
- After that we go from Medium to High Mandatory Level

UAC Bypasses

- A UAC "bypass" is a technique that allows a medium integrity process to elevate itself or spawn a new process in high integrity, without prompting the user for consent. Being in high integrity is important for attackers because it's required for various post-exploitation actions such as dumping credentials.

- Cobalt strike beacon has a few built-in UAC bypasses.
- A few more with Elevate Kit

- Some insights for UAC Bypass Techniques:

https://www.elastic.co/security-labs/exploring-windows-uac-bypasses-techniques-and-detection-strategies

Fodhelper UAC Bypass

- Windows 10 environments allow users to manage language settings for a variety of Windows features such as typing, text to speech etc. When a user is requesting to open “Manage Optional Features” in Windows Settings in order to make a language change a process is created under the name fodhelper.exe. This process is running as high integrity due to the fact the binary has the autoelevate setting to “true”.
- However processes that are running with higher privileges can give the opportunity to an attacker to execute code with the same level of privileges if they can be abused in a certain way. Specifically winscripting discovered that the “fodhelper” process when it starts it tries to find some registry keys which doesn’t exist.

Fodhelper UAC Bypass

The following checks are performed in the registry upon start of fodhelper.exe:

- HKCU:\Software\Classes\ms-settings\shell\open\command
- HKCU:\Software\Classes\ms-settings\shell\open\command\DelegateExecute
- HKCU:\Software\Classes\ms-settings\shell\open\command\(default)

- When fodhelper (a Windows binary that allows elevation without requiring a UAC prompt) is launched by malware as a Medium integrity process, Windows automatically elevates fodhelper from a Medium to a High integrity process. The High integrity fodhelper then attempts to open an ms-settings file using its default handler. Since the medium-integrity malware has hijacked this handler, the elevated fodhelper will execute a command of the attacker's choosing as a high integrity process.

Fodhelper UAC Bypass with Havoc

- In a medium integrity level grunt import Fodhelper.ps1 powershell script from Tools:

  Powershell Import…..
- After that run:
  - Powershell helper –custom “cmd.exe /c <powershell launcer payload>”
- And we have a high integrity level session grunt abusing a legit Microsoft binary.
- Then create a binary payload for http listener and inject it in winlogon process that is running as System and you upgrade to a System level session of session 1 windows session.

Passwords Privilege Escalation Registry

- Plenty of programs store configuration options in the Windows Registry.
- Windows itself sometimes will store passwords in plaintext in the Registry.
- It is always worth searching the Registry for passwords

Passwords Privilege Escalation

Saved Credentials

- Windows has a runas command which allow users to run commands with the privileges of other users.
- This usually requires the knowledge of the other user’s password.
- However, Windows also allows users to save their credentials to the system , and these saved credentials can be used to bypass this requiremets.

Passwords Privilege Escalation
Searching for Configuration Files

- Recursively search for files in the current directory with pass in the name , or ending in “.config”:
  - Dir /s *pass* ==*.config
- Recursively search for files in the current directory that contain the word “password” and also end in either .xml, .ini, or txt.
  - Findstr /si password *.xml *.ini *.txt

Passwords Privilege Escalation
Custom Applications

- Custom application may contain password in order to communicate with an other system or an api.
- PS> strings custom.exe
- Use a debugger o disassembler to examine the executable