Windows-Privasc

What we are going to learn

- Kernel Exploits
- password hunting
- impresonation Attacks
- Registry Attacks
- Executable Files
- Schedule Tasks
- Startup Applications
- DLL Hijacking
- Service Permissions
- Windows Sbusystem for linux
- CVE-2019-1388

Resources

- Fuzzysecurity.com Windows Privilege Escalation
- swisskyrepo/PayloadsAllTheThings Github Windows Privasc
- http://www.absolomb.com/2018-01-26-Windows-Privilege-Escalation-Guide/
- http://sushant747.gitbooks.io/total-oscp-guide/privilege_escalation_windows.html

Gaining A Foothold

- nmap -A -T4 -p- <ip>

Initial Enumeration

- **system Enumeration**
  - systeminfo

    - systeminfo | findstr /B /C:″OS Version″ /C:″System Type″
    - hostname
    - wmic qfe
    - wmic logicaldisk get caption,description,providername
- **User Enumeration**
    - whoami
    - whoami /priv
    - whoami
    - net user
    - net user Administrator
    - net localgroup
    - net local group administrators
- **Network Enumeration**
    - ipconfig
    - ipconfig -all
    - arp -a
    - netstat -ano
- **Password Hunting**
    - findstr /si password *.txt
- **Firewall Enumeration and AV**
    - sc query windefend “sc =service control″
    - sc qeuryex type
    - query windefend
    - sc queryex type= service
    - netsh advfirewall firewall dump
    - netsh firewall sow state

Automated Tools

- **Executables**
  - winPEAS.exe
  - Seatbelt.exe (compile)
  - Watson.exe (compile)
  - SharpUP.exe (compile)
- **PoserShell**
  - Shelock.ps1
  - PowerUP.ps1
  - jaws-enum.ps1
- Other
  - windows-exploit-suggester.py (local)
  - Exploit Suggester (Metasploit)

Path Kernel Exploit

- “use automated tool for get the vul”
- “msfvernom for shell”
- nc listner
- execute it

Path Password and Port Forwading

- “look for the pasword serach”
- “sometimes you need to forward the port”

Windows Subsystem for Linux

- where /R c:/windows bash.exe
- “there are more things, ive not make notes”

Impersonation and Potato Attacks

- **Tokens**
  - temporary keys that allow you access to a system/network without having to provide credentials eachtime you access a file. think cookies for computers
- **Two types**
  - Delegate - Created for logging into a machine or using Remote Desktop
  - Impersonate - “non-interactive” such as attaching a network driver or a domain logon script
- whoami /priv
- **Impersonations**
  - An impersonator is someone who imitates or copies the behavior or actions of another.
- **What is Token Impersonation?**

  Token impersonation is a mechanism in Windows that allows a thread to temporarily "impersonate" another user's security context. This is useful in scenarios where a service needs to perform actions on behalf of a client.
- **Potato Attack ***
  - $you can also use windows exploit suggestor to check for potato or token impersonation attack
  - ADS “hiddern file in windows” Aleternate Data Stream

GETSYSTEM

- Mestasploit

RunAs

“it is use to run tha command like somebody else”

Registry

What is the Windows Registry?

The Windows Registry is a hierarchical database that stores low-level settings for the operating system and for applications that opt to use it. Think of it as a central repository for configuration settings, options, and preferences for Windows and installed software. It contains information like:

- User profiles and preferences
- System hardware details
- Installed software configurations
- Startup programs
- Security policies
- File associations
- And much more

The Registry is organized into a tree-like structure with **keys** and **values**. The main sections of the Registry are called **hives**, and they are divided into five root keys:

1. **HKEY_CLASSES_ROOT (HKCR)**: File associations and COM object registration.
2. **HKEY_CURRENT_USER (HKCU)**: Settings for the currently logged-in user.
3. **HKEY_LOCAL_MACHINE (HKLM)**: System-wide settings for all users.
4. **HKEY_USERS (HKU)**: Contains profiles for all users on the system.
5. **HKEY_CURRENT_CONFIG (HKCC)**: Hardware profile settings used at startup.

How is the Registry Involved in Privilege Escalation?

The Registry is a common target for privilege escalation because it controls many aspects of how Windows operates. Attackers (or pentesters) can exploit misconfigurations or weak permissions in the Registry to escalate privileges. Here are some ways the Registry is involved:

1. Service Misconfigurations

- Windows services often store their configurations in the Registry (e.g., under `HKLM\SYSTEM\CurrentControlSet\Services`).
- If a service is configured to run with elevated privileges (e.g., as `SYSTEM`) and its executable path or dependencies can be modified via the Registry, an attacker can replace the executable with a malicious one.

2. Autorun Programs

- The Registry contains keys that define which programs run at startup (e.g., `HKLM\Software\Microsoft\Windows\CurrentVersion\Run`).
- If an attacker can write to these keys, they can make a malicious program run with elevated privileges when the system starts or a user logs in.

3. AlwaysInstallElevated

- The `AlwaysInstallElevated` Registry keys (`HKLM\Software\Policies\Microsoft\Windows\Installer` and `HKCU\Software\Policies\Microsoft\Windows\Installer`) allow non-admin users to install MSI packages with elevated privileges.
- If these keys are enabled (set to `1`), an attacker can create a malicious MSI package to gain admin privileges.

4. Unquoted Service Paths

- Some services reference executables with unquoted paths in the Registry (e.g., `C:\Program Files\Some Folder\Service.exe`).
- If the path contains spaces and is unquoted, Windows will look for the executable in each segment of the path. An attacker can place a malicious executable in a writable directory (e.g., `C:\Program.exe`) to escalate privileges.

5. Weak Registry Permissions

- If the permissions on a Registry key are misconfigured, an attacker might be able to modify or replace critical values.
- For example, if `HKCU\Environment` has weak permissions, an attacker could add a malicious executable to the `PATH` or modify environment variables to execute code.

6. Credential Storage

- The Registry sometimes stores credentials or sensitive data (e.g., in `HKLM\SECURITY` or `HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings`).
- If an attacker can access these keys, they might extract passwords or other sensitive information.

7. AppInit_DLLs

- The `AppInit_DLLs` key (`HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows`) can be used to load DLLs into every process that uses `User32.dll`.
- If an attacker can write to this key, they can load a malicious DLL into high-privilege processes.

Executable Files

Startup Applications

DLL-hijacking

Service path permissions